NVIDIA Orin Series System-on-Chip

Technical Reference Manual

Abstract

The Technical Reference Manual focuses on the logical organization and control of the NVIDIA Orin Series System-on-Chip. It provides information for those modules that interface to external devices, or those that control fundamental chip operations. The modules detailed in this document provide an overview, any necessary programming guidelines, and a register listing for that module. Internal functional units such as video and graphics hardware acceleration are controlled by NVIDIA provided software and are not documented here.

Version: 1.2

Date: 29-September-2023

ID: DP-10508-002

Table of Contents

1. Revision History 25
1.1 About This Release 25
1.2 Change Log 26
1.2.1 20230929 (Version 1.2) 27
2. Getting Started 28
2.1 Overview 28
2.1.1 Reading Register Tables 29
2.1.2 Units 32
2.1.3 Glossary 32
3. Memory Architecture and Memory Mapped I/O 42
3.1 MSS Registers 42
3.2 Address Map 44
3.2.1 Overview 44
3.2.1.1 Features 45
3.2.1.2 AMAP and Aperture 45
3.2.1.3 MMIO Aperture 46
3.2.1.4 On-Chip Data/Sync Plane Aperture 47
3.2.1.5 DRAM Aperture 48
3.2.1.6 PCIe Aperture (32-bit OS) 49
3.2.1.7 Off-Chip Aperture 50
3.2.2 System Address Map 51
3.3 Address Space Translation (AST) 81
3.3.1 Overview 81
3.3.1.1 Features 82
3.3.2 Functional Description 83
3.3.2.1 Address Generation Blocks 84
3.3.2.2 Read Error Response 84
3.3.2.3 Write Error Response 85
3.3.2.4 Decode Error Generation 85

3.3.2.5 Clocks and Resets 85
3.3.2.6 AST Software Requirements 86
3.3.2.7 Programming Examples 86
3.3.3 Programming Guidelines 87
3.3.3.1 Steps to Configure Region #i in AST 87
3.3.4 AST Registers 88
3.4 General Purpose Direct Memory Access (DMA) Engines 117
3.4.1 Overview 117
3.4.1.1 Features 119
3.4.1.2 New Hardware Features 119
3.4.1.3 Support for Legacy MMIO DMA Hardware Features 119
3.4.1.4 Supported Legacy Software Features 120
3.4.2 Functional Description 121
3.4.2.1 DMA Functionality: Mem-MMIO 121
3.4.2.2 Peripheral Rx Mode: DMA Read from MMIO Peripheral to Memory 121
3.4.2.3 Peripheral Tx Mode: DMA Write from Memory to MMIO 122
3.4.2.4 MEM-MEM DMA Mode 122
3.4.3 Programming Guidelines 122
3.4.3.1 Main Programming Steps 122
3.4.3.2 GPC-DMA Address Space 123
3.4.3.3 Pause Mode 123
3.4.3.4 Abrupt Channel Disable 124
3.4.3.5 Busy Bit 124
3.4.3.6 Enable (ENB) Bit 124
3.4.3.7 DMA Interrupt 124
3.4.3.8 Controller FIFO Size 125
3.4.3.9 Transfer Alignment 125
3.4.3.10 WCOUNT Register 125
3.4.3.11 Continuous Mode 125
3.4.3.12 HOL Blocking 126
3.4.3.13 Non-continuous SPI Mode 126
3.4.4 GPC-DMA Registers 126
3.4.4.1 COMMON Registers 126

3.4.4.2 CHANNEL Registers 212
3.5 System Memory Management Unit (SMMU) 468
3.5.1 Overview 468
3.5.1.1 SMMU Configuration 468
3.5.1.2 StreamID 469
4. Boot and Power Management 470
4.1 Boot and Power Management Processor (BPMP) 470
4.1.1 Overview 470
5. CPU Complex (CCPLEX) 471
5.1 Overview 471
5.1.1 Reference Documentation 471
5.1.1.1 List of References 471
5.1.2 Glossary 472
5.1.3 Conventions of Units 474
5.2 CCPLEX Functional Description 474
5.2.1 Architecture Overview 474
5.2.1.1 Key Features 475
5.2.1.2 Architecture Summary 475
5.2.1.3 CPU Clusters 477
6. GPU 484
6.1 Overview 484
6.1.1 Streaming Multiprocessor (SM) 485
6.1.1.1 Third-Generation Tensor Core 487
6.1.1.2 2x FP32 Processing 488
6.1.1.3 Second-Generation Ray Tracing (RT) Core 489
6.1.1.4 Larger and Faster Unified Shared Memory and L1 Data Cache 490
6.1.1.5 Texture (TEX) unit 490
6.1.2 PolyMorph Engine 491
6.1.3 Raster Engine 491
6.1.4 L2 Cache 491
6.1.5 Host Interface 491
6.1.6 MSS Interface 491
7. Multimedia Complex 492

7.1 Host Controller 492
7.1.1 Overview 492
7.1.1.1 List of References 492
7.1.1.2 Glossary 492
7.1.1.3 Relevant Chapters in the TRM 495
7.1.1.4 Features 495
7.1.2 Functional Description 496
7.1.2.1 Channels 496
7.1.2.2 Classes 496
7.1.2.3 AXI Interface Block 498
7.1.2.4 Syncpt (Sync Points) Block 499
7.1.2.5 Unit Activity Monitor (ACTMON) 501
7.1.2.6 Command Processor 501
7.1.2.7 Command DMA Block 502
7.1.2.8 Interrupt Logic 502
7.1.2.9 Filtering out Kernel Commands from the Gather Buffer 504
7.1.2.10 SMMU Context Protection (StreamID Protection) 504
7.1.2.11 Direct Register Access 506
7.1.2.12 Error Detection 509
7.1.3 Programming Guidelines 510
7.1.3.1 Channel Programming Model 510
7.1.3.2 Priority between Channels 517
7.1.3.3 ACQUIRE_MLOCK 518
7.1.3.4 Syncpt Protection 518
7.1.3.5 DMA Commands 520
7.1.3.6 Progress Status 522
7.1.3.7 Programming Model for MMIO-Based Clients (Single MMIO Range) 522
7.1.3.8 Programming Model for MMIO-Based Clients (Multiple MMIO Range) 523
7.1.3.9 Host Controller Class 528
7.1.3.10 Class-based Programming 529
7.1.3.11 Context Switching 529
7.1.3.12 Specific Programming Guidelines 530
7.2 Camera Subsystem 534

7.2.1 MIPI Camera Serial Interface (CSI) 534
7.2.1.1 NVCSI CORE 537
7.2.1.2 NVCSI SCIL 547
7.2.1.3 NVCSI Registers 555
7.2.2 Video Input (VI) 1904
7.2.2.1 Overview 1904
7.2.2.2 Functional Description 1908
7.2.2.3 Video Input Registers 1945
7.3 Video Image Compositor (VIC) 3335
7.3.1 Overview 3335
7.3.1.1 Use Cases 3336
7.3.1.2 Hardware Description 3337
7.3.1.3 Pixel Format Conversion 3340
7.3.1.4 Geometry Transform Processing 3358
7.3.1.5 Power Management 3361
7.3.1.6 Virtualization 3364
7.3.2 VIC Programming Guidelines 3365
7.3.2.1 Class Overview 3365
7.3.2.2 Application Enumerations 3368
7.3.2.3 Application Memory Structures 3377
7.3.2.4 Application Methods 3398
7.3.2.5 Method Naming and Programming 3400
7.3.2.6 Programming Restrictions 3407
7.3.2.7 Performance and Power Tuning Guidelines 3409
7.3.2.8 Program Guideline 3413
7.3.3 VIC Registers 3419
7.3.3.1 THI Registers 3419
7.3.3.2 FALCON Registers 3428
7.3.3.3 TFBIF Registers 3524
7.4 High Definition Audio (HDA) 3538
7.4.1 Overview 3538
7.4.1.1 High-level Features 3538
7.4.2 HDA Registers 3539

7.4.2.1 FCPI Interface Registers 3539
7.4.2.2 HDMI HDAC Registers 3561
7.5 Display Controller 3569
7.5.1 Overview 3569
7.5.1.1 Standards Supported 3570
7.5.1.2 Glossary 3570
7.5.1.3 Relevant Chapters in the TRM 3572
7.5.1.4 Features 3572
7.5.1.5 Modes of Operation 3586
7.5.2 Functional Description 3592
7.5.2.1 Front End Host Interface 3593
7.5.2.2 Clocks 3599
7.5.2.3 Coordinate Systems 3603
7.5.3 Display Controller Registers 3607
7.6 Consumer Electronics Control (CEC) 3615
7.6.1 Overview 3615
7.6.2 Functional Description 3616
7.6.3 Programming Guidelines 3616
7.6.3.1 Initialization 3616
7.6.4 CEC Registers 3617
7.6.4.1 Registers 3617
7.7 Audio Processing Engine (APE) 3633
7.7.1 Overview 3633
7.7.1.1 Standard and Compatibility 3635
7.7.1.2 List of Reference 3636
7.7.1.3 Glossary 3636
7.7.1.4 Relevant Chapters in the TRM 3637
7.7.1.5 Features 3638
7.7.2 Functional Descriptions 3638
7.7.2.1 Audio Digital Signal Processor 3639
7.7.2.2 Audio Hub 3639
7.7.2.3 ADMA 3654
7.7.2.4 REGDEC 3657

7.7.2.5 ACONNECT 3658
7.7.2.6 Audio RAM and Audio Memory Controller 3658
7.7.2.7 APE Interrupt Controller 3658
7.7.2.8 Audio Miscellaneous (AMISC) 3662
7.7.2.9 ACAST and ADAST 3663
7.7.2.10 APB-AXI Shim 3663
7.7.2.11 APE HSP 3663
7.7.2.12 APE TKE 3664
7.7.3 Programming Guidelines 3664
7.7.3.1 System Level Programming 3664
7.7.3.2 Using Interrupts 3665
7.7.3.3 Programming Module RAMs 3665
7.7.3.4 AHUB Sub-modules Soft Reset Sequence 3666
7.7.3.5 Boot Sequence 3666
7.7.3.6 Power Management Sequence 3670
7.7.3.7 SC7 Entry and Exit 3673
7.7.3.8 AAS Programming Guidelines 3675
7.7.3.9 ACIF Programming Guidelines 3675
7.7.3.10 I2S Controller Programming Guidelines 3676
7.7.3.11 DMIC Programming Guidelines 3678
7.7.3.12 Mixer Programming Guidelines 3679
7.7.3.13 AMX Programming Guidelines 3684
7.7.3.14 ADX Programming Guidelines 3685
7.7.3.15 SFC Programming Guidelines 3686
7.7.3.16 AFC Programming Guidelines 3687
7.7.3.17 OPE Programming Guidelines 3688
7.7.3.18 MVC Programming Guidelines 3689
7.7.3.19 DSPK Programming Guidelines 3691
7.7.3.20 ASRC Programming Guidelines 3692
7.7.3.21 ARAD Programming Guidelines 3695
7.7.3.22 ADMA Programming Guidelines 3696
7.7.4 APE Registers 3697
7.7.4.1 Audio Crossbar (AXBAR) Registers 3697

7.7.4.2 Audio Hub Configuration (AHC) Registers 3741
7.7.4.3 Inter-Chip Sound (I2S) Control Registers 3751
7.7.4.4 Digital Microphone Control (DMIC) Registers 3766
7.7.4.5 Digital Speaker (DSPK) Control Registers 3777
7.7.4.6 Audio Multiplexer (AMX) Control Registers 3787
7.7.4.7 Audio De-Multiplexer (ADX) Control Registers 3808
7.7.4.8 Mixer Control Registers 3824
7.7.4.9 Master Volume Control (MVC) Registers 3844
7.7.4.10 Arbitrary Sample Rate Control (ASRC) Registers 3859
7.7.4.11 Audio Sampling Rate Detector (ARAD) Control Registers 3913
7.7.4.12 Audio Flow Control (AFC) Registers 3945
7.7.4.13 Sampling Frequency Converter (SFC) Control Registers 3965
7.7.4.14 Output Processing Engine (OPE) Control Registers 3974
7.7.4.15 Parametric Equalizer (PEQ) Control Registers 3981
7.7.4.16 Multiband Dynamic Range Compression (MBDRC) Control Registers 3985
7.7.4.17 Audio Direct Memory Access Interface (ADMAIF) Control Registers 3995
7.7.4.18 Audio Direct Memory Access (ADMA) Control Registers 4195
7.7.4.19 Audio Memory Control (AMC) Registers 5363
7.7.4.20 Audio Digital Signal Processor Peripheral (ADSP_PERIPH) Control Registers 5372
7.7.4.21 Audio Miscellaneous (AMISC) Control Registers 5433
7.7.4.22 Audio Generic Interrupt Control (AGIC) Registers 5445
7.8 Always On Digital Microphone (AODMIC) 5497
7.8.1 Overview 5497
7.8.1.1 Features 5499
7.8.2 Functional Description 5499
7.8.2.1 Receive Buffer and Differentiator 5499
7.8.2.2 Integrator 5500
7.8.2.3 Downsampler 5500
7.8.2.4 Physical Structure 5501
7.8.3 Programming Guidelines 5502
7.8.3.1 Normal Operation (Enabling) 5502
7.8.3.2 Disabling 5502
7.8.3.3 Optional 5502

7.8.4 AODMIC Registers 5502
7.9 Pixel Memory Formats 5516
7.9.1 Functional Description 5516
7.9.1.1 Tiling Formats 5516
7.9.1.2 DRAM Organization 5518
7.9.1.3 Block Linear 5519
7.9.1.4 Sub-Partitions 5522
7.9.1.5 Hardware Support 5523
7.9.1.6 Pixel Format Naming 5523
7.9.2 Programming Guidelines 5560
7.9.2.1 Software Guidelines 5560
7.9.2.2 Block Height Limitations 5561
8. System Components 5563
8.1 Clock and Reset Controller (CAR) 5563
8.1.1 Overview 5563
8.1.2 Reset Controller 5563
8.1.3 Clock Controller 5564
8.2 Interrupt Controllers 5565
8.2.1 Overview 5565
8.2.1.1 List of References 5566
8.2.1.2 Glossary 5566
8.2.1.3 Relevant Chapters in the TRM 5567
8.2.1.4 Specifics in Reading This Chapter 5567
8.2.2 Functional Description 5568
8.2.2.1 Interrupt Handling Mechanism 5568
8.2.2.2 Interrupt Controllers 5571
8.2.2.3 Interrupt Routing 5574
8.2.2.4 Interrupt Mapping 5575
8.2.2.5 Interrupt Handling by Different Cortex-R5 Clusters 5600
8.2.3 Interrupt Controller Registers 5625
8.2.3.1 CHANNEL Registers Summary 5626
8.2.3.2 COMMON Registers Summary 5634
8.2.3.3 Interrupt Controller Common Registers 5636

8.2.3.4 Interrupt Controller Channel Registers 5647
8.3 Timers 5779
8.3.1 Overview 5779
8.3.1.1 Glossary and Acronyms 5780
8.3.1.2 Relevant Chapters in the TRM 5781
8.3.1.3 Address Map 5781
8.3.1.4 Clocks 5782
8.3.1.5 Resets 5783
8.3.2 Functional Description 5784
8.3.2.1 Hierarchical Grouping 5786
8.3.2.2 TSC 5788
8.3.2.3 Main TSC Generation 5789
8.3.2.4 Secondary TSC 5791
8.3.2.5 External Reference Tracking 5791
8.3.2.6 TSC Controlled Signals 5795
8.3.2.7 TSC Distribution Logic 5798
8.3.2.8 Halting TSC During Debug 5798
8.3.2.9 Generic Timestamping Engine 5802
8.3.2.10 RefClk Source Change Logic 5804
8.3.2.11 Power Management 5804
8.3.2.12 Real Time Clock 5806
8.3.2.13 Always On WatchDog Timer 5808
8.3.2.14 Microsecond Counter and Time Reference Pulses Generation 5808
8.3.2.15 TKE Module 5809
8.3.2.16 Generic Timers 5814
8.3.2.17 Cortex-A9 Timers 5815
8.3.2.18 Functional Safety Island Timers 5815
8.3.2.19 TSC Update Frequency 5822
8.3.2.20 Safety Mechanism 5823
8.3.3 Programming Guidelines 5824
8.3.3.1 Use of Top TKE for CPU Core Wake 5824
8.3.3.2 Context Save and Restore Across Power State Transitions 5825
8.3.4 Timer Registers 5831

8.3.4.1 Time Keeping Element Shared Registers 5832
8.3.4.2 Time Keeping Element: TOP Registers 5890
8.3.4.3 Real-Time Clock Registers 5956
8.3.4.4 Time Stamp Counter System Control Registers 5969
8.3.4.5 Time Stamp Counter System Control Registers Continued 5983
8.3.4.6 Time Stamp Counter Registers 6006
8.3.4.7 Microsecond Counter Registers 6038
8.3.4.8 Generic Timestamping Engine (GTE) APS Registers 6040
8.3.4.9 Generic Timestamping Engine (GTE) GPIO Registers 6046
8.3.4.10 Generic Timestamping Engine (GTE) Interrupt Controller Registers 6053
8.4 Multi-Purpose I/O Pins and Pin Multiplexing (PinMux) 6070
8.4.1 Overview 6070
8.4.1.1 Standard and Compatibility 6072
8.4.1.2 Glossary 6073
8.4.1.3 Relevant Chapters in the TRM 6074
8.4.1.4 MPIO Pad Features 6075
8.4.1.5 Pad Control Unit 6084
8.4.1.6 Pad Control Grouping 6085
8.4.1.7 PinMux 6086
8.4.2 Functional Description 6088
8.4.2.1 Programmable Drive Strength Capabilities 6088
8.4.2.2 Pin-states and Default Pad Controls 6089
8.4.2.3 I/O Electrical Capabilities and Special Handling 6090
8.4.2.4 External Wake Events 6093
8.4.2.5 GPIO Controller to Ports Mapping 6094
8.4.2.6 DFD OBS (Observation Bus) Signals 6095
8.4.2.7 DPD 6096
8.4.2.8 IST and I/O Considerations 6100
8.4.3 Programming Guidelines 6108
8.4.3.1 PinMux Programming 6108
8.4.3.2 Pad Control Programming 6111
8.4.3.3 I/O Safe State Programming during IST 6112
8.4.3.4 Unused I/O Programming 6113

8.4.3.5 DPD Sequencing 6115
8.4.3.6 Wake Events Configuration 6117
8.4.4 G3 PAD Control Registers 6118
8.4.5 G5 PAD Control Registers 6154
8.4.6 G4 PAD Control Registers 6172
8.4.7 Debug PAD Control Registers 6211
8.4.8 EMMC PAD Control Registers 6221
8.4.9 PEX_CTL PAD Control Registers 6238
8.4.10 SDMMC1 High Voltage PAD Control Registers 6262
8.4.11 QSPI PAD Control Registers 6281
8.4.12 SYS PAD Control Registers 6302
8.4.13 G2 PAD Control Registers 6318
8.4.14 Always On PAD Control Registers 6356
8.4.15 Always On High Voltage PAD Control Registers 6370
8.4.16 EDP PAD Control Registers 6388
8.4.17 UFS PAD Control Registers 6417
8.4.18 PEX_CTL_2 PAD Control Registers 6429
8.4.19 EQOS PAD Control Registers 6443
8.4.20 PEX_CTL_3 PAD Control Registers 6467
8.4.21 G7 PAD Control Registers 6486
8.4.22 G8 PAD Control Registers 6509
8.4.23 G9 PAD Control Registers 6522
8.4.24 G10 PAD Control Registers 6549
8.4.25 G11 PAD Control Registers 6570
8.5 GPIO Controllers 6582
8.5.1 Overview 6582
8.5.1.1 Relevant Chapters in the TRM 6583
8.5.1.2 Features 6583
8.5.2 Functional Description 6585
8.5.2.1 Basic GPIO Controller Structure 6585
8.5.2.2 GPIO TimeStamping Interface 6591
8.5.2.3 GPIO Clock and Reset Control 6592
8.5.2.4 GPIO Interrupts 6594

8.5.2.5 AON GPIO Controller 6599
8.5.2.6 APB Slave Error support 6600
8.5.2.7 Virtualization 6600
8.5.3 Registers 6602
8.5.3.1 System Address Map (AMAP) 6602
8.5.3.2 Registers Details 6604
8.5.4 Programming Model 6614
8.5.4.1 GPIO Pad Control Initialization 6614
8.5.4.2 GPIO Input Configuration 6615
8.5.4.3 GPIO Output/Bi-directional Configuration 6616
8.5.4.4 GPIO Interrupts 6617
8.5.4.5 Enabling Security Attributes for GPIO Registers 6619
8.5.4.6 GPIOs in Loopback Mode 6619
8.5.4.7 Disabling Specific GPIO 6619
8.5.4.8 GPIO Programming Entities 6620
8.5.4.9 Interrupt Handling Options and Guidelines 6620
8.5.5 Platform Aspects 6620
8.5.5.1 De-bounce Filter Threshold Selection 6620
8.5.5.2 GPIO Usage Restrictions in Platforms 6620
8.5.6 GPIO Control Registers 6621
8.5.7 GPIO Common Control Registers 6718
8.5.8 GPIO FSI Control Registers 6752
8.5.9 GPIO FSI Common Control Registers 6768
8.6 Hardware Synchronization Primitives (HSP) 6774
8.6.1 Overview 6774
8.6.1.1 Rationale 6774
8.6.1.2 Use Cases 6774
8.6.1.3 Shared Mailboxes 6775
8.6.1.4 Shared Semaphores 6775
8.6.1.5 Arbitrated Semaphores 6775
8.6.1.6 Doorbells 6775
8.6.2 Functional Description 6775
8.6.2.1 Sub-Unit Descriptions 6778

8.6.2.2 Multiple Processors Access to Shared Resources 6783
8.6.2.3 Using Doorbells for Handshake During Boot Sequence 6784
8.6.2.4 Shared Interrupts Configuration 6784
8.6.2.5 Programming Model 6785
8.6.3 Programming Guidelines 6785
8.6.3.1 Shared Mailboxes 6785
8.6.3.2 Shared Semaphores 6786
8.6.3.3 Arbitrated Semaphores 6787
8.6.3.4 Doorbells 6787
8.6.4 HSP Registers 6788
8.6.4.1 Common Registers 6788
8.6.4.2 HSP Interrupt Registers 6789
8.6.4.3 Shared Semaphore Registers 6793
8.6.4.4 Arbitrated Semaphore Registers 6794
8.6.4.5 Invalid Registers 6799
8.7 Design for Debugging (DFD) 6799
8.7.1 Overview 6799
8.7.1.1 System Connectivity Overview 6801
8.7.1.2 Summary of Major Features 6803
8.7.2 Functional Description 6803
8.7.2.1 CoreSight 6803
8.7.2.2 Closed Box Debug Over USB2 6868
8.7.2.3 Miscellaneous Debug Features 6909
8.7.2.4 Watchdog for Debug and Recovery 6929
8.7.2.5 Global Freezing for WDTs 6931
8.7.2.6 Debugging Low Power States 6939
8.7.3 DFD Registers 6941
8.7.3.1 CoreSight_CFG Registers 6941
8.7.3.2 Scratch Registers 6968
8.8 System Registers 6968
8.8.1 Miscellaneous Registers 6968
8.9 Always-On Cluster (AON) and SPE 6969
8.9.1 Overview 6969

8.9.1.1 AON Cluster Diagram 6970
8.9.2 AON Cluster Functional Description 6971
8.9.2.1 Cortex-R5 Processor 6971
8.9.2.2 Processor Configuration 6972
8.9.2.3 Interrupt Controller 6972
8.9.2.4 Processor and Interrupt Controller Documentation 6972
8.9.2.5 Timers 6972
8.9.2.6 TCM Interface 6973
8.9.2.7 ATCM Interface 6973
8.9.2.8 Error Checking and Signaling 6975
8.9.2.9 BTCM Interface 6975
8.9.2.10 BTCM RAM Configuration 6976
8.9.2.11 Error Checking and Signaling 6976
8.9.2.12 Interrupts 6976
8.9.2.13 AXI Block Support 6986
8.9.2.14 Interrupts 6987
8.9.2.15 Cortex-R5 Activity Monitor 6988
8.9.2.16 Programming Model 6990
8.9.3 AON Programming Guidelines 7001
8.9.3.1 Power Management 7001
8.9.3.2 AON Cluster SC7 Entry 7009
8.9.3.3 AON Cortex-R5 Activity Monitor 7009
8.9.3.4 Reset and Clocking 7010
9. I/O Controllers and Interfaces 7012
9.1 High-Speed I/O Cluster 7012
9.1.1 Overview 7012
9.1.1.1 HSIO UPHY Group 7012
9.1.1.2 NVHS UPHY Group 7015
9.1.1.3 GbE UPHY Group 7016
9.2 USB Complex 7018
9.2.1 Overview 7018
9.2.2 USB Functional Description 7020
9.2.2.1 USB Controller 7020

9.2.2.2 Interface Restrictions 7021
9.2.2.3 Host and Memory Access Interfaces 7022
9.2.2.4 USB 3.2 Device Controller 7022
9.2.2.5 xHCI Controller Programming Interface 7022
9.2.2.6 USB PADCTL 7023
9.2.2.7 USB AO 7024
9.2.3 USB Programming Guidelines 7024
9.2.3.1 Cold Boot 7024
9.2.3.2 SC7 (LP0) 7031
9.2.3.3 XUSB Controller Power Gating 7036
9.2.3.4 USB AO Programming 7050
9.2.3.5 USB 2 Pad Tracking Programming 7057
9.2.3.6 Battery Charging over XUSB 7058
9.2.3.7 Data Contact Detection 7059
9.2.3.8 Primary Charger Detection 7060
9.2.3.9 Secondary Charger Detection 7060
9.2.3.10 XUSB Programming Guidelines for Device Mode 7060
9.2.3.11 Recommended PHY Settings 7083
9.2.3.12 BIAS PAD Configuration 7083
9.2.3.13 Boot ROM Initialization Sequence for USB Recovery 7083
9.2.3.14 Performance Settings for USB Controllers 7085
9.2.3.15 USB Controller Handling of USB Resume Sequence 7085
9.2.4 USB Registers 7086
9.2.4.1 USB 3.0 Device Controller Memory-Mapped I/O Registers 7086
9.2.4.2 USB PADCTL and AO Registers 7136
9.2.4.3 XHCI Controller Configuration Registers 7186
9.2.4.4 XHCI Controller Memory-Mapped I/O Registers 7313
9.2.4.5 XUSB DEV Registers 7340
9.3 PCI Express (PCIe) Controller 7356
9.3.1 Overview 7356
9.3.1.1 Features 7356
9.3.1.2 External Connectivity Diagrams 7358
9.3.2 Interfaces 7361

9.3.2.1 System Interconnect Interfaces 7361
9.3.2.2 External I/O Interfaces 7362
9.3.3 Operation Models 7366
9.3.3.1 Normal Root Port Operation 7366
9.3.3.2 Normal Endpoint Operation 7366
9.3.4 Programming Guidelines 7367
9.3.4.1 Initialization 7367
9.3.4.2 PCIe Device Discovery and Enumeration 7371
9.3.4.3 PCIe Controller Hot Reset 7373
9.3.4.4 Interrupt and Message Handling 7373
9.3.4.5 PCIe Hierarchy Enumeration 7377
9.3.4.6 Partition Power-Gating 7378
9.3.4.7 SC7 7396
9.3.4.8 Fault Handling 7397
9.3.4.9 Shutdown and Reboot 7397
9.3.5 PCIe Registers 7397
9.3.5.1 PCIe iATU 7397
9.3.5.2 PCIe Root Port 7516
9.3.5.3 PCIe x1/x4/x8 Root Complex 7581
9.3.5.4 PCIe x4/x8 Endpoint 7918
9.4 Controller Area Network (CAN) 8188
9.4.1 Overview 8188
9.4.1.1 References 8189
9.4.1.2 Hardware Features 8189
9.4.1.3 Software Features 8190
9.4.2 Functional Description 8191
9.4.2.1 CAN Clock Domains 8191
9.4.2.2 CAN Resets 8192
9.4.2.3 External Time Stamp 8192
9.4.2.4 CAN Message RAM 8193
9.4.2.5 Round Robin Arbiter Between APB and CAN Core 8193
9.4.2.6 Configuration and Status Registers 8194
9.4.2.7 Time-Triggered (TT) CAN 8195

9.4.2.8 System Interfaces 8196
9.4.2.9 CAN SoC Interface 8196
9.4.3 Programming Guidelines 8196
9.4.3.1 Selecting Trigger Source for SWT and EVT 8197
9.4.3.2 Selecting Sources for Fault Detect Signal to HSM 8197
9.4.3.3 Orin Specific Controller Initialization 8198
9.4.3.4 Arbitration Between STD Message and EXT Message 8198
9.4.3.5 Prefetch Feature on Transmit Message Handler 8198
9.4.4 CAN Registers 8198
9.4.4.1 CAN Core Registers 8204
9.5 SDMMC Controller 8242
9.5.1 Overview 8242
9.5.2 SDMMC Controller Functional Description 8244
9.5.2.1 Supported Features 8244
9.5.2.2 Supported Specifications and Standards 8245
9.5.2.3 Hardware/Software Partitioning 8247
9.5.2.4 SDMMC Clocking 8250
9.5.2.5 Interface to Memory Controller 8251
9.5.2.6 Address Width 8251
9.5.2.7 SMMU Translation 8252
9.5.2.8 ADMA2 64 Bit 8252
9.5.2.9 Descriptors for 64-bit Addressing 8253
9.5.2.10 ADMA3 8254
9.5.2.11 Register Access 8255
9.5.2.12 Non-DMA Mode of Transfer (PIO mode) 8255
9.5.2.13 Standard Driver Support 8255
9.5.3 SDMMC Programming Guidelines 8256
9.5.3.1 Clocks and Reset 8256
9.5.3.2 Initialization 8257
9.5.3.3 SDMMC1 Initialization Sequence 8257
9.5.3.4 SDMMC4 Initialization Sequence 8263
9.5.3.5 General Guidelines 8268
9.5.3.6 ADMA2 8269

9.5.3.7 ADMA3 8269
9.5.3.8 Programming Guidelines for eMMC HS400 Mode 8271
9.5.3.9 Clock Trimmer Settings 8276
9.5.3.10 SDR50/SDR104/HS200 Tuning Procedure 8277
9.5.3.11 Boot ROM Support 8279
9.5.3.12 SD3.0 Signal Voltage Switching 8280
9.5.4 SDMMC Registers 8280
9.5.4.1 SDMMCAB Registers 8358
9.5.4.2 SDMMCA Registers 8463
9.6 I2C Controller (I2C) 8543
9.6.1 Overview 8543
9.6.1.1 Features 8543
9.6.1.2 Clocking 8544
9.6.2 Functional Description 8545
9.6.2.1 Second Level Clock Gating (SLCG) 8545
9.6.2.2 Bus Clear Master Operation 8546
9.6.2.3 Multi-Master Operation 8548
9.6.3 I2C Programming Guidelines 8556
9.6.3.1 I2C Frequency Divisor Register 8556
9.6.3.2 Programming Guidelines for Master in Normal Mode 8565
9.6.3.3 Programming Guidelines for Packet-Based Interface 8567
9.6.3.4 Master in Packet Mode Programming Guidelines 8573
9.6.3.5 Programming Guidelines for Slave in Byte Mode 8586
9.6.4 I2C Registers 8601
9.7 Universal Asynchronous Receiver/Transmitter (UART) 8637
9.7.1 Overview 8637
9.7.1.1 Features 8638
9.7.1.2 Security Features 8638
9.7.1.3 Hardware Signaling 8638
9.7.1.4 Clocking 8639
9.7.1.5 Use Cases 8641
9.7.1.6 Error Handling 8642
9.7.2 Programming Guidelines 8642

9.7.2.1 UART Register Access 8642
9.7.2.2 16450 Mode Programming 8643
9.7.2.3 16550 Mode Programming 8643
9.7.2.4 Transmitter and Receiver Holding Registers 8644
9.7.2.5 FIFO Interrupt Mode Operation 8645
9.7.2.6 FIFO Polled Mode Operation 8645
9.7.2.7 Programmable Baud Rate Generator 8646
9.7.2.8 Enable Register 8646
9.7.2.9 Interrupt Identification Register 8646
9.7.2.10 FIFO Control Register 8647
9.7.2.11 Line Control Register 8648
9.7.2.12 Modem Control Register 8648
9.7.2.13 Line Status Register 8648
9.7.2.14 Modem Status Register 8648
9.7.2.15 Scratchpad Register 8649
9.7.2.16 Auto-Baud Sense Register 8649
9.7.2.17 Baud Rate Generator 8650
9.7.2.18 SIR Pulse Encoder/Decoder 8650
9.7.2.19 UART Interrupts and Interrupt Service Routine 8651
9.7.2.20 FIFO Flushing Guidelines 8654
9.7.2.21 UART Boot 8655
9.7.2.22 IRDA_CSR Register (UART_IRDA_CSR_0) 8656
9.7.2.23 Pad and Pinmux Information 8657
9.7.3 UART Registers 8660
9.8 Server Base System Architecture (SBSA) UART 8671
9.8.1 Overview 8671
9.8.1.1 SBSA 8671
9.8.1.2 UART 8671
9.8.1.3 Feature List 8672
9.8.1.4 Interrupt and Status Generation 8674
9.8.1.5 UART DMA Interface 8675
9.8.1.6 Error Handling 8676
9.8.1.7 Clock and Reset 8677

9.8.2 Programming Guidelines 8679
9.8.2.1 Initial Sequence (Clock, Reset, and Pinmux Programming) 8679
9.8.2.2 Basic Guideline for Tx 8679
9.8.2.3 Basic Guidelines for Rx 8679
9.8.2.4 Tx: Non-FIFO Mode Programming 8680
9.8.2.5 Tx: FIFO Mode Programming (Non-DMA) 8681
9.8.2.6 Tx: FIFO Mode Programming (DMA Mode) 8682
9.8.2.7 Rx: Non-FIFO Mode Programming 8683
9.8.2.8 Rx: FIFO Mode Programming (Non-DMA) 8684
9.8.2.9 Rx: FIFO Mode Programming (DMA Mode) 8685
9.8.2.10 BREAK Programming (Non-FIFO Mode) in Tx 8686
9.8.2.11 BREAK Programming by Not Flushing FIFO Data (FIFO mode) in Tx 8687
9.8.2.12 BREAK Programming Flushing FIFO Data (FIFO mode) in Tx 8687
9.8.2.13 Loopback 8688
9.8.3 SBSA UART Registers 8688
9.9 Serial Peripheral Interface (SPI) 8700
9.9.1 Overview 8700
9.9.1.1 Standard and Compatibility 8701
9.9.1.2 Glossary 8701
9.9.1.3 Features 8702
9.9.2 Functional Description 8702
9.9.2.1 Transmission Format 8704
9.9.2.2 Modes of Operation 8706
9.9.2.3 Error Conditions 8715
9.9.2.4 Programming Guidelines 8717
9.9.3 SPI Registers 8722
9.10 Quad Serial Peripheral Interface (QSPI) 8738
9.10.1 Overview 8738
9.10.1.1 Standard and Compatibility 8738
9.10.1.2 Glossary 8739
9.10.1.3 Relevant Chapters in the TRM 8739
9.10.1.4 Features 8739
9.10.2 Functional Description 8740

9.10.2.1 Flash Devices Configurations 8740
9.10.2.2 Dual QSPI Bus Configuration 8742
9.10.2.3 System Configuration 8742
9.10.2.4 Drive Strength Programming 8743
9.10.2.5 Safety Mechanisms 8744
9.10.3 Programming Guidelines 8745
9.10.3.1 Clock Initialization and Control 8745
9.10.3.2 Clock Programming for Various QSPI Modes 8745
9.10.3.3 PLL Clocking Policy 8746
9.10.3.4 QSPI Controller Configuration 8746
9.10.3.5 PIO Data Transfer Using CPU 8747
9.10.3.6 DMA Data Transfer 8747
9.10.3.7 Programming Trimmers 8748
9.10.3.8 Auto-Drive Strength Calibration Sequence 8749
9.10.4 QSPI Registers 8749
9.11 Pulse Width Modulator (PWM) 8767
9.11.1 Overview 8767
9.11.1.1 PWM Diagrams 8767
9.11.1.2 Use Cases 8769
9.11.2 Programming Guidelines 8769
9.11.3 PWM Registers 8770
9.12 Fan Tachometer 8771
9.12.1 Overview 8771
9.12.2 Functional Description 8773
9.12.2.1 Fan Tachometer Clocks 8773
9.12.3 Programming Guidelines 8777
9.12.3.1 Interrupt Handling 8778
9.12.4 Fan Tachometer Registers 8778

1. Revision History

Document ID: DP-10508-002

Version	Date	Description of Change
1.0	Mar 24, 2022	Initial release
1.1	Mar 7, 2023	See "About this Release" and "Change Log" for details about changes and updates.
1.2	Sep 29, 2023	See "About this Release" and "Change Log" for details about changes and updates.

1.1 About This Release

Introduction

- Getting Started
- About this Release

Memory Architecture and Memory Mapped I/O

- Memory Subsystem (MSS) Registers
- Address Map
- Address Space Translation (AST)
- General Purpose Direct Memory Access (DMA) Engines
- System Memory Management Unit (SMMU)

Boot and Power Management

- Boot and Power Management Processor (BPMP)

CPU Complex

GPU

Multimedia Complex

- Host Controller
- MIPI Camera Serial Interface (CSI) **UPDATED**
- Video Input (VI)

- Video Image Compositor (VIC)
- High Definition Audio (HDA)
- Display Controller
- Consumer Electronics Control (CEC)
- Audio Processing Engine (APE) **UPDATED**
- Always On Digital Microphone (AODMIC)
- Pixel Memory Formats

System Components

- Clock and Reset Controller (CAR)
- Interrupt Controllers
- Timers
- Multi-Purpose I/O Pins and Pin Multiplexing (PinMux) **UPDATED**
- GPIO Controllers
- Hardware Synchronization Primitives (HSP)
- Design for Debugging (DFD) **UPDATED**
- System Registers
- Always-On Cluster (AON) and SPE **NEW**

I/O Controllers and Interfaces

- High-Speed I/O Cluster
- USB Complex **UPDATED**
- PCI Express (PCIe) Controller **UPDATED**
- Controller Area Network (CAN)
- SDMMC Controller **UPDATED**
- I2C Controller (I2C)
- Universal Asynchronous Receiver/Transmitter (UART)
- Server Based System Architecture (SBSA) UART
- Serial Peripheral Interface (SPI)
- Quad Serial Peripheral Interface (QSPI)
- Pulse Width Modulator (PWM) **UPDATED**
- Fan Tachometer

1.2 Change Log

This change log identifies technical changes and/or significant modification to already released materials, it does not include editorial changes made for readability.

All Sections - Ongoing

Correcting KB notation where appropriate; updating to KiB. This follows the IEEE and NIST convention using an 'i' to indicate the binary convention, and its absence to indicate decimal: i.e., 1 KiB is 210 or 1,024 bytes, and 1 KB is 103 or 1,000 bytes.

1.2.1 20230929 (Version 1.2)

Multimedia Complex

- MIPI Camera Serial Interface (CSI); Changed VI 5.0 and ISP 5.0 both to 6.0
- APE Registers; updates to the following register files:
  - Audio Crossbar (AXBAR) Registers
  - Digital Microphone Control (DMIC) Registers
  - Audio De-Multiplexer (ADX) Control Registers
  - Audio Miscellaneous (AMISC Control Registers
  - Audio Generic Interrupt Control (AGIC) Registers

System Components

- Multi-Purpose I/O Pins and Pin Multiplexing (PinMux) Registers
  - G3 PAD Control Registers; Updated note to use Block Name (PADCTL_A0) to find the Base Address of these registers in the System Address Map
  - UFS PAD Control Registers; Updated note to use Block Name (PADCTL_A17) to find the Base Address of these registers in the System Address Map
- Design for Debugging (DFD); Updates to FSI TSC section: added FSI_CSITE_TSINTP Render Configuration, Connectivity, Calculating Time-Interval Using Timestamps subsections
- Always-On Cluster (AON) and SPE; new chapter

I/O Controllers and Interfaces

- USB Complex; Changed PADCTRL_G2_CTL_ to PADCTRL_EDP_
- PCIe Registers (PCIe Root Port); Updated PCIE_RP_APPL_DEBUG_0: added encoding definition to SMLH_LTSSM_STATE
- SDMMC Controller; Removed Boot ROM support
- Pulse Width Modulator (PWM); PinMux naming changed to soc_gpio37 for PWM8 interface in Table 1.1

2. Getting Started

2.1 Overview

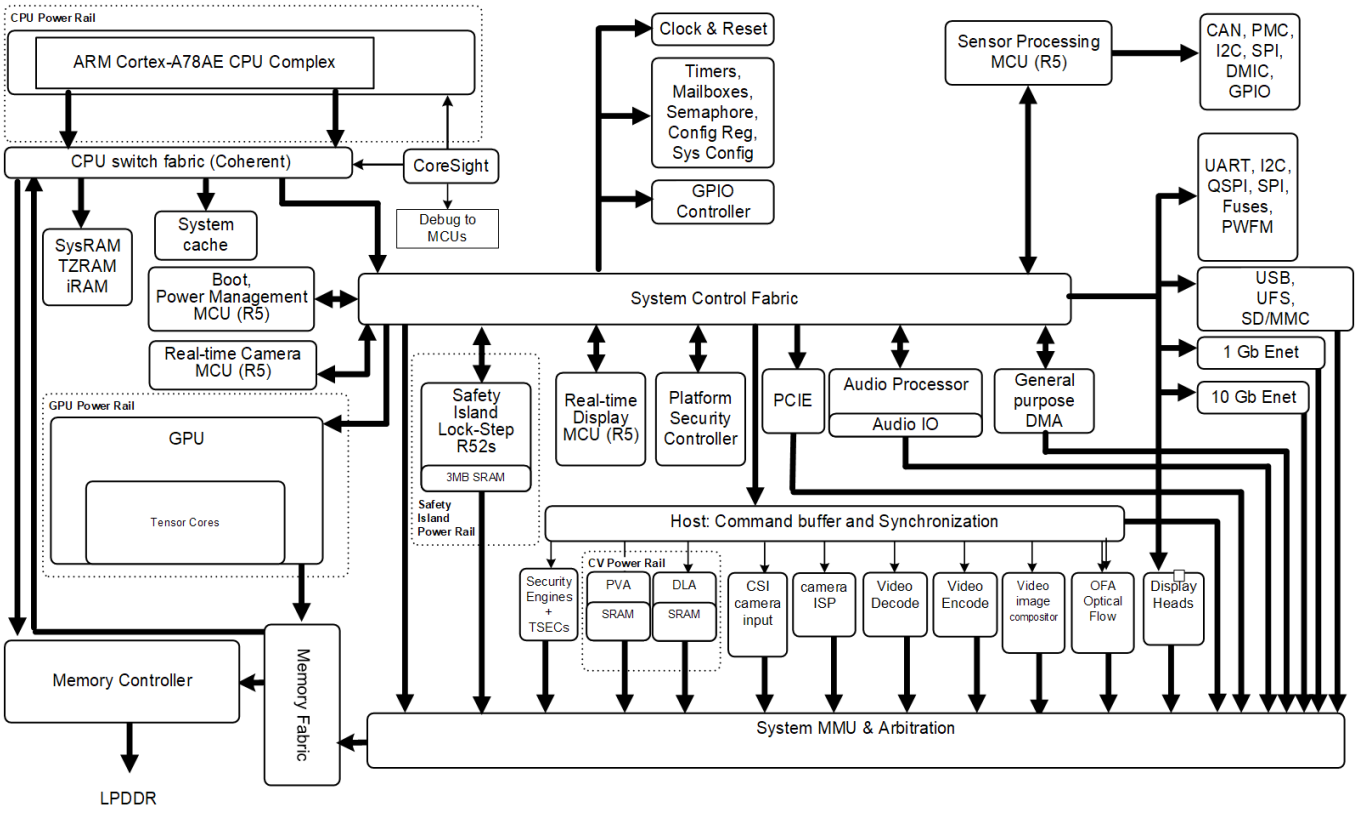
This Technical Reference Manual (TRM) describes how the NVIDIA® Orin™ series system-on-chip (SoC) functions; and is a guide for writing, understanding, and modifying code that controls those functions. It contains functional descriptions of how the Orin hardware works, and also describes the registers and related programming interfaces.

Not all the units in Orin are described in detail here; as some, like the GPU, are only intended to be used with NVIDIA supplied drivers. This document is intended to provide deep technical detail where appropriate, users should first refer to the datasheets and other published information for an overview of Orin.

Refer to the datasheet for supported features and speeds. This document may describe hardware features and functionality not currently supported on specific Orin product SKUs, and may describe clock speeds or data rates that cannot be achieved on all SKUs. Refer to the appropriate Orin product data sheet as this is the primary authority for supported features, functionality, and operating speeds of a particular product SKU.

Refer to software documentation for supported features. This document may describe hardware features not supported by NVIDIA software. Description of a hardware capability in this document does not imply software support for that function. Refer to the appropriate BSP or software release notes for detailed information about currently supported software functionality.

Figure 2.1 Orin Series SoC Block Diagram



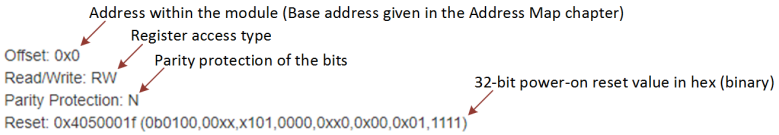
2.1.1 Reading Register Tables

Every register table has an address line followed by a table containing the bit descriptions for that register. The address line contains:

- **Offset:** the address of the register within the specific module. Refer to the system memory map for the start address of the module; apply the offset at the top of the table to get the register address.
- **Read/Write:** the register access type. If a register table contains the R/W column, individual bits within the register will have different R/W properties. If there is no R/W column, all bits in that register have the same R/W property. Values are RO (read only), R/W (read/write), WO (write only, and RWC (read/write to clear).
- **Parity Protection:** per register parity diagnostic in hardware implemented by safety critical IPs. The parity diagnostic detects random faults in the register fields.
  'Y' indicates that the field is continuously checked by the parity diagnostic of the register. Single bit flips in the register, if not intended, are reported as a fault.
  'N' indicates that the field is not checked by the parity diagnostic of the register.
  If all fields of the register have Parity Protection column as 'N', the register does not implement any hardware based parity diagnostic.

- **Reset:** gives the power-on reset value in 32-bit binary. A value of x implies that the register bit has an undefined value at reset. A hexadecimal value is listed for convenience, where appropriate.
- **Default:** only displayed when the default setting is different from the Reset value.

Unspecified bits may not appear in tables (see example below). Unspecified bits should be written with their Reset values, while reads return an unknown value.



Bit	Reset	Description
31	0x0	**PIO**: Programmable IO. Program this bit to 1, after all the other bits in the QSPI Command Register and QSPI Command Register 2 are programmed to start the transfer. Hardware clears this bit automatically after the transfer is done. Clearing of this bit by Software will stop the shifter and latch the partial data into the buffer (in Receive Mode). **0** = STOP **1** = PIO
30	0x1	**M_S**: **0** = Reserved **1** = Master Mode (internal clock) (default)
29:28	0x0	**Mode**: The QSPI controller need be programmed according to the device it is communicating with. Only Master Mode 0 is supported. **0** = Mode 0 (for SDR and DDR) **1** = RSVD **2** = RSVD **3** = RSVD

Access type different among fields;
a separate R/W column in the table specifies the fields' access types.

Offset: 0x14
Read/Write: See table below.
Parity Protection: N
Reset: 0x00400005 (0bxx00,0000,0100,0000,00xx,xxx0,0000,0101)

Bit	R/W	Reset	Description
29:23	RO	0x0	**RX_FIFO_FULL_COUNT**: Indicates the number of slots in the receive FIFO remaining before the FIFO is empty. This field is used by Software for debugging purposes.
22:16	RO	0x40	**TX_FIFO_EMPTY_COUNT**: Indicates the number of slots in the transmit FIFO remaining before the FIFO is full. This field is used by Software for debugging purposes.
15	RW	0x0	**RX_FIFO_FLUSH**: Software writes a 1 to this bit to flush the Rx FIFO. This bit reads as 1 when the flush operation is in progress and returns to 0 when it is finished. **0** = NOP **1** = FLUSH
14	RW	0x0	**TX_FIFO_FLUSH**: Software writes a 1 to this bit to flush the Tx FIFO. This bit reads as 1 when the flush operation is in progress and returns to 0 when it is finished. **0** = NOP **1** = FLUSH

Unspecified bits (31:30, 13:0 as in this example) are treated as "0" in the hex and "x" in the binary representation of the Reset value.

Parity Protection different among fields;
a separate Parity Protection column in the table specifies the fields' parity protection.

Offset: 0x2030
Read/Write: See table below.
Parity Protection: See table below.
Reset: 0x00010000 (0bxxxx,xxxx,xxxx,0001,xxxx,xxxx,0000,0000)

Bit	R/W	Parity Protection	Reset	Description
19:16	RW	Y	0x1	**THOST_MLOCK_VM**
7:2	RW	Y	0x0	**THOST_MLOCK_CH**
1	RO	N	0x0	**THOST_MLOCK_TZLOCKED**
0	RO	N	0x0	**THOST_MLOCK_LOCKED**

Offset: 0x10
Read/Write: R/W
Parity Protection: Y
SCR Protection: USECSCR_0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xxx1)

Register is a Protected Register (PR) by the Security Control Register (SCR) as specified.

Bit	Reset	Description
5:4	0x0	**MBS**: STSC Bit Select, STSC[1+MBS] edge used as reference when not free running.
0	0x1	**FR**: Free Running , controls the microsecond counter TSC lock behavior **0** = the microsecond locks the usec reference pulse to a selected bit of STSC[4:1] using the adjust signal (recommended mode). **1** = Usec Counter counts on each edge of OSC in free running fashion (legacy mode).

2.1.2 Units

This TRM follows the IEEE and NIST conventions for multiplying prefixes.

Among other things, this convention uses an 'i' to indicate the binary convention, and its absence to indicate decimal. So, 1 KiB is 2^{10} or 1,024 bytes, and 1 KB is 10^3 or 1,000 bytes. Similarly it uses:

- Mi for 2^{20} and M for 10^6
- Gi for 2^{30} and G for 10^9
- Ti for 2^{40} and T for 10^{12}

2.1.3 Glossary

This glossary is intended to cover the acronyms used in this document; along with some others related to the Arm SoC world. Many other acronyms in this document are in broad engineering use and are not documented here.

Term	Definition
32K or 32k	32,768 Hz oscillator clock. Any references to 32k, 32 kHz, 32 KHz, or 32K in the context of this clock should always be interpreted as referring to a frequency of 32,768 Hz.
444/422/420	Pixel storage formats. 444 refers to formats where there are equal amounts of information for all three elements, 422 and 420 refer to formats where the color difference information is stored at half-resolution in either one direction or both.
ADAS	Advanced Driving Assistance System.
ADSP	Audio DSP. Refers to the Cortex-A9 processor in the APE.
ADX	Audio Demultiplexer. Part of the Audio Hub used to demultiplex multiple audio streams.
AMBA	Advanced Microcontroller Bus Architecture. A set of standard buses defined by ARM.
AMX	Audio Multiplexer. Part of the Audio Hub used to multiplex multiple audio streams together.
AO or AON	Always-On power domain. This part of the chip is always powered on, even in the deepest sleep state, except complete processor shutdown. See also SPE.
AON I/O rails	Inputs/Outputs in VDDIO_SYS, VDDIO_AO, and VDDIO_AO_HV serving the AON cluster logic
AONPG	Always-On Non-Power Gated. Used to indicate when an AO partition has no power gating implemented.
AOPG	Always-On Power Gated. Used to indicate whether an AO partition has power gating implemented.
AOPM	Always-On Cluster Power Management Module. Power management module implements/manages the Cortex-R5 power state transitions. AOPM manages power state transitions for the SPE Cortex-R5 and its subsystem.
AOTAG	Always-On Thermal Alarm Generator.

Term	Definition
AOVC	Always-On Voltage Controller.
AP	Applications Processor. Refers to the Orin device, means the same as SoC.
APB	AMBA Peripheral Bus. Arm defined simple 32-bit single master bus for peripheral devices.
APE	Audio Processing Engine.
Aperture	A named region of address space.
APS	Auxiliary processor subsystem. Refers to an implementation boundary that is shared across Cortex-R5 clusters on the chip.
ARM	Arm is a company provides the CPU architectural specification for Orin. Arm is also Architecture Reference Manual, as in (the second Arm of) ARM ARM, which defines the CPU architecture.
AST	Address Space Translator. A subunit associated with some of the embedded Arm cores that does address translation from local addresses to system addresses along with appending some AXI attributes.
AUTOSAR	Automotive Open System Architecture (an OS used in automotives)
AXI	AMBA Advanced eXtensible Interface. A more advanced bus than AHB defined as part of AMBA 3 by ARM.
AVIC	Arm PL192 Vectored Interrupt Controller. Used as the Cortex-R5 interrupt controller for all the Cortex-R5 processors (SPE, SCE, and BPMP).
Bayer	A type of image sampling pattern invented by Dr. Bryce E. Bayer of Eastman Kodak. The pattern consists of quads of pixels with two green samples, one red sample, and one blue sample.
BCT	Boot Configuration Table. Stored on external boot device, contains config parameters for boot decisions.
BIT	Boot Information Table. Maintained internally by Boot ROM in RAM for boot path/error tracking and logging.
BKV	Best Known Value. Configurations determined by system characterization.
BOM	Bottom of Memory. Refers to the lowest address in an address map.
BPMP	Boot and Power Management Processor.
BPMP-FW	BPMP Firmware. This refers to the power management firmware that would be executed on BPMP (post-boot). This is also stored in external boot media.
BR	Boot ROM. Power-on start boot code. Stored/burnt in chip IROM, executes from BPMP.
Brick	Input/output interface block with analog and other special functions.
CAR	Clock and Reset module. Controls clocks and resets to the various parts of Orin.
CBB	Control Backbone.
CCPLEX	CPU complex (i.e., CPU subsystem).
CDE	Color Decompression Engine.

Term	Definition
CEC	Consumer Electronics Control. A part of the HDMI interface specification used for sending device control commands, often from a remote control.
CID	Client ID.
CIL	Control and Interface Logic.
Cold boot	The SoC partition power transitions from OFF to ON with no previous state available. Software must construct all states from scratch. Boot ROM is executed. DRAM is brought on-line.
CoT	Chain of trust. A security term used to denote any code that is trusted because it is loaded securely from the root of trust (Boot ROM).
CP	Color Parser.
C-PHY	A MIPI standard physical layer that can carry CSI data. Clocks are transmitted along with data; data lanes are three wires.
CPU	CPU generally refers to the main CPUs unless specified otherwise.
CRC	Cyclic Redundancy Check.
CSI	MIPI Camera Serial Interface. A standard high-speed serial interface for connecting cameras to Orin.
CUDA	Compute Unified Device Architecture.
CVC	Central Voltage Controller.
CV Cluster	Computer Vision Cluster. A partition in Orin that includes NVDLA, PVA, and CVNAS.
CVNAS	Computer Vision NoC and SRAM.
CVNOC	A subblock within CVNAS–the NoC portion of CVNAS.
CVSRAM	A subblock within CVNAS–the memory storage portion of CVNAS.
CZ	Controlled-output impedance MPIO pads.
DBC	Dead Battery Charging.
DBP	Dead Battery Provisions.
DBB	Data Back-Bone.
DCLS	Dual Core Lock Step, a technique used for functional safety where two processing cores receive the same inputs, and the outputs are compared to detect errors. Usually one of the cores is delayed with respect to the other.
DDA	Digital Differential Analyzer. A technique commonly used in graphics for interpolation of variables over an interval between start and end point, and also applied to other problems.
DDIC	Display Driver Integrated Circuit
Deep Sleep	See SC7.
DFD	Design for Debug
DFS	Dynamic Frequency Scaling

Term	Definition
DFT	Design for Test
dGPU or DGPU	Discrete GPU. Refers to an attached GPU that is external to the Orin SoC, as opposed to the internal GPU.
DLA	Deep Learning Accelerator
DLS	Delayed Lock-Step
DMIC	Digital microphone interface. Supports direct attach of PDM microphones.
DPD	Deep Power Down. A mode in which the pad can tolerate VDD_CORE being turned off.
D-PHY	A MIPI standard physical layer that can carry CSI data. Clocks are transmitted separately from data; data lanes are two wires and clock lanes are two wires.
DRCM	Debug Recovery Mode (also known as RCM-exit-to-JTAG).
DSI	MIPI Display Serial Interface. A standard high-speed serial interface for connecting displays to Orin.
DVC	Dynamic Voltage Controller block.
DVFS	Dynamic Voltage and Frequency Scaling.
EAVB	Ethernet Audio Video Bridging. Includes extension of the Ethernet standard supporting real-time streaming; more recently referred to as Time-Sensitive Networking.
EC	Error Collator
ECC	Error Correction Code.
eDP	Embedded DisplayPort™.
EMC	External Memory Controller. A block that interfaces with external DDR/LPDDR devices.
EOF	End of Frame. Refers to the last non-cropped long packet in a frame, or to an ISP EOF packet.
EVP	Exception Vector Pointer.
FA	Failure Analysis.
FCM	Full Custom Macro.
FE	Frame End. Refers to the NVCSI short packet.
FIQ	Fast Interrupt Request.
FMEA	Failure Mode and Effects Analysis.
FMON	Frequency Monitoring logic.
FS	Frame Start. Refers to the NVCSI short packet.
FSI	Functional Safety Island
FSM	Finite State Machine. This is a hardware engineering phrase used to describe a hardware block that controls the operation of some logic function.

Term	Definition
GIC	Generic Interrupt Controller. Normally used to describe an Arm supplied interrupt controller used for a specific set of processors.
GPIO	General Purpose Input/Output. An I/O signal uncommitted to a specific role and controlled by software.
GTE	Generic Timestamping Engine
HDMI	High-Definition Multimedia Interface. A digital connection carrying video and audio at high speed over a single connector.
HDR	High Dynamic Range. Usually a reference to cameras or displays using a higher dynamic range for pixel values.
HSIO	High-Speed I/O Interfaces. See the corresponding chapter of this TRM for more details.
HSM	Hardware Safety Manager.
HV	Hypervisor
HVC	Hardware Vmin Control. The hardware initiated flow to enter/exit the Vmin state on VDD_CPU. Now referred to as CC3.
ICG	Internal Clock Gate
iGPU or IGPU	Internal GPU. Refers to the GPU within the Orin SoC.
IRAM	Internal RAM used by the boot process until DRAM is configured; now deprecated, and replaced by TCMs and SysRAM.
IROM	Internal chip ROM which contains the Boot ROM code and data.
IPI	Inter-Processor Interrupt.
IPT	Inverse Perspective Transform.
IRQ	Interrupt Request.
ISP	Image Signal Processor. A hardware engine that is part of the camera processing pipeline.
ISR	Interrupt Service Routine
JTAG	Joint Test Action Group Standard for Test Access Port and Boundary Scan Architecture. A serial bus used to external devices used for debug and testing.
KMD	Kernel mode driver.
LDC	Lens Distortion Correction
LIC	"Legacy" interrupt controller, a central interrupt controller in Orin.
LP1	Low Power 1 state. Devices are power-gated, SoC clock domains are set to the minimum frequency (12 MHz and 38.4 MHz), the flow controller is configured to monitor "LP1 exit wake events," DRAM is put in self-refresh, and the VDD_CPU rail is powered off. Also known as the Suspend state.
LSIO	Low-Speed I/O
LV	Low-voltage MPIO pads.

Term	Definition
MB1 and MB2	Microboot stages 1 and 2, stored on external boot media. Refer to the Boot chapter for more details.
MC	Memory Controller. Handles requests from internal clients and arbitrates among them to allocate memory bandwidth. Also referred to as MSS.
MCA	Machine Check Architecture
MCCIF or MC-CIF	Memory Controller Client Interface. The standard interface block between the memory controller subsystem fabric and the client device. Note that some modules may have multiple client interfaces.
MDMM	Multi-drop multi-merge.
MIPI	The Mobile Industry Processor Interface. An industry alliance promoting a number of standard interfaces for mobile devices.
MMIO	Memory-Mapped I/O (transactions).
MODS	Modular Diagnostic Software. It is a powerful software program that allows users to test the NVIDIA hardware. MODS is used for three primary purposes–chip and board functional validation, chip and board failure analysis and debug, and architectural verification.
MPCORE	Multiprocessor CPU core. A generic term for a CPU capable of operating as part of an SMP group.
M-PHY or MPHY	MIPI M-PHY. An embedded-clock serial-interface technology with high bandwidth capabilities.
MPIO	Multi-purpose Input/Output.
MSI	Message Signaled Interrupt.
MSS	Memory Subsystem. Refer to the corresponding chapter.
MTS	An alternate name for the Carmel CPU Microcode generated by Dynamic Code Optimization.
NOC	Network On Chip (referring to the backbone architecture of the SoC)
NVDEC	NVIDIA Video Decoder engine.
NVENC	NVIDIA Video Encoder engine.
NVJPG	NVIDIA JPEG engine.
NVM	Non-Volatile memory. Data retained even after power is turned off. All boot media have non-volatile memory storage.
OD	Open Drain MPIO pads.
OGL	Open Graphics Library (also known as OpenGL). An API supported on Orin and accelerated in hardware by dedicated 3D and 2D engines.
PA	Physical Address.
Partition	A physical sub-region of the Orin device. Power gating is usually applied at the partition level.
PCIe	Peripheral Component Interconnect Express. A high-speed interface for connecting to external devices.
PG	Power Gating; Power Gate-able

Term	Definition
PMIC	Power Management Integrated Control (synonymous with PMU).
PMC	Power Management Controller. Controls the various power management features in the system.
PMIC	Power management IC. Off-die module that controls various voltage regulators, provides the 32 kHz (32.768 kHz) clock source and provides the main system reset to the SoC.
POR	Power On Reset.
PPC	Pixels Per Clock.
PPI	Private Peripheral Interrupt within an Arm processor core.
PWFM	Pulse Width Frequency Modulation. Generates programmed pulse widths typically used to control backlight in display panels.
PVA	Programmable Vision Accelerator. Custom computer vision DSP, Orin has two of them.
PVT	Process, Voltage, and Temperature.
R5	Cortex-R5 is a mid-range ARMv7 CPU cluster used for multiple engines in the SoC.
RAZ	Read As Zero.
RCE	Real-time Camera-control Engine. See the corresponding chapter.
RCM	Recovery Mode. Used for re-flashing the external boot device image.
RGB	Name given to pixels with red, green, and blue color components. This is the pixel format typically found in most display technologies because each color component corresponds to the colors of the filters or phosphors used in the display device.
RISC	Reduced Instruction Set Computer. The CPU architecture used by Arm CPUs.
RMMI	Reference M-PHY Module Interface.
R/O	Read only.
RTC	Real Time Clock (as in VDD_RTC)
R/W	Read write.
SATA	Serial Advanced Technology Attachment (ATA).
SC7	Low power 0 state in which DRAM is put in self-refresh. The system state is saved in the PMC and in DRAM. VDD_CORE and VDD_CPU rails are powered off, and PMC is configured to monitor "LP0 wake events" that trigger LP0 exit. Also known as Deep-Sleep state.
SCE	Safety Cluster Engine. See the corresponding chapter.
SCF	System Coherency Fabric.
SCR	Security Control Register.
SE	Security Engine. Used for hardware acceleration of authentication and decryption steps.
SGI	Software Generated Interrupt.

Term	Definition
SDMMC	SD and MMC Controller. An I/O controller supporting both the SD/SDIO interface standards and the eMMC standard.
SFIO	Special Function Input/Output
SLCG	Second Level Clock Gating. A hardware technique to reduce power.
SLINK	Serial Link. A legacy and now obsolete name for the SPI controller.
SM	Security Master
SMMU	System Memory Management Unit. Block within the memory controller used to map from a virtual address space to physical addresses for device DMA.
SMP	Symmetric Multi-Processing.
SNIC	System Network InterConnect. Used to refer to the control fabric in SoC.
SoC	System on a Chip. An integrated circuit containing a CPU, memory controller and the peripheral devices needed for a computing system.
SOF	Start of Frame. Used to refer to the first non-cropped long packet in a frame, or to an ISP SOF packet.
SOR	Serial Output Resource. SOR is GPU IP for driving HDMI/DP/LVDS. It converts the output of the display to a more modern high-speed serial protocol. DSI is not included since it's not GPU IP based.
S/PDIF	Sony/Philips Digital Interconnect Format.
SPE	Sensor Processing Engine. See the Always-On Cluster chapter.
SPI	(a) Serial Peripheral Interface Bus. A synchronous serial data link, that operates in full-duplex mode. (b) A Shared Peripheral Interrupt within an Arm core.
ST	Standard MPIO pads.
Sub-aperture	An aperture whose address space is a subset of another aperture and whose accessibility is also a subset of another aperture.
SysRAM	Name for the shared on-chip memory.
TBSA	Trusted Base System Architecture (an Arm specification).
TCM	Tightly Coupled Memory. This refers to internal local RAM that is associated with some of the Arm CPU Cores. These are used as local, low-latency scratch pad memory.
Tegra	The name formerly used to describe the NVIDIA family of SoCs, and now used only in certain applications. References to Tegra that still exist in this TRM, such as Tegra Host or Tegra pixel formats, may be considered to apply to all the NVIDIA family of SoCs, up to and including Orin.
THI	Host Interface.
TNR	Temporal Noise Reduction.
TOM	Top of Memory. Refers to the highest address in an address map.
TOS	Trusted OS.

Term	Definition
TS	TimeStamp
TSC	Timestamp System Counter.
TSEC	Security co-processor. An embedded security processor used mainly to manage the HDCP encryption and keys on the HDMI link.
TSOSC	Ring oscillator based thermal sensors.
TZ	TrustZone® is a secure operating environment of the Arm CPU architecture and the related secure parts of the SoC backbone and devices.
TZRAM	TrustZone secured RAM on the SoC.
UFS	Universal Flash Storage.
UFSHC	UFS Host Controller.
Uncore	The CPU related logic outside of the CPU processing core itself.
UniPro	MIPI Unified Protocol, a link layer communication protocol.
U-PHY or UPHY	Universal PHY, and NVIDIA reference to a multi-mode Serializer-Deserializer (SerDes) with analog pads for high-speed signaling to support various protocols.
VA	Virtual address.
VDD_CORE	SoC power rail.
VDD_CPU	CPU power rail.
VDD_RTC	Always-On power rail.
vGPIO	Virtual General-Purpose Input/Output.
vGIC	Virtualization capable Generic Interrupt Controller
VI	Video Input. The acronym used to describe the Orin block used for camera and related pixel input functions.
VIC	(a) Video Image Compositor. A SoC block that implements video post-processing functions needed by a video playback application to produce the final image for the player window. (b) The Arm name for the PL192 Vectored Interrupt Controller; used alongside the Cortex-R5 cores. Also referred to here as AVIC to avoid a name-space conflict with the previous entry.
VM	Virtual Machine
VPR	Video Protect Region.
Warm boot	Exit from SC7 state.
WDT	Watchdog Timer. These timers can generate interrupts or resets to attempt to break the AP out of undesirable states.
WFE	Wait For Event (an Arm instruction).
WFI	Wait For Interrupt (an Arm instruction).

Term	Definition
WID	Write ID (from the AXI specification).
W1C	Write as 1 to Clear.
xHCI	eXtensible Host Controller Interface for USB.
XIP	eXecute In Place. Debug-only scheme where the primary IROM code is bypassed, and external code is fetched instead at reset, to test out Boot ROM code. This code is executed in-place from the external device, i.e., executed per instruction without fetching the entire code chunk.
YCbCr	An alternative pixel representation that can take advantage of the properties of the human psycho-perceptual vision system and store the color difference information with lower spatial resolution. It consists of a luminance channel Y and two color difference signals Cb and Cr. See the definition of 444/422/420.
YUV	See YCbCr. U and V are equivalent to Cb and Cr, respectively.
ZSL	Zero Shutter Lag.

3. Memory Architecture and Memory Mapped I/O

3.1 MSS Registers

The Memory Subsystem (MSS) is controlled by NVIDIA provided software, and so is not documented in detail here. However, some registers are exposed to support customers, and these are listed below.

MC_EMEM_ADR_CFG_CHANNEL_ENABLE_0

External memory address configuration channel select mask configures the routing of requests between the two memory channels. The channel select mask is ANDed with the address of a transaction. The resulting value is XORd to a single bit, which is used to select the channel for the transaction Boot requirements:

- coldboot - This register should be parameterized in the BCT and written by the BootROM during coldboot.
- warmboot - This register should be saved in the scratch registers and restored by the BootROM during warmboot.

Mask bits [10:9] select single channel vs. dual-channel modes and 512B vs. 1KB interleave as follows, each with its own physical address to <channel, device,="" row,="" bank,="" column=""> mapping.

```
Interleave      MASK[10:9]    Remarks
Single channel     2'b00
512B               2'bx1      Channels alternate on (most) 512B boundaries
1KB                2'b10      Channels alternate on (most) 1KB boundaries
```

While decoding the DRBC data from the address: if mask bit 9 is set then the 9th bit of address is dropped for DRBC decoding; and else if mask bit 10 is set then the 10th bit address is dropped for address decoding.

Write access to this register is controlled by the EMEM_CFG_ACCESS_CTRL_0 register </channel,>.

Offset: 0xdf8
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,1111)

Bit	Reset	Description
15:0	0xf	**EMEM_CHANNEL_ENABLE**: [PMC_SECURE] Selects which MC channels are enabled for normal read/writes

MC_EMEM_ADR_CFG_CHANNEL_MASK_0

Write access to this register is controlled by the EMEM_CFG_ACCESS_CTRL register

Offset: 0x60
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x6db66200 (0b0110,1101,1011,0110,0110,001x,xxxx,xxxx)

Bit	Reset	Description
31:9	0x36db31	**EMEM_CHANNEL_MASK**: [PMC_SECURE] Mask is ANDed with address and the resulting value is XORd to a single bit, giving bit 0 of the channel index

MC_EMEM_ADR_CFG_CHANNEL_MASK_1_0

Write access to this register is controlled by the EMEM_CFG_ACCESS_CTRL register

Offset: 0xdfc
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x6db66400 (0b0110,1101,1011,0110,0110,010x,xxxx,xxxx)

Bit	Reset	Description
31:9	0x36db32	**EMEM_CHANNEL_MASK_1**: [PMC_SECURE] Mask is ANDed with address and the resulting value is XORd to a single bit, giving bit 1 of the channel index

MC_EMEM_ADR_CFG_CHANNEL_MASK_2_0

Write access to this register is controlled by the EMEM_CFG_ACCESS_CTRL register

Offset: 0xdf4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x6db66800 (0b0110,1101,1011,0110,0110,100x,xxxx,xxxx)

Bit	Reset	Description
31:9	0x36db34	**EMEM_CHANNEL_MASK_2**: [PMC_SECURE] Mask is ANDed with address and the resulting value is XORd to a single bit, giving bit 2 of the channel index

MC_EMEM_ADR_CFG_CHANNEL_MASK_3_0

Write access to this register is controlled by the EMEM_CFG_ACCESS_CTRL register

Offset: 0xdf0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x6db67000 (0b0110,1101,1011,0110,0111,000x,xxxx,xxxx)

Bit	Reset	Description
31:9	0x36db38	**EMEM_CHANNEL_MASK_3**: [PMC_SECURE] Mask is ANDed with address and the resulting value is XORd to a single bit, giving bit 2 of the channel i ndex

3.2 Address Map

3.2.1 Overview

This chapter defines the system Address Map (AMAP).

The term “Address” reflects the Physical Address as seen by the main CPU complex, unless specified otherwise. The SoC has several subsystems such as the BPMP, SPE, Audio, etc. The processors in these subsystems may have a different view of the System AMAP. The AMAP specifications for these subsystems are explained in more detail in the corresponding subsystem section of this document.

The term AMAP implies System Address Map or Global Address Map (with the terms System Address Map and Global Address Map being used interchangeably), unless otherwise specified.

3.2.1.1 Features

- **64 KiB Alignment**

  Arm Architecture recommends aligning all peripheral address ranges along the MMU page sizes. This enables each device to occupy a single entry in the Page Table and makes it possible to uniquely identify and describe device access attributes. The page can be uniquely classified under a non-normal memory type such as nGnRnE (for SO) or nGnRE (for DEV). Refer to the ARMv8 Architecture Reference Manuals for more information. Page-alignment of peripherals also improves security across virtualized guest Operating Systems. Armv8 supports three types of page sizes in its MMU: 4 KiB, 16 KiB, and 64 KiB. By aligning with 64 KiB, all three implementations are supported and sufficient MMIO for each device is allocated.

- **40-bit Address Map**

  The SoC supports a one Terabyte AMAP (1 TiB or 40 bits of addressing).

- **PCIe Aperture**

  The SoC offers two apertures for PCIe: one for a 32-bit address OS and the other for a greater-than 32-bit address OS. Each PCIe controller is provided with a 32 MiB aperture. The PCIe aperture for greater-than 32-bit address OS handles situations with large AMAP requirements, which cannot be addressed by drivers running on a 32-bit address OS.

3.2.1.2 AMAP and Aperture

The SoC supports a 1-Terabyte AMAP (40 bits). See below for a high-level overview of the AMAP.

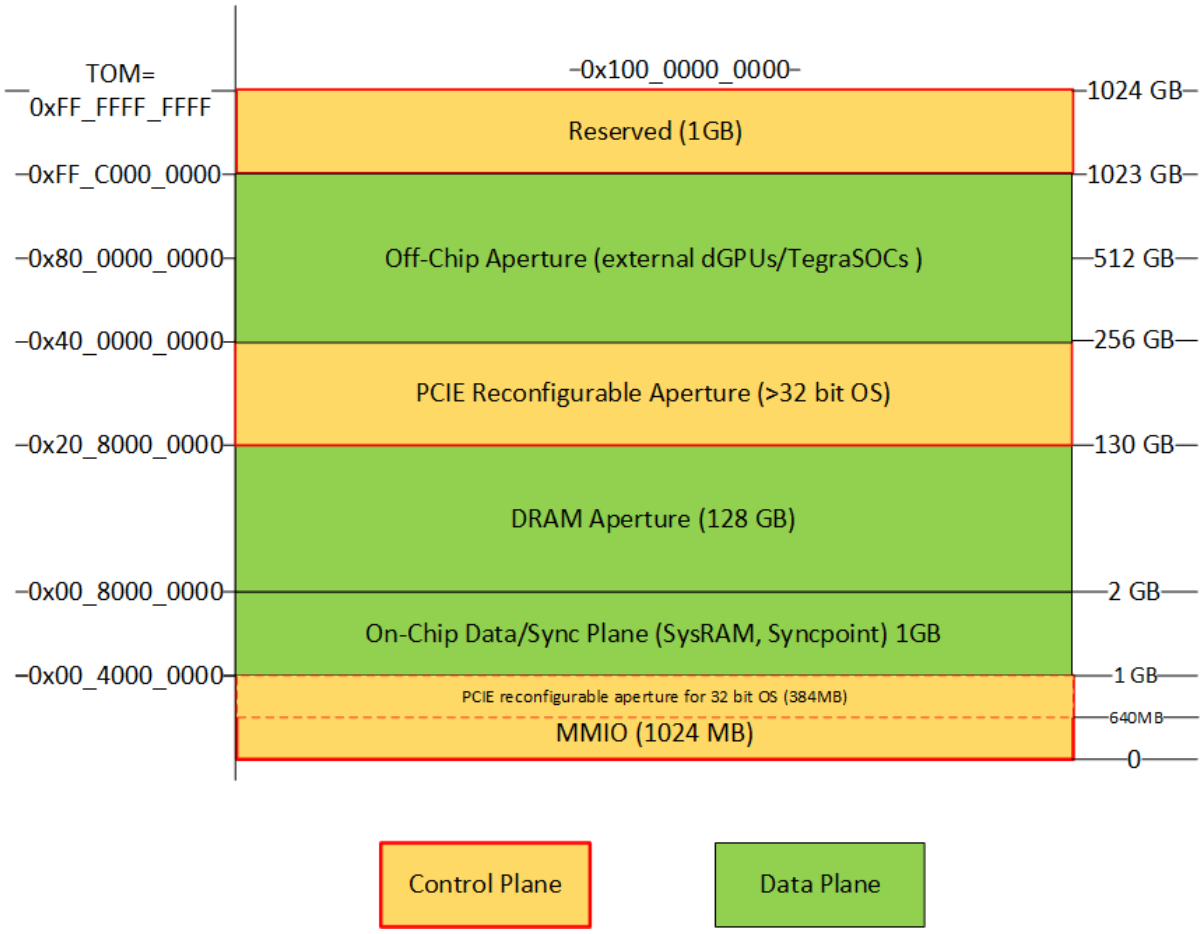
Table 3.1 AMAP Overview

Name	Address Range
Reserved (1023 GiB - 1024 GiB)	0xFF_C000_0000 - 0xFF_FFFF_FFFF
Off-Chip Aperture (256 GiB - 1023 GiB)	0x40_0000_0000 - 0xFF_BFFF_FFFF
PCIe Reconfigurable Aperture for > 32-bit OS (130 GiB - 256 GiB)(1)	0x20_8000_0000 - 0x3F_FFFF_FFFF
DRAM Aperture (2 GiB - 130 GiB)	0x00_8000_0000 - 0x20_7FFF_FFFF
On-Chip Data/Sync Plane (1 GiB - 2 GiB)	0x00_4000_0000 - 0x00_7FFF_FFFF
PCIe Reconfigurable Aperture for 32-bit OS (640 MB - 1 GiB)	0x00_3000_0000 - 0x00_3FFF_FFFF

Name	Address Range
MMIO Aperture (0 GiB - 640 MB)	0x00_0000_0000 - 0x00_3FFF_FFFF

1. The control plane extends beyond 4 GiB for PCIe controllers controlled by > 32-bit OS.

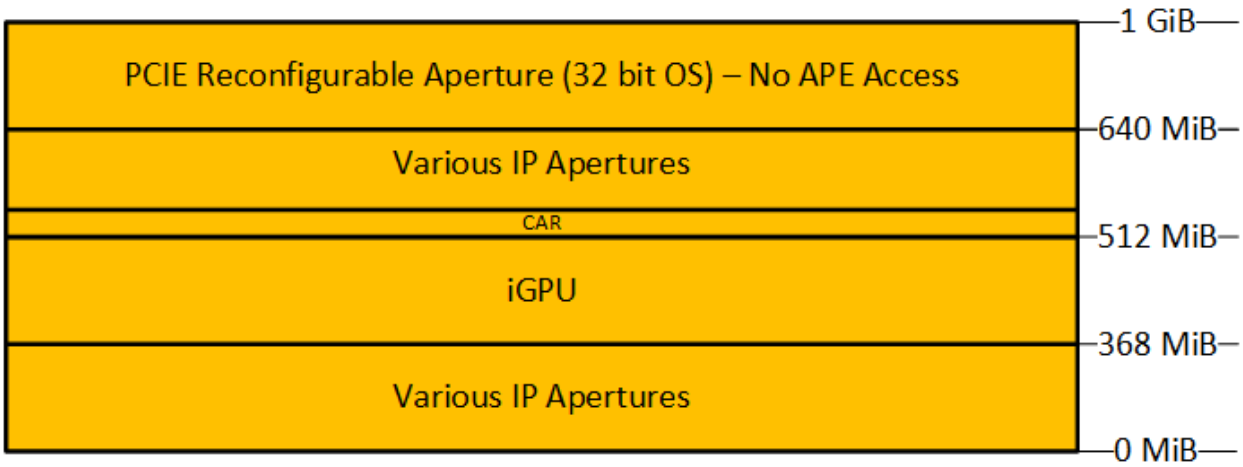
Figure 3.1 AMAP Diagram



3.2.1.3 MMIO Aperture

The MMIO aperture begins at the bottom of memory (0x0) and extends to 1 GiB.

Figure 3.2 MMIO Aperture



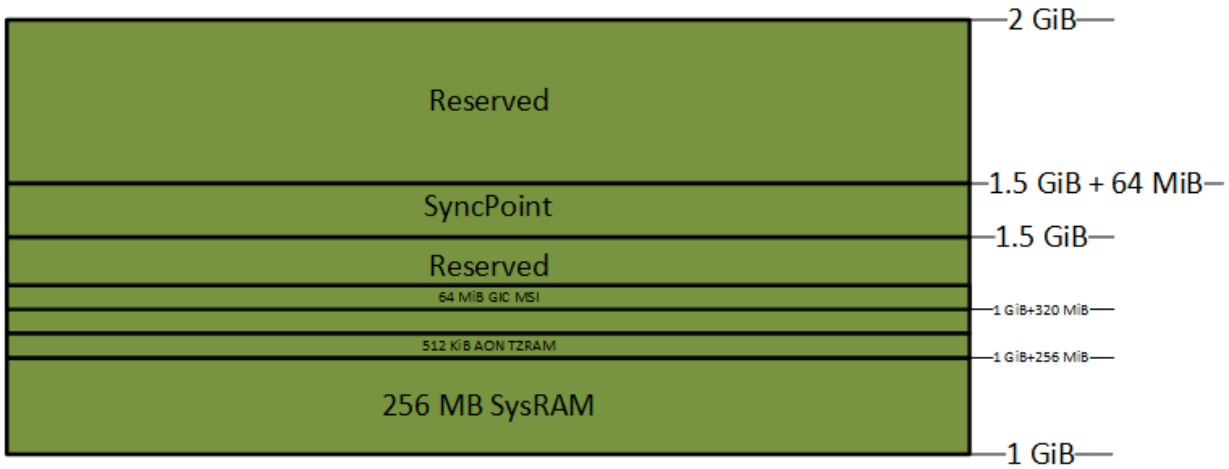
This region houses the following apertures:

- Configuration register apertures of all IPs of the SoC
- 144 MiB iGPU aperture
- PCIe aperture for 32-bit OS in its top 384 MiB. This is used for configuration, MMIO, IOIO space and is accessible by 32-bit OS.
  (Note: the APE only decodes apertures below 0.75 GiB to the control backbone, so it cannot access PCIe. Since the High Definition Audio (HDA) controller sits under the 0.75 GiB range, there is no need for APE to ever access any PCIe controller.)
  - Each PCIe controller is provided with a 32 MiB aperture.
  - PCIe APB configuration space is disjointed from this space and can live anywhere outside this range.

3.2.1.4 On-Chip Data/Sync Plane Aperture

The following figure shows the On-Chip Data/Sync Plane aperture.

Figure 3.3 On-Chip Data/Sync Plane Aperture



This region houses the following:

- The SysRAM region starting at 1 GiB. A 256 MiB region is reserved, although physical SysRAM Size is 512 KiB. 256 MiB allows for a higher steering granularity at the System Coherency Fabric (SCF) in the CCPLEX. All CPUs should access SysRAM at this physical location, without the need of any address translation.
- The 4 MiB Compute Vision SRAM (CV-SRAM), used by Programmable Vision Accelerators (PVAs) and Deep Learning Accelerators. A 256 MiB region is reserved for this similarly to SysRAM.
- The dGPU Host Controller Sync Point aperture. This is a 64 MiB Sync Point Region that is used to convert the dGPU's semaphores into Host Controller sync points. Note that this region can also be used by any non Host Controller clients. This region shadows the Sync Point RAM in Host Controller using a sideband interface between Host Controller and Memory. Previously, the GPU needed a sideband interface into the Host Controller Sync Point to talk to any other Host Controller client. With this Sync Point region, the sideband with Host Controller is removed from the GPU.
- Reserved region beyond the dGPU Sync Point aperture.

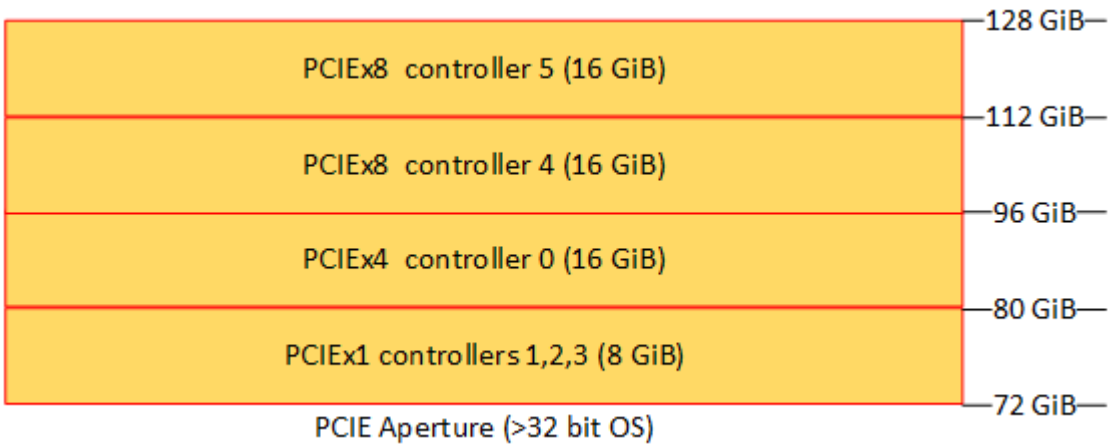
3.2.1.5 DRAM Aperture

The DRAM aperture is used for the Physical Address of the off-chip local DRAM. The AMAP supports up to 128 GiB of DRAM. A 32-bit OS can access the lower 2 GiB Physical DRAM location (AMAP region from 2 GiB to 4 GiB). In order to access DRAM above 4 GiB in the AMAP, a 32-bit OS must use an additional address translation capability like AST or the SMMU.

3.2.1.6 PCIe Aperture (32-bit OS)

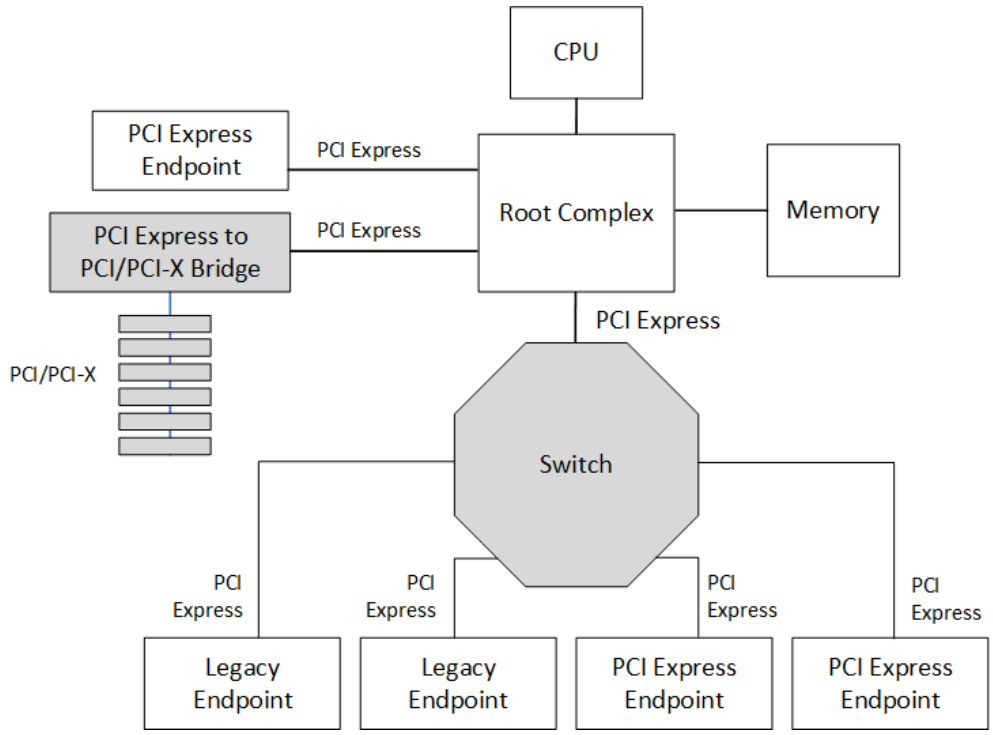
The SoC has two apertures for PCIe: one for 32-bit OS (a sub-aperture within the MMIO aperture) and the other for >32-bit OS (see figure below).

Figure 3.4 PCIe Aperture (32-bit OS)



The PCIe aperture for a 64-bit OS handles situations with large AMAP requirements. When a PCIe controller is connected to a device that has multiple functions, where each function can in turn be a switch, it presents large AMAP requirements. Such large AMAP requirements cannot be met under 4 GiB and hence cannot be addressed by drivers running on 32-bit CPUs.

Figure 3.5 PCIe Controller Connection Example



The SoC platforms can present a similar large AMAP requirement. PCIe x8/x4 controllers are used to connect SoC devices to dGPUs or NVSwitch. There are also three PCIe x1 controllers that present a 1 GiB per controller requirement. An 8 GiB space is reserved for the PCIe x1 controllers to allow for future expansion.

3.2.1.7 Off-Chip Aperture

This configurable (BOM/TOM registers in SCF and MCF) aperture allows the AMAP to be adjusted to the required platform. This region can be used to access

- Peer DRAM Dual SoC platforms
- dGPUs VIDMEM in SoC with Discrete GPU platforms
- Other dGPUs VIDMEMs and other DRAMs in Single/Multiple SoC with Multiple dGPUs platform

3.2.2 System Address Map

The locality column indicates if apertures are only accessible by some masters. For example:

- SYSTEM - indicates access is possible from all Initiators unless the paths from a master to slave is not physically present.
- CCPLEX - indicates access is only possible from CCPLEX, the main CPU complex.
- SYSTEM_CFG - same as SYSTEM. In addition, when an IP can exist in various modes, this indicates the exact register bit that is used to decide the mode.
  For example, the PCIe controller can exist as a Root Complex or as an End Point. And for each mode, the header file is different.

Block Name	Address Start	Address End	Address Locality
MMIO	0x00000000	0x3fffffff	SYSTEM
LOVEC	0x00000000	0x0000ffff	CCPLEX
APE_ADSP_EVP	0x00000000	0x00001fff	APE
MISC	0x00100000	0x0017ffff	SYSTEM
TOP2_HSP	0x01600000	0x0168ffff	SYSTEM
TOP2_HSP_COMMON	0x01600000	0x0160ffff	SYSTEM
TOP2_HSP_SM	0x01610000	0x0164ffff	SYSTEM
TOP2_HSP_SM_0_1	0x01610000	0x0161ffff	SYSTEM
TOP2_HSP_SM_2_3	0x01620000	0x0162ffff	SYSTEM
TOP2_HSP_SM_4_5	0x01630000	0x0163ffff	SYSTEM
TOP2_HSP_SM_6_7	0x01640000	0x0164ffff	SYSTEM
TOP2_HSP_SS	0x01650000	0x0168ffff	SYSTEM
TOP2_HSP_SS_0	0x01650000	0x0165ffff	SYSTEM
TOP2_HSP_SS_1	0x01660000	0x0166ffff	SYSTEM
TOP2_HSP_SS_2	0x01670000	0x0167ffff	SYSTEM
TOP2_HSP_SS_3	0x01680000	0x0168ffff	SYSTEM
TSA_0	0x02000000	0x0207ffff	SYSTEM
TSA_NODE_0	0x02000000	0x02000fff	SYSTEM
TSA_NODE_1	0x02001000	0x02001fff	SYSTEM

Block Name	Address Start	Address End	Address Locality
TSA_NODE_2	0x02002000	0x02002fff	SYSTEM
TSA_NODE_3	0x02003000	0x02003fff	SYSTEM
TSA_NODE_4	0x02004000	0x02004fff	SYSTEM
TSA_NODE_5	0x02005000	0x02005fff	SYSTEM
TSA_NODE_6	0x02006000	0x02006fff	SYSTEM
TSA_NODE_7	0x02007000	0x02007fff	SYSTEM
TSA_NODE_8	0x02008000	0x02008fff	SYSTEM
TSA_NODE_10	0x0200a000	0x0200afff	SYSTEM
TSA_NODE_13	0x0200d000	0x0200dfff	SYSTEM
TSA_NODE_14	0x0200e000	0x0200efff	SYSTEM
TSA_NODE_16	0x02010000	0x02010fff	SYSTEM
TSA_NODE_18	0x02012000	0x02012fff	SYSTEM
TSA_NODE_21	0x02015000	0x02015fff	SYSTEM
TSA_NODE_24	0x02018000	0x02018fff	SYSTEM
TSA_NODE_25	0x02019000	0x02019fff	SYSTEM
TSA_NODE_26	0x0201a000	0x0201afff	SYSTEM
TSA_NODE_27	0x0201b000	0x0201bfff	SYSTEM
TSA_NODE_29	0x0201d000	0x0201dfff	SYSTEM
TSA_NODE_30	0x0201e000	0x0201efff	SYSTEM
TSA_NODE_33	0x02021000	0x02021fff	SYSTEM
TSA_NODE_36	0x02024000	0x02024fff	SYSTEM
TSA_NODE_37	0x02025000	0x02025fff	SYSTEM
TSA_NODE_39	0x02027000	0x02027fff	SYSTEM
TSA_NODE_40	0x02028000	0x02028fff	SYSTEM
TSA_NODE_42	0x0202a000	0x0202afff	SYSTEM
TSA_NODE_44	0x0202c000	0x0202cfff	SYSTEM
TSA_NODE_45	0x0202d000	0x0202dfff	SYSTEM
TSA_NODE_46	0x0202e000	0x0202efff	SYSTEM
TSA_NODE_48	0x02030000	0x02030fff	SYSTEM

Block Name	Address Start	Address End	Address Locality
TSA_NODE_52	0x02034000	0x02034fff	SYSTEM
TSA_NODE_53	0x02035000	0x02035fff	SYSTEM
TSA_NODE_57	0x02039000	0x02039fff	SYSTEM
TSA_NODE_58	0x0203a000	0x0203afff	SYSTEM
TSA_NODE_60	0x0203c000	0x0203cfff	SYSTEM
TSA_NODE_61	0x0203d000	0x0203dfff	SYSTEM
TSA_NODE_63	0x0203f000	0x0203ffff	SYSTEM
TSA_NODE_65	0x02041000	0x02041fff	SYSTEM
TSA_NODE_67	0x02043000	0x02043fff	SYSTEM
TSA_NODE_68	0x02044000	0x02044fff	SYSTEM
TSA_NODE_72	0x02048000	0x02048fff	SYSTEM
TSA_NODE_75	0x0204b000	0x0204bfff	SYSTEM
TSA_NODE_77	0x0204d000	0x0204dfff	SYSTEM
TSA_NODE_78	0x0204e000	0x0204efff	SYSTEM
TSA_NODE_81	0x02051000	0x02051fff	SYSTEM
TSA_NODE_83	0x02053000	0x02053fff	SYSTEM
TSA_NODE_85	0x02055000	0x02055fff	SYSTEM
TSA_NODE_90	0x0205a000	0x0205afff	SYSTEM
TSA_NODE_96	0x02060000	0x02060fff	SYSTEM
TSA_NODE_97	0x02061000	0x02061fff	SYSTEM
TSA_NODE_98	0x02062000	0x02062fff	SYSTEM
TSA_NODE_99	0x02063000	0x02063fff	SYSTEM
TSA_NODE_100	0x02064000	0x02064fff	SYSTEM
TSA_NODE_101	0x02065000	0x02065fff	SYSTEM
TSA_NODE_104	0x02068000	0x02068fff	SYSTEM
TSA_NODE_106	0x0206a000	0x0206afff	SYSTEM
TSA_NODE_109	0x0206d000	0x0206dfff	SYSTEM
TSA_NODE_110	0x0206e000	0x0206efff	SYSTEM
TSA_NODE_117	0x02075000	0x02075fff	SYSTEM

Block Name	Address Start	Address End	Address Locality
TSA_NODE_120	0x02078000	0x02078fff	SYSTEM
TSA_NODE_123	0x0207b000	0x0207bfff	SYSTEM
TOP_TKE	0x02080000	0x021bffff	SYSTEM
TMR_SHARED	0x02080000	0x0208ffff	SYSTEM
TOP_TKE_TMR	0x02090000	0x0218ffff	SYSTEM
TMR0	0x02090000	0x0209ffff	SYSTEM
TMR1	0x020a0000	0x020affff	SYSTEM
TMR2	0x020b0000	0x020bffff	SYSTEM
TMR3	0x020c0000	0x020cffff	SYSTEM
TMR4	0x020d0000	0x020dffff	SYSTEM
TMR5	0x020e0000	0x020effff	SYSTEM
TMR6	0x020f0000	0x020fffff	SYSTEM
TMR7	0x02100000	0x0210ffff	SYSTEM
TMR8	0x02110000	0x0211ffff	SYSTEM
TMR9	0x02120000	0x0212ffff	SYSTEM
TMR10	0x02130000	0x0213ffff	SYSTEM
TMR11	0x02140000	0x0214ffff	SYSTEM
TMR12	0x02150000	0x0215ffff	SYSTEM
TMR13	0x02160000	0x0216ffff	SYSTEM
TMR14	0x02170000	0x0217ffff	SYSTEM
TMR15	0x02180000	0x0218ffff	SYSTEM
TOP_TKE_WDT	0x02190000	0x021bffff	SYSTEM
WDT0	0x02190000	0x0219ffff	SYSTEM
WDT1	0x021a0000	0x021affff	SYSTEM
WDT2	0x021b0000	0x021bffff	SYSTEM
GPIO_CTL	0x02200000	0x022fffff	SYSTEM
GPIO_CTL_COMMON	0x02200000	0x0220ffff	SYSTEM
GPIO_CTL_COMMON_GPIO0	0x02200000	0x02200fff	SYSTEM

Block Name	Address Start	Address End	Address Locality
GPIO_CTL_COMMON_GPIO1	0x02201000	0x02201fff	SYSTEM
GPIO_CTL_COMMON_GPIO2	0x02202000	0x02202fff	SYSTEM
GPIO_CTL_COMMON_GPIO3	0x02203000	0x02203fff	SYSTEM
GPIO_CTL_COMMON_GPIO4	0x02204000	0x02204fff	SYSTEM
GPIO_CTL_COMMON_GPIO5	0x02205000	0x02205fff	SYSTEM
GPIO_CTL0	0x02210000	0x0221ffff	SYSTEM
GPIO_CTL0_GPIO0	0x02210000	0x02210fff	SYSTEM
GPIO_CTL0_GPIO1	0x02211000	0x02211fff	SYSTEM
GPIO_CTL0_GPIO2	0x02212000	0x02212fff	SYSTEM
GPIO_CTL0_GPIO3	0x02213000	0x02213fff	SYSTEM
GPIO_CTL0_GPIO4	0x02214000	0x02214fff	SYSTEM
GPIO_CTL0_GPIO5	0x02215000	0x02215fff	SYSTEM
GPIO_CTL1	0x02220000	0x0222ffff	SYSTEM
GPIO_CTL1_GPIO0	0x02220000	0x02220fff	SYSTEM
GPIO_CTL1_GPIO1	0x02221000	0x02221fff	SYSTEM
GPIO_CTL1_GPIO2	0x02222000	0x02222fff	SYSTEM
GPIO_CTL1_GPIO3	0x02223000	0x02223fff	SYSTEM
GPIO_CTL1_GPIO4	0x02224000	0x02224fff	SYSTEM
GPIO_CTL1_GPIO5	0x02225000	0x02225fff	SYSTEM
GPIO_CTL2	0x02230000	0x0223ffff	SYSTEM
GPIO_CTL2_GPIO0	0x02230000	0x02230fff	SYSTEM
GPIO_CTL2_GPIO1	0x02231000	0x02231fff	SYSTEM
GPIO_CTL2_GPIO2	0x02232000	0x02232fff	SYSTEM
GPIO_CTL2_GPIO3	0x02233000	0x02233fff	SYSTEM
GPIO_CTL2_GPIO4	0x02234000	0x02234fff	SYSTEM
GPIO_CTL2_GPIO5	0x02235000	0x02235fff	SYSTEM
GPIO_CTL3	0x02240000	0x0224ffff	SYSTEM

Block Name	Address Start	Address End	Address Locality
GPIO_CTL3_GPIO0	0x02240000	0x02240fff	SYSTEM
GPIO_CTL3_GPIO1	0x02241000	0x02241fff	SYSTEM
GPIO_CTL3_GPIO2	0x02242000	0x02242fff	SYSTEM
GPIO_CTL3_GPIO3	0x02243000	0x02243fff	SYSTEM
GPIO_CTL3_GPIO4	0x02244000	0x02244fff	SYSTEM
GPIO_CTL3_GPIO5	0x02245000	0x02245fff	SYSTEM
GPIO_CTL4	0x02250000	0x0225ffff	SYSTEM
GPIO_CTL4_GPIO0	0x02250000	0x02250fff	SYSTEM
GPIO_CTL4_GPIO1	0x02251000	0x02251fff	SYSTEM
GPIO_CTL4_GPIO2	0x02252000	0x02252fff	SYSTEM
GPIO_CTL4_GPIO3	0x02253000	0x02253fff	SYSTEM
GPIO_CTL4_GPIO4	0x02254000	0x02254fff	SYSTEM
GPIO_CTL4_GPIO5	0x02255000	0x02255fff	SYSTEM
GPIO_CTL5	0x02260000	0x0226ffff	SYSTEM
GPIO_CTL5_GPIO0	0x02260000	0x02260fff	SYSTEM
GPIO_CTL5_GPIO1	0x02261000	0x02261fff	SYSTEM
GPIO_CTL5_GPIO2	0x02262000	0x02262fff	SYSTEM
GPIO_CTL5_GPIO3	0x02263000	0x02263fff	SYSTEM
GPIO_CTL5_GPIO4	0x02264000	0x02264fff	SYSTEM
GPIO_CTL5_GPIO5	0x02265000	0x02265fff	SYSTEM
GPIO_CTL6	0x02270000	0x0227ffff	SYSTEM
GPIO_CTL6_GPIO0	0x02270000	0x02270fff	SYSTEM
GPIO_CTL6_GPIO1	0x02271000	0x02271fff	SYSTEM
GPIO_CTL6_GPIO2	0x02272000	0x02272fff	SYSTEM
GPIO_CTL6_GPIO3	0x02273000	0x02273fff	SYSTEM
GPIO_CTL6_GPIO4	0x02274000	0x02274fff	SYSTEM
GPIO_CTL6_GPIO5	0x02275000	0x02275fff	SYSTEM
GPIO_CTL7	0x02280000	0x0228ffff	SYSTEM
GPIO_CTL7_GPIO0	0x02280000	0x02280fff	SYSTEM

Block Name	Address Start	Address End	Address Locality
GPIO_CTL7_GPIO1	0x02281000	0x02281fff	SYSTEM
GPIO_CTL7_GPIO2	0x02282000	0x02282fff	SYSTEM
GPIO_CTL7_GPIO3	0x02283000	0x02283fff	SYSTEM
GPIO_CTL7_GPIO4	0x02284000	0x02284fff	SYSTEM
GPIO_CTL7_GPIO5	0x02285000	0x02285fff	SYSTEM
HDACODEC	0x0242c000	0x0242cfff	SYSTEM
PADCTL_A	0x02430000	0x0244ffff	SYSTEM
PADCTL_A0	0x02430000	0x02430fff	SYSTEM
PADCTL_A2	0x02432000	0x02432fff	SYSTEM
PADCTL_A4	0x02434000	0x02434fff	SYSTEM
PADCTL_A5	0x02435000	0x02435fff	SYSTEM
PADCTL_A6	0x02436000	0x02436fff	SYSTEM
PADCTL_A7	0x02437000	0x02437fff	SYSTEM
PADCTL_A8	0x02438000	0x02438fff	SYSTEM
PADCTL_A11	0x0243b000	0x0243bfff	SYSTEM
PADCTL_A13	0x0243d000	0x0243dfff	SYSTEM
PADCTL_A16	0x02440000	0x02440fff	SYSTEM
PADCTL_A17	0x02441000	0x02441fff	SYSTEM
PADCTL_A20	0x02444000	0x02444fff	SYSTEM
PADCTL_A21	0x02445000	0x02445fff	SYSTEM
PADCTL_A24	0x02448000	0x02448fff	SYSTEM
PADCTL_A25	0x02449000	0x02449fff	SYSTEM
I2S7	0x02450000	0x0245ffff	SYSTEM
I2S8	0x02460000	0x0246ffff	SYSTEM
MPHY_L0	0x02470000	0x0247ffff	SYSTEM
MPHY_L1	0x02480000	0x0248ffff	SYSTEM
MISC_ERR_COLLATOR	0x024e0000	0x024effff	SYSTEM
PROTO	0x024f0000	0x024fffff	SYSTEM
GPCDMA	0x02600000	0x0280ffff	SYSTEM

Block Name	Address Start	Address End	Address Locality
GPCDMA_COMMON_0	0x02600000	0x0260ffff	SYSTEM
GPCDMA_CH0	0x02610000	0x0261ffff	SYSTEM
GPCDMA_CH1	0x02620000	0x0262ffff	SYSTEM
GPCDMA_CH2	0x02630000	0x0263ffff	SYSTEM
GPCDMA_CH3	0x02640000	0x0264ffff	SYSTEM
GPCDMA_CH4	0x02650000	0x0265ffff	SYSTEM
GPCDMA_CH5	0x02660000	0x0266ffff	SYSTEM
GPCDMA_CH6	0x02670000	0x0267ffff	SYSTEM
GPCDMA_CH7	0x02680000	0x0268ffff	SYSTEM
GPCDMA_CH8	0x02690000	0x0269ffff	SYSTEM
GPCDMA_CH9	0x026a0000	0x026affff	SYSTEM
GPCDMA_CH10	0x026b0000	0x026bffff	SYSTEM
GPCDMA_CH11	0x026c0000	0x026cffff	SYSTEM
GPCDMA_CH12	0x026d0000	0x026dffff	SYSTEM
GPCDMA_CH13	0x026e0000	0x026effff	SYSTEM
GPCDMA_CH14	0x026f0000	0x026fffff	SYSTEM
GPCDMA_CH15	0x02700000	0x0270ffff	SYSTEM
GPCDMA_CH16	0x02710000	0x0271ffff	SYSTEM
GPCDMA_CH17	0x02720000	0x0272ffff	SYSTEM
GPCDMA_CH18	0x02730000	0x0273ffff	SYSTEM
GPCDMA_CH19	0x02740000	0x0274ffff	SYSTEM
GPCDMA_CH20	0x02750000	0x0275ffff	SYSTEM
GPCDMA_CH21	0x02760000	0x0276ffff	SYSTEM
GPCDMA_CH22	0x02770000	0x0277ffff	SYSTEM
GPCDMA_CH23	0x02780000	0x0278ffff	SYSTEM
GPCDMA_CH24	0x02790000	0x0279ffff	SYSTEM
GPCDMA_CH25	0x027a0000	0x027affff	SYSTEM
GPCDMA_CH26	0x027b0000	0x027bffff	SYSTEM
GPCDMA_CH27	0x027c0000	0x027cffff	SYSTEM

Block Name	Address Start	Address End	Address Locality
GPCDMA_CH28	0x027d0000	0x027dffff	SYSTEM
GPCDMA_CH29	0x027e0000	0x027effff	SYSTEM
GPCDMA_CH30	0x027f0000	0x027fffff	SYSTEM
GPCDMA_CH31	0x02800000	0x0280ffff	SYSTEM
APE	0x02900000	0x02affffff	SYSTEM
AHUB	0x02900000	0x0291ffff	SYSTEM
AXBAR	0x02900800	0x02900fff	SYSTEM
I2S1	0x02901000	0x029010ff	SYSTEM
I2S2	0x02901100	0x029011ff	SYSTEM
I2S3	0x02901200	0x029012ff	SYSTEM
I2S4	0x02901300	0x029013ff	SYSTEM
I2S5	0x02901400	0x029014ff	SYSTEM
I2S6	0x02901500	0x029015ff	SYSTEM
SFC1	0x02902000	0x029021ff	SYSTEM
SFC2	0x02902200	0x029023ff	SYSTEM
SFC3	0x02902400	0x029025ff	SYSTEM
SFC4	0x02902600	0x029027ff	SYSTEM
AMX1	0x02903000	0x029030ff	SYSTEM
AMX2	0x02903100	0x029031ff	SYSTEM
AMX3	0x02903200	0x029032ff	SYSTEM
AMX4	0x02903300	0x029033ff	SYSTEM
ADX1	0x02903800	0x029038ff	SYSTEM
ADX2	0x02903900	0x029039ff	SYSTEM
ADX3	0x02903a00	0x02903aff	SYSTEM
ADX4	0x02903b00	0x02903bff	SYSTEM
DMIC1	0x02904000	0x029040ff	SYSTEM
DMIC2	0x02904100	0x029041ff	SYSTEM
DMIC3	0x02904200	0x029042ff	SYSTEM
DMIC4	0x02904300	0x029043ff	SYSTEM

Block Name	Address Start	Address End	Address Locality
DSPK1	0x02905000	0x029050ff	SYSTEM
DSPK2	0x02905100	0x029051ff	SYSTEM
SPDIF1	0x02906000	0x029061ff	SYSTEM
AFC1	0x02907000	0x029070ff	SYSTEM
AFC2	0x02907100	0x029071ff	SYSTEM
AFC3	0x02907200	0x029072ff	SYSTEM
AFC4	0x02907300	0x029073ff	SYSTEM
AFC5	0x02907400	0x029074ff	SYSTEM
AFC6	0x02907500	0x029075ff	SYSTEM
OPE1	0x02908000	0x029083ff	SYSTEM
OPE1_COMMON	0x02908000	0x029080ff	SYSTEM
OPE1_PEQ	0x02908100	0x029081ff	SYSTEM
OPE1_MBDRC	0x02908200	0x029083ff	SYSTEM
SPKPROT1	0x02908c00	0x02908fff	SYSTEM
MVC1	0x0290a000	0x0290a1ff	SYSTEM
MVC2	0x0290a200	0x0290a3ff	SYSTEM
AHC	0x0290b900	0x0290baff	SYSTEM
MIXER1	0x0290bb00	0x0290c2ff	SYSTEM
IQC1	0x0290e000	0x0290e1ff	SYSTEM
IQC2	0x0290e200	0x0290e3ff	SYSTEM
ARAD	0x0290e400	0x0290e7ff	SYSTEM
ADMAIF	0x0290f000	0x0290ffff	SYSTEM
ASRC	0x02910000	0x02911fff	SYSTEM
APE_ADMA	0x02930000	0x0297ffff	SYSTEM
APE_ADMA_GLOBAL	0x02930000	0x0293ffff	SYSTEM
APE_ADMA_PAGE1	0x02940000	0x0294ffff	SYSTEM
APE_ADMA_PAGE2	0x02950000	0x0295ffff	SYSTEM
APE_ADMA_PAGE3	0x02960000	0x0296ffff	SYSTEM
APE_ADMA_PAGE4	0x02970000	0x0297ffff	SYSTEM

Block Name	Address Start	Address End	Address Locality
APE_AMISC	0x02990000	0x02991fff	SYSTEM
APE_AMISC_AMISC	0x02990000	0x029907ff	SYSTEM
APE_AMISC_ACTMON	0x02990800	0x02990bff	SYSTEM
APE_AMC	0x02993000	0x02993fff	SYSTEM
APE_ACAST	0x02994000	0x02995fff	SYSTEM
APE_ADAST	0x02996000	0x02997fff	SYSTEM
APE_HSP	0x029a0000	0x02a2ffff	SYSTEM
APE_HSP_COMMON	0x029a0000	0x029affff	SYSTEM
APE_HSP_SM	0x029b0000	0x029effff	SYSTEM
APE_HSP_SM_0_1	0x029b0000	0x029bffff	SYSTEM
APE_HSP_SM_2_3	0x029c0000	0x029cffff	SYSTEM
APE_HSP_SM_4_5	0x029d0000	0x029dffff	SYSTEM
APE_HSP_SM_6_7	0x029e0000	0x029effff	SYSTEM
APE_HSP_SS	0x029f0000	0x02a2ffff	SYSTEM
APE_HSP_SS_0	0x029f0000	0x029fffff	SYSTEM
APE_HSP_SS_1	0x02a00000	0x02a0ffff	SYSTEM
APE_HSP_SS_2	0x02a10000	0x02a1ffff	SYSTEM
APE_HSP_SS_3	0x02a20000	0x02a2ffff	SYSTEM
APE_AGIC	0x02a40000	0x02a7ffff	SYSTEM
APE_AGIC_PAGE0	0x02a40000	0x02a4ffff	SYSTEM
APE_AGIC_PAGE1	0x02a50000	0x02a5ffff	SYSTEM
APE_AGIC_PAGE2	0x02a60000	0x02a6ffff	SYSTEM
APE_AGIC_PAGE3	0x02a70000	0x02a7ffff	SYSTEM
APE_TKE	0x02a80000	0x02adffff	SYSTEM
APE_TKE_SHARED	0x02a80000	0x02a8ffff	SYSTEM
APE_TKE_TMR	0x02a90000	0x02acffff	SYSTEM
APE_TKE_TMR_0	0x02a90000	0x02a9ffff	SYSTEM
APE_TKE_TMR_1	0x02aa0000	0x02aaffff	SYSTEM
APE_TKE_TMR_2	0x02ab0000	0x02abffff	SYSTEM

Block Name	Address Start	Address End	Address Locality
APE_TKE_TMR_3	0x02ac0000	0x02acffff	SYSTEM
APE_TKE_WDT	0x02ad0000	0x02adffff	SYSTEM
APE_TKE_WDT_0	0x02ad0000	0x02adffff	SYSTEM
TSA_NODE_128	0x02b00000	0x02b00fff	SYSTEM
TSA_NODE_129	0x02b01000	0x02b01fff	SYSTEM
TSA_NODE_130	0x02b02000	0x02b02fff	SYSTEM
TSA_NODE_131	0x02b03000	0x02b03fff	SYSTEM
TSA_NODE_132	0x02b04000	0x02b04fff	SYSTEM
TSA_NODE_133	0x02b05000	0x02b05fff	SYSTEM
TSA_NODE_134	0x02b06000	0x02b06fff	SYSTEM
TSA_NODE_135	0x02b07000	0x02b07fff	SYSTEM
TSA_NODE_136	0x02b08000	0x02b08fff	SYSTEM
TSA_NODE_137	0x02b09000	0x02b09fff	SYSTEM
TSA_NODE_138	0x02b0a000	0x02b0afff	SYSTEM
TSA_NODE_139	0x02b0b000	0x02b0bfff	SYSTEM
TSA_NODE_140	0x02b0c000	0x02b0cfff	SYSTEM
TSA_NODE_141	0x02b0d000	0x02b0dfff	SYSTEM
TSA_NODE_142	0x02b0e000	0x02b0efff	SYSTEM
TSA_NODE_143	0x02b0f000	0x02b0ffff	SYSTEM
TSA_NODE_144	0x02b10000	0x02b10fff	SYSTEM
TSA_NODE_145	0x02b11000	0x02b11fff	SYSTEM
TSA_NODE_146	0x02b12000	0x02b12fff	SYSTEM
TSA_NODE_147	0x02b13000	0x02b13fff	SYSTEM
TSA_NODE_148	0x02b14000	0x02b14fff	SYSTEM
TSA_NODE_149	0x02b15000	0x02b15fff	SYSTEM
TSA_NODE_150	0x02b16000	0x02b16fff	SYSTEM
TSA_NODE_151	0x02b17000	0x02b17fff	SYSTEM
TSA_NODE_152	0x02b18000	0x02b18fff	SYSTEM
TSA_NODE_153	0x02b19000	0x02b19fff	SYSTEM

Block Name	Address Start	Address End	Address Locality
TSA_NODE_154	0x02b1a000	0x02b1afff	SYSTEM
TSA_NODE_155	0x02b1b000	0x02b1bfff	SYSTEM
TSA_NODE_156	0x02b1c000	0x02b1cfff	SYSTEM
TSA_NODE_157	0x02b1d000	0x02b1dfff	SYSTEM
TSA_NODE_158	0x02b1e000	0x02b1efff	SYSTEM
TSA_NODE_159	0x02b1f000	0x02b1ffff	SYSTEM
TSA_NODE_160	0x02b20000	0x02b20fff	SYSTEM
TSA_NODE_161	0x02b21000	0x02b21fff	SYSTEM
TSA_NODE_162	0x02b22000	0x02b22fff	SYSTEM
TSA_NODE_163	0x02b23000	0x02b23fff	SYSTEM
TSA_NODE_164	0x02b24000	0x02b24fff	SYSTEM
TSA_NODE_165	0x02b25000	0x02b25fff	SYSTEM
TSA_NODE_166	0x02b26000	0x02b26fff	SYSTEM
TSA_NODE_167	0x02b27000	0x02b27fff	SYSTEM
TSA_NODE_168	0x02b28000	0x02b28fff	SYSTEM
TSA_NODE_169	0x02b29000	0x02b29fff	SYSTEM
TSA_NODE_170	0x02b2a000	0x02b2afff	SYSTEM
TSA_NODE_171	0x02b2b000	0x02b2bfff	SYSTEM
TSA_NODE_172	0x02b2c000	0x02b2cfff	SYSTEM
TSA_NODE_173	0x02b2d000	0x02b2dfff	SYSTEM
TSA_NODE_174	0x02b2e000	0x02b2efff	SYSTEM
TSA_NODE_175	0x02b2f000	0x02b2ffff	SYSTEM
TSA_NODE_176	0x02b30000	0x02b30fff	SYSTEM
TSA_NODE_177	0x02b31000	0x02b31fff	SYSTEM
TSA_NODE_178	0x02b32000	0x02b32fff	SYSTEM
TSA_NODE_179	0x02b33000	0x02b33fff	SYSTEM
TSA_NODE_180	0x02b34000	0x02b34fff	SYSTEM
TSA_NODE_181	0x02b35000	0x02b35fff	SYSTEM
TSA_NODE_182	0x02b36000	0x02b36fff	SYSTEM

Block Name	Address Start	Address End	Address Locality
TSA_NODE_183	0x02b37000	0x02b37fff	SYSTEM
TSA_NODE_184	0x02b38000	0x02b38fff	SYSTEM
TSA_NODE_185	0x02b39000	0x02b39fff	SYSTEM
TSA_NODE_186	0x02b3a000	0x02b3afff	SYSTEM
TSA_NODE_187	0x02b3b000	0x02b3bfff	SYSTEM
TSA_NODE_188	0x02b3c000	0x02b3cfff	SYSTEM
TSA_NODE_189	0x02b3d000	0x02b3dfff	SYSTEM
TSA_NODE_190	0x02b3e000	0x02b3efff	SYSTEM
TSA_NODE_191	0x02b3f000	0x02b3ffff	SYSTEM
TSA_NODE_192	0x02b40000	0x02b40fff	SYSTEM
TSA_NODE_193	0x02b41000	0x02b41fff	SYSTEM
TSA_NODE_194	0x02b42000	0x02b42fff	SYSTEM
TSA_NODE_195	0x02b43000	0x02b43fff	SYSTEM
TSA_NODE_196	0x02b44000	0x02b44fff	SYSTEM
TSA_NODE_197	0x02b45000	0x02b45fff	SYSTEM
TSA_NODE_198	0x02b46000	0x02b46fff	SYSTEM
TSA_NODE_199	0x02b47000	0x02b47fff	SYSTEM
TSA_NODE_200	0x02b48000	0x02b48fff	SYSTEM
TSA_NODE_201	0x02b49000	0x02b49fff	SYSTEM
TSA_NODE_202	0x02b4a000	0x02b4afff	SYSTEM
TSA_NODE_203	0x02b4b000	0x02b4bfff	SYSTEM
TSA_NODE_204	0x02b4c000	0x02b4cfff	SYSTEM
TSA_NODE_205	0x02b4d000	0x02b4dfff	SYSTEM
TSA_NODE_206	0x02b4e000	0x02b4efff	SYSTEM
TSA_NODE_207	0x02b4f000	0x02b4ffff	SYSTEM
TSA_NODE_208	0x02b50000	0x02b50fff	SYSTEM
TSA_NODE_209	0x02b51000	0x02b51fff	SYSTEM
TSA_NODE_210	0x02b52000	0x02b52fff	SYSTEM
TSA_NODE_211	0x02b53000	0x02b53fff	SYSTEM

Block Name	Address Start	Address End	Address Locality
TSA_NODE_212	0x02b54000	0x02b54fff	SYSTEM
TSA_NODE_213	0x02b55000	0x02b55fff	SYSTEM
TSA_NODE_214	0x02b56000	0x02b56fff	SYSTEM
TSA_NODE_215	0x02b57000	0x02b57fff	SYSTEM
TSA_NODE_216	0x02b58000	0x02b58fff	SYSTEM
TSA_NODE_217	0x02b59000	0x02b59fff	SYSTEM
TSA_NODE_218	0x02b5a000	0x02b5afff	SYSTEM
TSA_NODE_219	0x02b5b000	0x02b5bfff	SYSTEM
TSA_NODE_220	0x02b5c000	0x02b5cfff	SYSTEM
TSA_NODE_221	0x02b5d000	0x02b5dfff	SYSTEM
TSA_NODE_222	0x02b5e000	0x02b5efff	SYSTEM
TSA_NODE_223	0x02b5f000	0x02b5ffff	SYSTEM
TSA_NODE_224	0x02b60000	0x02b60fff	SYSTEM
TSA_NODE_225	0x02b61000	0x02b61fff	SYSTEM
TSA_NODE_226	0x02b62000	0x02b62fff	SYSTEM
TSA_NODE_227	0x02b63000	0x02b63fff	SYSTEM
TSA_NODE_228	0x02b64000	0x02b64fff	SYSTEM
TSA_NODE_229	0x02b65000	0x02b65fff	SYSTEM
TSA_NODE_230	0x02b66000	0x02b66fff	SYSTEM
TSA_NODE_231	0x02b67000	0x02b67fff	SYSTEM
TSA_NODE_232	0x02b68000	0x02b68fff	SYSTEM
TSA_NODE_233	0x02b69000	0x02b69fff	SYSTEM
TSA_NODE_234	0x02b6a000	0x02b6afff	SYSTEM
TSA_NODE_235	0x02b6b000	0x02b6bfff	SYSTEM
TSA_NODE_236	0x02b6c000	0x02b6cfff	SYSTEM
TSA_NODE_237	0x02b6d000	0x02b6dfff	SYSTEM
TSA_NODE_238	0x02b6e000	0x02b6efff	SYSTEM
TSA_NODE_239	0x02b6f000	0x02b6ffff	SYSTEM
TSA_NODE_240	0x02b70000	0x02b70fff	SYSTEM

Block Name	Address Start	Address End	Address Locality
TSA_NODE_241	0x02b71000	0x02b71fff	SYSTEM
TSA_NODE_242	0x02b72000	0x02b72fff	SYSTEM
TSA_NODE_243	0x02b73000	0x02b73fff	SYSTEM
TSA_NODE_244	0x02b74000	0x02b74fff	SYSTEM
TSA_NODE_245	0x02b75000	0x02b75fff	SYSTEM
TSA_NODE_246	0x02b76000	0x02b76fff	SYSTEM
TSA_NODE_247	0x02b77000	0x02b77fff	SYSTEM
TSA_NODE_248	0x02b78000	0x02b78fff	SYSTEM
TSA_NODE_249	0x02b79000	0x02b79fff	SYSTEM
TSA_NODE_250	0x02b7a000	0x02b7afff	SYSTEM
TSA_NODE_251	0x02b7b000	0x02b7bfff	SYSTEM
TSA_NODE_252	0x02b7c000	0x02b7cfff	SYSTEM
TSA_NODE_253	0x02b7d000	0x02b7dfff	SYSTEM
TSA_NODE_254	0x02b7e000	0x02b7efff	SYSTEM
TSA_NODE_255	0x02b7f000	0x02b7ffff	SYSTEM
UPHY_0	0x02d00000	0x02d01fff	SYSTEM
UPHY_PLL0	0x02d00000	0x02d01fff	SYSTEM
UPHY_1	0x02d02000	0x02d03fff	SYSTEM
UPHY_LANE0	0x02d02000	0x02d02fff	SYSTEM
UPHY_LANE1	0x02d03000	0x02d03fff	SYSTEM
UPHY_2	0x02d04000	0x02d05fff	SYSTEM
UPHY_PLL1	0x02d04000	0x02d05fff	SYSTEM
UPHY_3	0x02d06000	0x02d07fff	SYSTEM
UPHY_LANE2	0x02d06000	0x02d06fff	SYSTEM
UPHY_LANE3	0x02d07000	0x02d07fff	SYSTEM
UPHY_4	0x02d08000	0x02d09fff	SYSTEM
UPHY_PLL2	0x02d08000	0x02d09fff	SYSTEM
UPHY_5	0x02d0a000	0x02d0bfff	SYSTEM
UPHY_LANE4	0x02d0a000	0x02d0afff	SYSTEM

Block Name	Address Start	Address End	Address Locality
UPHY_LANE5	0x02d0b000	0x02d0bfff	SYSTEM
UPHY_6	0x02d0c000	0x02d0dfff	SYSTEM
UPHY_LANE6	0x02d0c000	0x02d0cfff	SYSTEM
UPHY_LANE7	0x02d0d000	0x02d0dfff	SYSTEM
UPHY_7	0x02d0e000	0x02d0ffff	SYSTEM
UPHY_PLL3	0x02d0e000	0x02d0ffff	SYSTEM
NVHSUPHY_0	0x02f00000	0x02f01fff	SYSTEM
NVHSUPHY_PLL0	0x02f00000	0x02f01fff	SYSTEM
NVHSUPHY_1	0x02f02000	0x02f03fff	SYSTEM
NVHSUPHY_LANE0	0x02f02000	0x02f02fff	SYSTEM
NVHSUPHY_LANE1	0x02f03000	0x02f03fff	SYSTEM
NVHSUPHY_2	0x02f04000	0x02f05fff	SYSTEM
NVHSUPHY_LANE2	0x02f04000	0x02f04fff	SYSTEM
NVHSUPHY_LANE3	0x02f05000	0x02f05fff	SYSTEM
NVHSUPHY_3	0x02f06000	0x02f07fff	SYSTEM
NVHSUPHY_LANE4	0x02f06000	0x02f06fff	SYSTEM
NVHSUPHY_LANE5	0x02f07000	0x02f07fff	SYSTEM
NVHSUPHY_4	0x02f08000	0x02f09fff	SYSTEM
NVHSUPHY_LANE6	0x02f08000	0x02f08fff	SYSTEM
NVHSUPHY_LANE7	0x02f09000	0x02f09fff	SYSTEM
NVHSUPHY_5	0x02f0a000	0x02f0bfff	SYSTEM
NVHSUPHY_PLL1	0x02f0a000	0x02f0bfff	SYSTEM
LIC	0x03000000	0x0300ffff	SYSTEM
LIC_CH0	0x03000000	0x030007ff	SYSTEM
LIC_CH1	0x03000800	0x03000fff	SYSTEM
LIC_CH2	0x03001000	0x030017ff	SYSTEM
LIC_CH3	0x03001800	0x03001fff	SYSTEM
LIC_CH4	0x03002000	0x030027ff	SYSTEM
LIC_CH5	0x03002800	0x03002fff	SYSTEM

Block Name	Address Start	Address End	Address Locality
LIC_CH6	0x03003000	0x030037ff	SYSTEM
LIC_CH7	0x03003800	0x03003fff	SYSTEM
LIC_CH8	0x03004000	0x030047ff	SYSTEM
LIC_CH9	0x03004800	0x03004fff	SYSTEM
LIC_CH10	0x03005000	0x030057ff	SYSTEM
LIC_CH11	0x03005800	0x03005fff	SYSTEM
LIC_COMMON	0x0300f800	0x0300ffff	SYSTEM
UARTA	0x03100000	0x0310ffff	SYSTEM
UARTB	0x03110000	0x0311ffff	SYSTEM
UARTD	0x03130000	0x0313ffff	SYSTEM
UARTE	0x03140000	0x0314ffff	SYSTEM
UARTF	0x03150000	0x0315ffff	SYSTEM
I2C1	0x03160000	0x0316ffff	SYSTEM
UARTH	0x03170000	0x0317ffff	SYSTEM
I2C3	0x03180000	0x0318ffff	SYSTEM
I2C4	0x03190000	0x0319ffff	SYSTEM
I2C5	0x031a0000	0x031affff	SYSTEM
I2C6	0x031b0000	0x031bffff	SYSTEM
I2C7	0x031c0000	0x031cffff	SYSTEM
UARTI	0x031d0000	0x031dffff	SYSTEM
I2C9	0x031e0000	0x031effff	SYSTEM
SPI1	0x03210000	0x0321ffff	SYSTEM
SPI3	0x03230000	0x0323ffff	SYSTEM
SPI4	0x03240000	0x0324ffff	SYSTEM
SPI5	0x03250000	0x0325ffff	SYSTEM
QSPI0	0x03270000	0x0327ffff	SYSTEM
PWM1	0x03280000	0x0328ffff	SYSTEM
PWM2	0x03290000	0x0329ffff	SYSTEM
PWM3	0x032a0000	0x032affff	SYSTEM

Block Name	Address Start	Address End	Address Locality
PWM5	0x032c0000	0x032cffff	SYSTEM
PWM6	0x032d0000	0x032dffff	SYSTEM
PWM7	0x032e0000	0x032effff	SYSTEM
PWM8	0x032f0000	0x032fffff	SYSTEM
QSPI1	0x03300000	0x0330ffff	SYSTEM
SDMMC1	0x03400000	0x0341ffff	SYSTEM
SDMMC1_IMPL	0x03400000	0x0340ffff	SYSTEM
SDMMC1B	0x03410000	0x0341ffff	SYSTEM
SDMMC4	0x03460000	0x0347ffff	SYSTEM
SDMMC4_IMPL	0x03460000	0x0346ffff	SYSTEM
SDMMC4B	0x03470000	0x0347ffff	SYSTEM
HDA	0x03510000	0x0351ffff	SYSTEM
XUSB_PADCTL	0x03520000	0x0353ffff	SYSTEM
XUSB_PADCTL_NONSECURE	0x03520000	0x0352ffff	SYSTEM
XUSB_PADCTL_SECURE	0x03530000	0x0353ffff	SYSTEM
XUSB_AO	0x03540000	0x0354ffff	SYSTEM
XUSB_DEV	0x03550000	0x0355ffff	SYSTEM
XUSB_DEV_BAR0	0x03550000	0x03557fff	SYSTEM
XUSB_DEV_CFG	0x03558000	0x0355ffff	SYSTEM
XUSB_HOST	0x03600000	0x0379ffff	SYSTEM
XUSB_HOST_PF	0x03600000	0x0365ffff	SYSTEM
XUSB_HOST_PF_CFG	0x03600000	0x0360ffff	SYSTEM
XUSB_HOST_PF_BAR0	0x03610000	0x0364ffff	SYSTEM
XUSB_HOST_PF_BAR0_OP	0x03610000	0x0362ffff	SYSTEM
XUSB_HOST_PF_BAR0_RT	0x03630000	0x0363ffff	SYSTEM
XUSB_HOST_PF_BAR0_DB	0x03640000	0x0364ffff	SYSTEM
XUSB_HOST_PF_BAR2	0x03650000	0x0365ffff	SYSTEM
XUSB_HOST_VF0	0x03660000	0x036affff	SYSTEM
XUSB_HOST_VF0_CFG	0x03660000	0x0366ffff	SYSTEM

Block Name	Address Start	Address End	Address Locality
XUSB_HOST_VF0_BAR0	0x03670000	0x036affff	SYSTEM
XUSB_HOST_VF0_BAR0_OP	0x03670000	0x0368ffff	SYSTEM
XUSB_HOST_VF0_BAR0_RT	0x03690000	0x0369ffff	SYSTEM
XUSB_HOST_VF0_BAR0_DB	0x036a0000	0x036affff	SYSTEM
XUSB_HOST_VF1	0x036b0000	0x036fffff	SYSTEM
XUSB_HOST_VF1_CFG	0x036b0000	0x036bffff	SYSTEM
XUSB_HOST_VF1_BAR0	0x036c0000	0x036fffff	SYSTEM
XUSB_HOST_VF1_BAR0_OP	0x036c0000	0x036dffff	SYSTEM
XUSB_HOST_VF1_BAR0_RT	0x036e0000	0x036effff	SYSTEM
XUSB_HOST_VF1_BAR0_DB	0x036f0000	0x036fffff	SYSTEM
XUSB_HOST_VF2	0x03700000	0x0374ffff	SYSTEM
XUSB_HOST_VF2_CFG	0x03700000	0x0370ffff	SYSTEM
XUSB_HOST_VF2_BAR0	0x03710000	0x0374ffff	SYSTEM
XUSB_HOST_VF2_BAR0_OP	0x03710000	0x0372ffff	SYSTEM
XUSB_HOST_VF2_BAR0_RT	0x03730000	0x0373ffff	SYSTEM
XUSB_HOST_VF2_BAR0_DB	0x03740000	0x0374ffff	SYSTEM
XUSB_HOST_VF3	0x03750000	0x0379ffff	SYSTEM
XUSB_HOST_VF3_CFG	0x03750000	0x0375ffff	SYSTEM
XUSB_HOST_VF3_BAR0	0x03760000	0x0379ffff	SYSTEM
XUSB_HOST_VF3_BAR0_OP	0x03760000	0x0377ffff	SYSTEM
XUSB_HOST_VF3_BAR0_RT	0x03780000	0x0378ffff	SYSTEM
XUSB_HOST_VF3_BAR0_DB	0x03790000	0x0379ffff	SYSTEM
PEXCLK0	0x037a0000	0x037a0fff	SYSTEM
PEXCLK2	0x037a1000	0x037a1fff	SYSTEM
CV0_POD	0x037a3000	0x037a3fff	SYSTEM
CV1_POD	0x037a4000	0x037a4fff	SYSTEM

Block Name	Address Start	Address End	Address Locality
CV2_POD	0x037a5000	0x037a5fff	SYSTEM
PEXCLK1	0x037a7000	0x037a7fff	SYSTEM
MIOBFM	0x03800000	0x0380ffff	SYSTEM
APB2JTAG	0x03980000	0x0398ffff	SYSTEM
MIPI_CAL	0x03990000	0x0399ffff	SYSTEM
TACH_1	0x039b0000	0x039bffff	SYSTEM
TACH_0	0x039c0000	0x039cffff	SYSTEM
IST	0x03a60000	0x03a6ffff	SYSTEM
LIC_GTE0	0x03aa0000	0x03aaffff	SYSTEM
LIC_GTE1	0x03ab0000	0x03abffff	SYSTEM
TOP0_HSP	0x03c00000	0x03c9ffff	SYSTEM
TOP0_HSP_COMMON	0x03c00000	0x03c0ffff	SYSTEM
TOP0_HSP_SM	0x03c10000	0x03c4ffff	SYSTEM
TOP0_HSP_SM_0_1	0x03c10000	0x03c1ffff	SYSTEM
TOP0_HSP_SM_2_3	0x03c20000	0x03c2ffff	SYSTEM
TOP0_HSP_SM_4_5	0x03c30000	0x03c3ffff	SYSTEM
TOP0_HSP_SM_6_7	0x03c40000	0x03c4ffff	SYSTEM
TOP0_HSP_SS	0x03c50000	0x03c6ffff	SYSTEM
TOP0_HSP_SS_0	0x03c50000	0x03c5ffff	SYSTEM
TOP0_HSP_SS_1	0x03c60000	0x03c6ffff	SYSTEM
TOP0_HSP_AS	0x03c70000	0x03c8ffff	SYSTEM
TOP0_HSP_AS_0	0x03c70000	0x03c7ffff	SYSTEM
TOP0_HSP_AS_1	0x03c80000	0x03c8ffff	SYSTEM
TOP0_HSP_DB	0x03c90000	0x03c9ffff	SYSTEM
TOP0_HSP_DB_0	0x03c90000	0x03c9ffff	SYSTEM
TOP1_HSP	0x03d00000	0x03d8ffff	SYSTEM
TOP1_HSP_COMMON	0x03d00000	0x03d0ffff	SYSTEM
TOP1_HSP_SM	0x03d10000	0x03d4ffff	SYSTEM
TOP1_HSP_SM_0_1	0x03d10000	0x03d1ffff	SYSTEM

Block Name	Address Start	Address End	Address Locality
TOP1_HSP_SM_2_3	0x03d20000	0x03d2ffff	SYSTEM
TOP1_HSP_SM_4_5	0x03d30000	0x03d3ffff	SYSTEM
TOP1_HSP_SM_6_7	0x03d40000	0x03d4ffff	SYSTEM
TOP1_HSP_SS	0x03d50000	0x03d8ffff	SYSTEM
TOP1_HSP_SS_0	0x03d50000	0x03d5ffff	SYSTEM
TOP1_HSP_SS_1	0x03d60000	0x03d6ffff	SYSTEM
TOP1_HSP_SS_2	0x03d70000	0x03d7ffff	SYSTEM
TOP1_HSP_SS_3	0x03d80000	0x03d8ffff	SYSTEM
PIPE2UPHY	0x03e00000	0x03ffffff	SYSTEM
P2U_HSIO_0	0x03e00000	0x03e0ffff	SYSTEM
P2U_HSIO_1	0x03e10000	0x03e1ffff	SYSTEM
P2U_HSIO_2	0x03e20000	0x03e2ffff	SYSTEM
P2U_HSIO_3	0x03e30000	0x03e3ffff	SYSTEM
P2U_HSIO_4	0x03e40000	0x03e4ffff	SYSTEM
P2U_HSIO_5	0x03e50000	0x03e5ffff	SYSTEM
P2U_HSIO_6	0x03e60000	0x03e6ffff	SYSTEM
P2U_HSIO_7	0x03e70000	0x03e7ffff	SYSTEM
P2U_HSIO_XBAR	0x03e80000	0x03e8ffff	SYSTEM
P2U_NVHS_0	0x03e90000	0x03e9ffff	SYSTEM
P2U_NVHS_1	0x03ea0000	0x03eaffff	SYSTEM
P2U_NVHS_2	0x03eb0000	0x03ebffff	SYSTEM
P2U_NVHS_3	0x03ec0000	0x03ecffff	SYSTEM
P2U_NVHS_4	0x03ed0000	0x03edffff	SYSTEM
P2U_NVHS_5	0x03ee0000	0x03eeffff	SYSTEM
P2U_NVHS_6	0x03ef0000	0x03efffff	SYSTEM
P2U_NVHS_7	0x03f00000	0x03f0ffff	SYSTEM
P2U_NVHS_XBAR	0x03f10000	0x03f1ffff	SYSTEM
EXIO	0x06000000	0x063fffff	SYSTEM
NITRO_IO	0x06400000	0x06400fff	SYSTEM

Block Name	Address Start	Address End	Address Locality
SMMU3	0x07000000	0x07ffffff	SYSTEM
SMMU4	0x08000000	0x08ffffff	SYSTEM
I2C2	0x0c240000	0x0c24ffff	SYSTEM
I2C8	0x0c250000	0x0c25ffff	SYSTEM
SPI2	0x0c260000	0x0c26ffff	SYSTEM
UARTJ	0x0c270000	0x0c27ffff	SYSTEM
UARTC	0x0c280000	0x0c28ffff	SYSTEM
RTC	0x0c2a0000	0x0c2afffff	SYSTEM
PADCTL_A12	0x0c301000	0x0c301fff	SYSTEM
PADCTL_A14	0x0c302000	0x0c302fff	SYSTEM
PADCTL_A15	0x0c303000	0x0c303fff	SYSTEM
CAN1	0x0c310000	0x0c31ffff	SYSTEM
CAN2	0x0c320000	0x0c32ffff	SYSTEM
DMIC5	0x0c330000	0x0c33ffff	SYSTEM
PWM4	0x0c340000	0x0c34ffff	SYSTEM
PMC	0x0c360000	0x0c3afffff	SYSTEM
PMC_IMPL	0x0c360000	0x0c36ffff	SYSTEM
WAKE	0x0c370000	0x0c37ffff	SYSTEM
SCRATCH	0x0c390000	0x0c39ffff	SYSTEM
PMC_MISC	0x0c3a0000	0x0c3afffff	SYSTEM
TSC	0x0c670000	0x0c6bffff	SYSTEM
SYSCTR0	0x0c670000	0x0c67ffff	SYSTEM
SYSCTR1	0x0c680000	0x0c68ffff	SYSTEM
SYSCTR2	0x0c690000	0x0c69ffff	SYSTEM
TSC_IMPL	0x0c6a0000	0x0c6afffff	SYSTEM
TSCUS	0x0c6b0000	0x0c6bffff	SYSTEM
ACTMON	0x0d230000	0x0d23ffff	SYSTEM
SOC_THERM	0x0d280000	0x0d28ffff	SYSTEM
CENTRAL_VTG_CTLR	0x0d290000	0x0d29ffff	SYSTEM

Block Name	Address Start	Address End	Address Locality
CENTRAL_PWR_MGR	0x0d2a0000	0x0d2affff	SYSTEM
CCPLEX_MMCRAB_ARM	0x0e000000	0x0e3fffff	CCPLEX
AXIS_NIC_0	0x0f000000	0x0f0fffff	SYSTEM
RPG_PM	0x0f100000	0x0f149fff	SYSTEM
RPG_PM_VI0	0x0f100000	0x0f100fff	SYSTEM
RPG_PM_VI1	0x0f101000	0x0f101fff	SYSTEM
RPG_PM_ISP0	0x0f102000	0x0f102fff	SYSTEM
RPG_PM_VIC	0x0f103000	0x0f103fff	SYSTEM
RPG_PM_OFA	0x0f104000	0x0f104fff	SYSTEM
RPG_PM_DISPLAY	0x0f10a000	0x0f10afff	SYSTEM
RPG_PM_PMA	0x0f10b000	0x0f10bfff	SYSTEM
RPG_PM_SCF	0x0f110000	0x0f110fff	SYSTEM
RPG_PM_NVDEC0	0x0f111000	0x0f111fff	SYSTEM
RPG_PM_NVENC0	0x0f112000	0x0f112fff	SYSTEM
RPG_PM_PCIE_C0	0x0f114000	0x0f114fff	SYSTEM
RPG_PM_PCIE_C1	0x0f115000	0x0f115fff	SYSTEM
RPG_PM_PCIE_C2	0x0f116000	0x0f116fff	SYSTEM
RPG_PM_PCIE_C3	0x0f117000	0x0f117fff	SYSTEM
RPG_PM_PCIE_C4	0x0f118000	0x0f118fff	SYSTEM
RPG_PM_PCIE_C5	0x0f119000	0x0f119fff	SYSTEM
RPG_PM_PCIE_C6	0x0f11a000	0x0f11afff	SYSTEM
RPG_PM_PCIE_C7	0x0f11b000	0x0f11bfff	SYSTEM
RPG_PM_PCIE_C8	0x0f11c000	0x0f11cfff	SYSTEM
RPG_PM_PCIE_C9	0x0f11d000	0x0f11dfff	SYSTEM
RPG_PM_PCIE_C10	0x0f11e000	0x0f11efff	SYSTEM
RPG_PM_MCF0	0x0f131000	0x0f131fff	SYSTEM
RPG_PM_MCF1	0x0f132000	0x0f132fff	SYSTEM
RPG_PM_MCF2	0x0f133000	0x0f133fff	SYSTEM
PMA	0x0f14a000	0x0f14bfff	SYSTEM

Block Name	Address Start	Address End	Address Locality
PMA_CFG	0x0f14a000	0x0f14afff	SYSTEM
PMA_SEC	0x0f14b000	0x0f14bfff	SYSTEM
RTR	0x0f14d000	0x0f14dfff	SYSTEM
CCPLEX_GIC	0x0f400000	0x0f7fffff	SYSTEM
CCPLEX_GICD	0x0f400000	0x0f40ffff	SYSTEM
CCPLEX_GICA	0x0f410000	0x0f41ffff	SYSTEM
CCPLEX_GICT	0x0f420000	0x0f42ffff	SYSTEM
CCPLEX_GICP	0x0f430000	0x0f43ffff	SYSTEM
CCPLEX_GICR0	0x0f440000	0x0f45ffff	SYSTEM
CCPLEX_GICR1	0x0f460000	0x0f47ffff	SYSTEM
CCPLEX_GICR2	0x0f480000	0x0f49ffff	SYSTEM
CCPLEX_GICR3	0x0f4a0000	0x0f4bffff	SYSTEM
CCPLEX_GICR4	0x0f4c0000	0x0f4dffff	SYSTEM
CCPLEX_GICR5	0x0f4e0000	0x0f4fffff	SYSTEM
CCPLEX_GICR6	0x0f500000	0x0f51ffff	SYSTEM
CCPLEX_GICR7	0x0f520000	0x0f53ffff	SYSTEM
CCPLEX_GICR8	0x0f540000	0x0f55ffff	SYSTEM
CCPLEX_GICR9	0x0f560000	0x0f57ffff	SYSTEM
CCPLEX_GICR10	0x0f580000	0x0f59ffff	SYSTEM
CCPLEX_GICR11	0x0f5a0000	0x0f5bffff	SYSTEM
CCPLEX_GICR12	0x0f5c0000	0x0f5dffff	SYSTEM
CCPLEX_GICR13	0x0f5e0000	0x0f5fffff	SYSTEM
CCPLEX_GICR14	0x0f600000	0x0f61ffff	SYSTEM
CCPLEX_GICR15	0x0f620000	0x0f63ffff	SYSTEM
CCPLEX_GICDA	0x0f640000	0x0f64ffff	SYSTEM
CCPLEX_GICFMU	0x0f7f0000	0x0f7fffff	SYSTEM
SMMU2	0x10000000	0x10ffffff	SYSTEM
SMMU1	0x11000000	0x11ffffff	SYSTEM
SMMU0	0x12000000	0x12ffffff	SYSTEM

Block Name	Address Start	Address End	Address Locality
DISP	0x13800000	0x138effff	SYSTEM
DISP_CORE_1	0x13800000	0x1383ffff	SYSTEM
DISP_EC	0x13840000	0x13840fff	SYSTEM
DISP_CORE_2	0x13841000	0x138effff	SYSTEM
HOST1X	0x13e00000	0x13fcffff	SYSTEM
HOST1X_COMMON	0x13e00000	0x13e0ffff	SYSTEM
HOST1X_0	0x13e40000	0x13e4ffff	SYSTEM
HOST1X_1	0x13e50000	0x13e5ffff	SYSTEM
HOST1X_2	0x13e60000	0x13e6ffff	SYSTEM
HOST1X_3	0x13e70000	0x13e7ffff	SYSTEM
HOST1X_4	0x13e80000	0x13e8ffff	SYSTEM
HOST1X_5	0x13e90000	0x13e9ffff	SYSTEM
HOST1X_6	0x13ea0000	0x13eaffff	SYSTEM
HOST1X_7	0x13eb0000	0x13ebffff	SYSTEM
HOST1X_ACTMON0	0x13ef0000	0x13efffff	SYSTEM
HOST1X_ACTMON1	0x13f00000	0x13f0ffff	SYSTEM
HOST1X_ACTMON2	0x13f10000	0x13f1ffff	SYSTEM
HOST1X_ACTMON3	0x13f20000	0x13f2ffff	SYSTEM
HOST1X_ACTMON4	0x13f30000	0x13f3ffff	SYSTEM
HOST1X_ACTMON5	0x13f40000	0x13f4ffff	SYSTEM
PCIE_CTL	0x14080000	0x141fffff	SYSTEM
PCIE_C8_CTL	0x140a0000	0x140bffff	SYSTEM
PCIE_C8_CTL_NONSECURE	0x140a0000	0x140affff	SYSTEM
PCIE_C9_CTL	0x140c0000	0x140dffff	SYSTEM
PCIE_C9_CTL_NONSECURE	0x140c0000	0x140cffff	SYSTEM
PCIE_C10_CTL	0x140e0000	0x140fffff	SYSTEM
PCIE_C10_CTL_NONSECURE	0x140e0000	0x140effff	SYSTEM
PCIE_C1_CTL	0x14100000	0x1411ffff	SYSTEM
PCIE_C1_CTL_NONSECURE	0x14100000	0x1410ffff	SYSTEM

Block Name	Address Start	Address End	Address Locality
PCIE_C2_CTL	0x14120000	0x1413ffff	SYSTEM
PCIE_C2_CTL_NONSECURE	0x14120000	0x1412ffff	SYSTEM
PCIE_C3_CTL	0x14140000	0x1415ffff	SYSTEM
PCIE_C3_CTL_NONSECURE	0x14140000	0x1414ffff	SYSTEM
PCIE_C4_CTL	0x14160000	0x1417ffff	SYSTEM
PCIE_C4_CTL_NONSECURE	0x14160000	0x1416ffff	SYSTEM
PCIE_C0_CTL	0x14180000	0x1419ffff	SYSTEM
PCIE_C0_CTL_NONSECURE	0x14180000	0x1418ffff	SYSTEM
PCIE_C5_CTL	0x141a0000	0x141bffff	SYSTEM
PCIE_C5_CTL_NONSECURE	0x141a0000	0x141affff	SYSTEM
PCIE_C6_CTL	0x141c0000	0x141dffff	SYSTEM
PCIE_C6_CTL_NONSECURE	0x141c0000	0x141cffff	SYSTEM
PCIE_C7_CTL	0x141e0000	0x141fffff	SYSTEM
PCIE_C7_CTL_NONSECURE	0x141e0000	0x141effff	SYSTEM
VI2	0x14c00000	0x14efffff	SYSTEM
VI2_THI	0x14f00000	0x14ffffff	SYSTEM
VI2_THI_CPU	0x14f00000	0x14f03fff	SYSTEM
VI2_THI_EC	0x14f04000	0x14f04fff	SYSTEM
VI2_THI_ENGINE	0x14f05000	0x14ffffff	SYSTEM
VIC	0x15340000	0x1537ffff	SYSTEM
VIC_SEC	0x1534e000	0x1534efff	SYSTEM
VIC_EC	0x1534f000	0x1534ffff	SYSTEM
NVJPG	0x15380000	0x153bffff	SYSTEM
NVDEC	0x15480000	0x154bffff	SYSTEM
NVENC	0x154c0000	0x154fffff	SYSTEM
NVENC_SEC	0x154ce000	0x154cefff	SYSTEM
NVENC_EC	0x154cf000	0x154cffff	SYSTEM
NVJPG1	0x15540000	0x1557ffff	SYSTEM
DPAUX	0x155c0000	0x155cffff	SYSTEM

Block Name	Address Start	Address End	Address Locality
SE1	0x15810000	0x1581ffff	SYSTEM
SE2	0x15820000	0x1582ffff	SYSTEM
SE3	0x15830000	0x1583ffff	SYSTEM
SE4	0x15840000	0x1584ffff	SYSTEM
NVCSI	0x15a00000	0x15a4ffff	SYSTEM
NVCSI_ENGINE	0x15a00000	0x15a4afff	SYSTEM
NVCSI_EC	0x15a4b000	0x15a4ffff	SYSTEM
OFA	0x15a50000	0x15a5ffff	SYSTEM
OFA_SEC	0x15a5e000	0x15a5efff	SYSTEM
OFA_EC	0x15a5f000	0x15a5ffff	SYSTEM
VI	0x15c00000	0x15effffff	SYSTEM
VI_THI	0x15f00000	0x15ffffff	SYSTEM
VI_THI_CPU	0x15f00000	0x15f03fff	SYSTEM
VI_THI_EC	0x15f04000	0x15f04fff	SYSTEM
VI_THI_ENGINE	0x15f05000	0x15ffffff	SYSTEM
GPU	0x17000000	0x1fffffff	SYSTEM
CSITE	0x24000000	0x27ffffff	SYSTEM
CSITE_MISC0	0x24000000	0x2401ffff	SYSTEM
CSITE_CFG	0x24020000	0x2403ffff	SYSTEM
CSITE_MISC1	0x24040000	0x2440ffff	SYSTEM
LA	0x24410000	0x2441ffff	SYSTEM
CSITE_MISC2	0x24420000	0x2443ffff	SYSTEM
CSITE_MISC3	0x24480000	0x2473ffff	SYSTEM
CSITE_MISC4	0x24780000	0x247fffff	SYSTEM
STM	0x25000000	0x25ffffff	SYSTEM
CSITE_CCPLEX	0x26000000	0x27ffffff	SYSTEM
PCIE_32BIT	0x28000000	0x3fffffff	SYSTEM
PCIE_C7_32BIT	0x28000000	0x29ffffff	SYSTEM

Block Name	Address Start	Address End	Address Locality
PCIE_C7_32BIT_EP	0x28000000	0x28001fff	SYSTEM_CFG.PCIE_C7_CTL.PCIE_RP_APPL_DM_TYPE_0.DEVICE_TYPE.END_POINT
PCIE_C7_32BIT_RP	0x28000000	0x28001fff	SYSTEM_CFG.PCIE_C7_CTL.PCIE_RP_APPL_DM_TYPE_0.DEVICE_TYPE.ROOT_PORT
PCIE_C7_32BIT_DMA	0x28040000	0x2807ffff	SYSTEM
PCIE_C8_32BIT	0x2a000000	0x2bffffff	SYSTEM
PCIE_C8_32BIT_EP	0x2a000000	0x2a001fff	SYSTEM_CFG.PCIE_C8_CTL.PCIE_RP_APPL_DM_TYPE_0.DEVICE_TYPE.END_POINT
PCIE_C8_32BIT_RP	0x2a000000	0x2a001fff	SYSTEM_CFG.PCIE_C8_CTL.PCIE_RP_APPL_DM_TYPE_0.DEVICE_TYPE.ROOT_PORT
PCIE_C8_32BIT_DMA	0x2a040000	0x2a07ffff	SYSTEM
PCIE_C9_32BIT	0x2c000000	0x2dffffff	SYSTEM
PCIE_C9_32BIT_EP	0x2c000000	0x2c001fff	SYSTEM_CFG.PCIE_C9_CTL.PCIE_RP_APPL_DM_TYPE_0.DEVICE_TYPE.END_POINT
PCIE_C9_32BIT_RP	0x2c000000	0x2c001fff	SYSTEM_CFG.PCIE_C9_CTL.PCIE_RP_APPL_DM_TYPE_0.DEVICE_TYPE.ROOT_PORT
PCIE_C9_32BIT_DMA	0x2c040000	0x2c07ffff	SYSTEM
PCIE_C10_32BIT	0x2e000000	0x2fffffff	SYSTEM
PCIE_C10_32BIT_EP	0x2e000000	0x2e001fff	SYSTEM_CFG.PCIE_C10_CTL.PCIE_RP_APPL_DM_TYPE_0.DEVICE_TYPE.END_POINT
PCIE_C10_32BIT_RP	0x2e000000	0x2e001fff	SYSTEM_CFG.PCIE_C10_CTL.PCIE_RP_APPL_DM_TYPE_0.DEVICE_TYPE.ROOT_PORT
PCIE_C10_32BIT_DMA	0x2e040000	0x2e07ffff	SYSTEM
PCIE_C1_32BIT	0x30000000	0x31ffffff	SYSTEM
PCIE_C1_32BIT_RP	0x30000000	0x30001fff	SYSTEM
PCIE_C1_32BIT_DMA	0x30040000	0x3007ffff	SYSTEM
PCIE_C2_32BIT	0x32000000	0x33ffffff	SYSTEM
PCIE_C2_32BIT_RP	0x32000000	0x32001fff	SYSTEM
PCIE_C2_32BIT_DMA	0x32040000	0x3207ffff	SYSTEM
PCIE_C3_32BIT	0x34000000	0x35ffffff	SYSTEM

Block Name	Address Start	Address End	Address Locality
PCIE_C3_32BIT_RP	0x34000000	0x34001fff	SYSTEM
PCIE_C3_32BIT_DMA	0x34040000	0x3407ffff	SYSTEM
PCIE_C4_32BIT	0x36000000	0x37ffffff	SYSTEM
PCIE_C4_32BIT_EP	0x36000000	0x36001fff	SYSTEM_CFG.PCIE_C4_CTL.PCIE_RP_APPL_DM_TYPE_0.DEVICE_TYPE.END_POINT
PCIE_C4_32BIT_RP	0x36000000	0x36001fff	SYSTEM_CFG.PCIE_C4_CTL.PCIE_RP_APPL_DM_TYPE_0.DEVICE_TYPE.ROOT_PORT
PCIE_C4_32BIT_DMA	0x36040000	0x3607ffff	SYSTEM
PCIE_C0_32BIT	0x38000000	0x39ffffff	SYSTEM
PCIE_C0_32BIT_EP	0x38000000	0x38001fff	SYSTEM_CFG.PCIE_C0_CTL.PCIE_RP_APPL_DM_TYPE_0.DEVICE_TYPE.END_POINT
PCIE_C0_32BIT_RP	0x38000000	0x38001fff	SYSTEM_CFG.PCIE_C0_CTL.PCIE_RP_APPL_DM_TYPE_0.DEVICE_TYPE.ROOT_PORT
PCIE_C0_32BIT_DMA	0x38040000	0x3807ffff	SYSTEM
PCIE_C5_32BIT	0x3a000000	0x3bffffff	SYSTEM
PCIE_C5_32BIT_EP	0x3a000000	0x3a001fff	SYSTEM_CFG.PCIE_C5_CTL.PCIE_RP_APPL_DM_TYPE_0.DEVICE_TYPE.END_POINT
PCIE_C5_32BIT_RP	0x3a000000	0x3a001fff	SYSTEM_CFG.PCIE_C5_CTL.PCIE_RP_APPL_DM_TYPE_0.DEVICE_TYPE.ROOT_PORT
PCIE_C5_32BIT_DMA	0x3a040000	0x3a07ffff	SYSTEM
PCIE_C6_32BIT	0x3c000000	0x3dffffff	SYSTEM
PCIE_C6_32BIT_EP	0x3c000000	0x3c001fff	SYSTEM_CFG.PCIE_C6_CTL.PCIE_RP_APPL_DM_TYPE_0.DEVICE_TYPE.END_POINT
PCIE_C6_32BIT_RP	0x3c000000	0x3c001fff	SYSTEM_CFG.PCIE_C6_CTL.PCIE_RP_APPL_DM_TYPE_0.DEVICE_TYPE.ROOT_PORT
PCIE_C6_32BIT_DMA	0x3c040000	0x3c07ffff	SYSTEM
APE_ARAM	0x3f800000	0x3fffffff	APE
ON_CHIP_DATA	0x40000000	0x7fffffff	SYSTEM
SYSRAM_0	0x40000000	0x4fffffff	SYSTEM
SYSRAM_0_IMPL	0x40000000	0x4007ffff	SYSTEM

Block Name	Address Start	Address End	Address Locality
GIC_MSI	0x54000000	0x57ffffff	SYSTEM
SYNCPOINT_0	0x60000000	0x63ffffff	SYSTEM
DRAM	0x80000000	0x207fffffff	SYSTEM
EMEM	0x80000000	0x207fffffff	SYSTEM
EMEM_32BIT	0x80000000	0xffffffff	SYSTEM
PCIE_64BIT	0x2080000000	0x3fffffffff	SYSTEM
PCIE_C1_64BIT	0x2080000000	0x20bfffffff	SYSTEM
PCIE_C2_64BIT	0x20c0000000	0x20ffffffff	SYSTEM
PCIE_C3_64BIT	0x2100000000	0x213fffffff	SYSTEM
PCIE_C6_64BIT	0x2140000000	0x217fffffff	SYSTEM
PCIE_C7_64BIT	0x2180000000	0x21bfffffff	SYSTEM
PCIE_C8_64BIT	0x21c0000000	0x21ffffffff	SYSTEM
PCIE_C9_64BIT	0x2200000000	0x223fffffff	SYSTEM
PCIE_C10_64BIT	0x2240000000	0x227fffffff	SYSTEM
PCIE_C4_64BIT	0x2280000000	0x267fffffff	SYSTEM
PCIE_C0_64BIT	0x2680000000	0x2a7fffffff	SYSTEM
PCIE_C5_64BIT	0x2a80000000	0x2e7fffffff	SYSTEM

3.3 Address Space Translation (AST)

3.3.1 Overview

The Address Space Translation (AST) converts local AXI physical addresses of the embedded Cortex-R5 and Cortex-A9 processor cores to either virtual or physical Memory Controller (MC) addresses. The AST also adds MC-specific attributes to each address range.

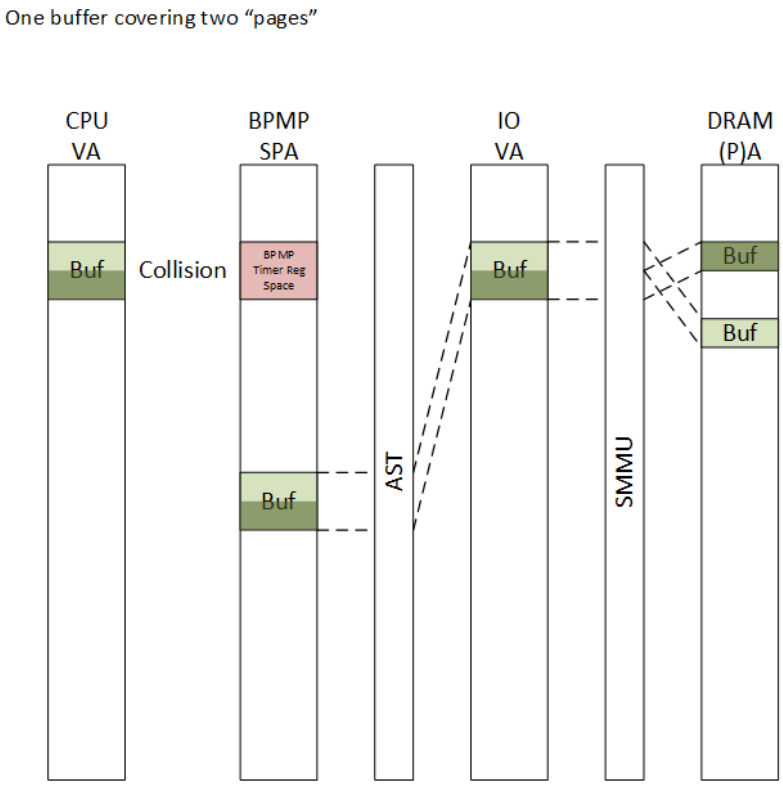
A processor may need to use both virtual and physical addresses if it has a dedicated portion of system DRAM that is not visible to the Operating System (OS), and is protected from access by other blocks in the SoC. Accesses to the dedicated region of DRAM must be sent to the MC as physical addresses, and must bypass the System Memory Management Unit (SMMU). If that same

processor uses virtual addresses provided by a driver to DMA data, them the processor must support sending both physical and virtual addresses to the MC.

The figure below shows a simple address map illustrating address space conflicts that the AST is designed to help resolve. In this example, the OS uses a shared page table for the CPU MMU and SMMU. The Virtual address used for the shared buffer would collide with the BPMP Timer address. The BPMP uses the AST to relocate the buffer in its local address space, but still generates the correct virtual address to the SMMU.

This document describes an implementation where the mapping of protected physical regions is handled by boot code during initialization and is completely transparent to the OS.

Figure 3.6 Address Space Conflict Example



3.3.1.1 Features

The AST block provides the following function:

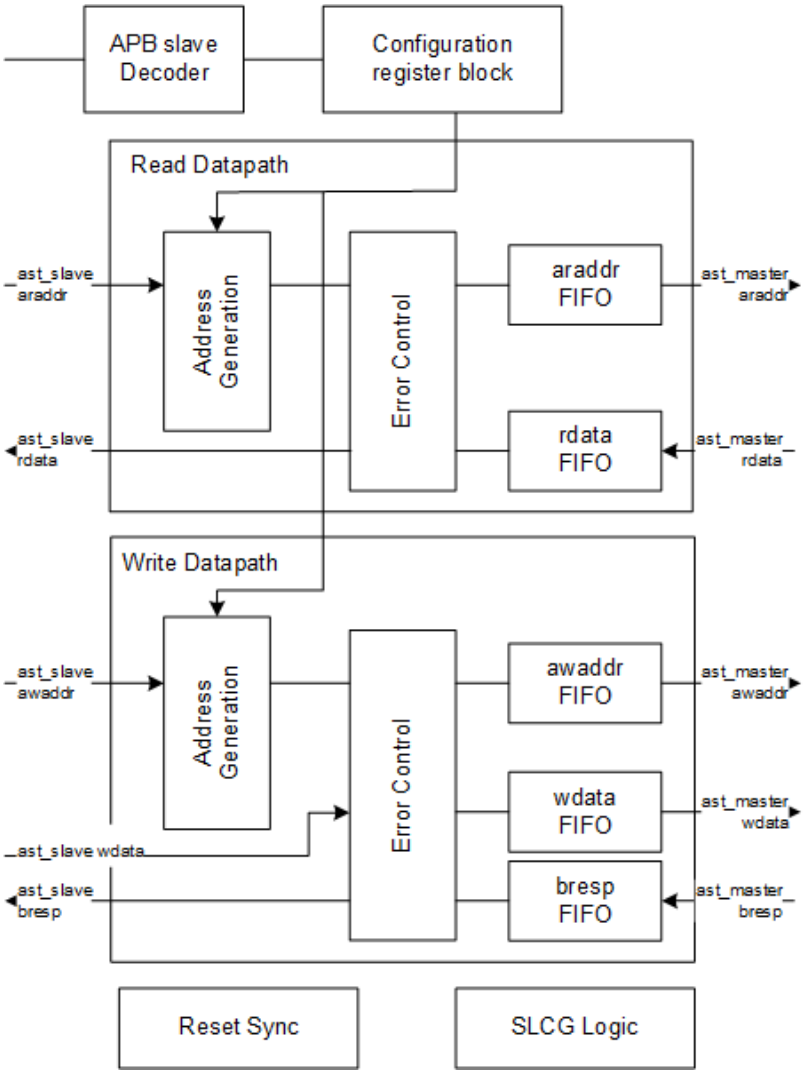
- A set of regions that provide a mapping from the local AXI address space to the MC address space. Each region consists of:
  - The base address and size of the local region
  - The base address of the MC region

    - Snoop attribute (if routed to SCF)
    - Address space ID (StreamID)
- AXI Master and Slave interfaces
- APB Configuration Register block
    - Per region access controls
    - Per region lock controls
    - Global Translation attribute lock
- Error Detection
    - Configurable behavior for requests that do not match a region or do not have the right protection attributes (TrustZone® or security group):
        - either pass-through unchanged with a default set of attributes,
        - or return a DECERR response

3.3.2 Functional Description

During normal operation the AST block functions as an AXI pipeline with internal address translation. Each pipeline stage captures information from the previous stage using the AXI read/valid protocol.

Figure 3.7 AST Block Diagram



3.3.2.1 Address Generation Blocks

The address generation blocks are responsible for address translation and attribute insertion. AST supports eight regions (address generation blocks) in each path (write and read data paths) using which it can translate minimum of eight address ranges. Each region consists of SLAVE_BASE, MASTER_BASE, MASK, and CONTROL registers to facilitate address translation.

3.3.2.2 Read Error Response

When a match error is detected, the read error control block generates a DECERR response for each required data beat. Read error responses can only be pushed into the FIFO if the master read

response interface is idle. The read error control block also generates DECERR responses if the block signal is asserted.

3.3.2.3 Write Error Response

When a match error is detected, the write error control block generates a DECERR response. The write error control block also matches the AWID of the discarded transaction to discard the corresponding data on the WDATA channel. The write error control block also generates DECERR responses when the block signal is asserted.

Since Cortex-R5 processors can send write data before the write address, the write data channel is stalled by the write error control block until the write address check occurs if data arrives before the address.

3.3.2.4 Decode Error Generation

The AST is responsible for generating a decode error response in the following cases:

- A request matches multiple regions.
- A request does not match a region and AST_CONTROL[MatchErrCtl] is set.
- The VMIndx for the request points to an AST_STREAMID_CTL register where the enable bit is not set.
- The ast_block input is asserted.

If a decode error is generated, the AST logs the address of the request in the error log registers.

3.3.2.5 Clocks and Resets

3.3.2.5.1 Clock Requirements

AST has two clock ports: ast_pclk and ast_core_clk.

1. ast_pclk: Clocks the APB interface and the APB slave plugin
2. ast_core_clk: Clocks the input/output AXI interfaces, core logic, and configuration registers

3.3.2.6 AST Software Requirements

3.3.2.6.1 Requirements for Changing an AST Region Mapping

1. If the region is cacheable in a local cache, any cache lines with an address in the region must be flushed before changing the address map.
2. Before changing the mapping of an AST region all pending transactions to that region must be completed.

3.3.2.7 Programming Examples

To remap a region of local memory space using the AST the Region Save Base, Region Mask, and Region master base register must be programmed correctly.

3.3.2.7.1 32-bit Input Address and 32-bit Output Addresses

The following shows an example of how to program the AST region 0 to map the 64 KB local address region between 0x8000_0000 and 0x8000_FFFF to the system address region between 0x4FFF_0000 and 0x4FFF_FFFF.

- AST_REGION_0_MASTER_BASE_LO = 0x4FFF_0000
- AST_REGION_0_MASTER_BASE_HI = 0x0000_0000
- AST_REGION_0_MASK_LO = 0x0000_F000
  - Setting bits 15-12 of the mask
    - Defines the region as 64K
    - Defines a region match as address bits 31-16 equal to 0x8000
    - Define the output address as 0x4FFF_XXXX where XXXX is the value of the input address
- AST_REGION_0_MASK_HI = 0x0000_0000
- AST_REGION_0_SLAVE_BASE_HI = 0x0000_0000
- AST_REGION_0_SLAVE_BASE_LO = 0x8000_0001 (the LSB of AST_REGION_0_SLAVE_BASE_LO is the region enable bit, hence this bit needs to be configured at last).

In this example, an input address of 0x8000_1000 would generate an output address of 0x4FFF_1000.

3.3.2.7.2 32-bit Input Address and 40-bit Output Address

The following shows an example of how to program the AST region 0 to map the 1 MB local address region between 0x4000_0000 and 0x400F_FFFF to the system address region between 0x03_C000_0000 and 0x03_C00F_FFFF.

- AST_REGION_0_MASTER_BASE_LO = 0xC000_0000

- AST_REGION_0_MASTER_BASE_HI = 0x0000_0003
- AST_REGION_0_MASK_LO = 0x000F_F000
  - Setting bits 19-12 of the mask
    - Defines the region as 1MB
    - Defines a region match as address bits 31-20 equal to 0x400
    - Define the output address as 0x03_C00X_XXXX where X_XXXX is the value of the input address
- AST_REGION_0_MASK_HI = 0x0000_0000
- AST_REGION_0_SLAVE_BASE_HI = 0x0000_0000
- AST_REGION_0_SLAVE_BASE_LO = 0x4000_0001 (the LSB of AST_REGION_0_SLAVE_BASE_LO is the region enable bit, hence this bit needs to be configured at last).

In this example, an input address of 0x4006_1000 would generate an output address of 0x03_C006_1000.

3.3.3 Programming Guidelines

The following programming guidelines must be followed to ensure proper operation of the AST.

- The region mask must be programed to specify a power of two aligned regions.
- The region slave and master addresses must be aligned to the region size.
- Software must ensure that AST_REGION_SLAVE_BASE_LO[Enable] is 0 before programming any of the region registers.

3.3.3.1 Steps to Configure Region #i in AST

1. Program the following register fields to select the address range that need to be translated.
   a. AST_REGION_<i>_SLAVE_BASE_LO. SlvBase
   b. AST_REGION_<i>_SLAVE_BASE_HI. SlvBase
   c. AST_REGION_<i>_MASK_LO. Mask
   d. AST_REGION_<i>_MASK _HI. Mask
2. Program the following register fields for the desired output (translated) address.
   a. AST_REGION_<i>_MASTER_BASE _LO. MastBase
   b. AST_REGION_<i>_MASTER_BASE _HI. MastBase
3. If an AXI transaction must bypass SMMU (physical address), program.
   a. APS_AST_REGION_<i>_CONTROL_0. Physical = 1
4. If the AXI transaction must go through SMMU, select the appropriate stream id by programming the following fields.
   a. APS_AST_REGION_<i>_CONTROL_0. Physical = 0.

b. APS_AST_REGION_<i>_CONTROL_0. VMIndex field.
c. APS_AST_STREAMID_CTL_0[VMIndex]. StreamID field.
d. APS_AST_STREAMID_CTL_0[VMIndex]. Enable = 1.
Note: APS_AST_STREAMID_CTL_0[0:15] registers are typically programmed by the Hypervisor.

5. Program the following fields in APS_AST_REGION_<i>_CONTROL_0 register to select the valid memory attributes.
   a. APS_AST_REGION_<i>_CONTROL_0. CarveOutID
   b. APS_AST_REGION_<i>_CONTROL_0. Snoop
6. Enable the region by setting AST_REGION_<i>_SLAVE_BASE.Enable register bit to 1.

3.3.4 AST Registers

Refer to "Reading Register Tables" in the Introduction chapter for the register table protocol as well as recommendations for accessing registers.

There are 17 instances of the AST registers in the following modules as shown in the table below. The register descriptions in this section provide the offset of each register with base addresses listed in the table.

Module	Instance Name	Base Address
Always ON Cluster (AON) and SPE	AON_AST_0	0x0c040000
	AON_AST_1	0x0c050000
Audio Processing Engine (APE)	APE_ACAST	0x02994000
	APE_ADAST	0x02996000
Boot and Power Management Processor (BPMP)	BPMP_AST_0	0x0d040000
	BPMP_AST_1	0x0d050000
Real-time Camera Engine (RCE)	RCE_AST_0	0x0b840000
	RCE_AST_1	0x0b850000
Safety Cluster Engine (SCE)	SCE_AST_0	0x0b040000
	SCE_AST_1	0x0b050000
Display Cluster Engine (DCE)	DCE_AST_0	0x0d840000
	DCE_AST_1	0x0d850000
Functional Safety Island (FSI)	FSI_CHSM_AST	0x092d0000
	FSI_CPU0_AST	0x09280000

Module	Instance Name	Base Address
	FSI_CPU1_AST	0x09290000
	FSI_CPU2_AST	0x092a0000
	FSI_CPU3_AST	0x092b0000

R/W Attribute	Definition
RO	Read-only
RW	Read-write
RW1	Read-write one only: Once set this bit can only be cleared by a system reset
RWCL	Read-write-Carveout-Lock: ▪ Read-only (and can only be reset by a system reset) if (AST_CONTROL[CarveOutLock] == 1) ▪ Read-write if (AST_CONTROL[CarveOutLock] == 0)
RWGL	Read-write-Global-Lock: ▪ Read-only (and can only be reset by a system reset) if (AST_CONTROL[Lock] == 1) ▪ Read-write if AST_CONTROL[Lock] == 0
RWGL_Region	Read-only (and can only be reset by a system reset) if (AST_CONTROL[Lock] == 1 || AST_REGION_*_CONTROL[Lock] == 1) * Read-write if (AST_CONTROL[Lock] == 0 && AST_REGION_*_CONTROL[Lock] == 0)
RWRL	Read-write-Region-Lock: ▪ Read-only (and can only be reset by a system reset) if (AST_REGION_CONTROL[RegionNum][Lock] == 1) ▪ Read-write if (AST_REGION_CONTROL[RegionNum][Lock] == 0)

APS_AST_CONTROL_0

Offset: 0x0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_GBL_SEC_CONTROL_0
Reset: 0x1fc80000 (0b0001,1111,11x0,1000,0xxx,xx00,000x,x000)

Bit	R/W Attribute	Reset	Description
31	RW	0x0	**ApbOvrOn**: APB Clock Override: Set to 1 to force APB clock always on in AST. **1** = APB SLCG is disabled

Bit	R/W Attribute	Reset	Description
30	RW	0x0	**NicOvrOn**: NIC Clock Override: Set to 1 to force NIC clock always on in AST. **1** = NIC SLCG is disabled
20	RW1	0x0	**CarveOutLock**: Carveout Lock. This bit prevents writes to all Carve Out controls when set to 1. **0** = FALSE **1** = TRUE
19	RWGL	0x1	**DefPhysical**: Default Physical Select. Specifies the default how the StreamID is selected for default accesses. **0** = DefVMIndx is used to select the StreamID. **1** = PhysStreamID is used
18:15	RWGL	0x0	**DefVMIndex**: Default VM Index. Specifies the default VM Index used to select the Stream ID when (DefPhysical == 0).
9:5	RWCL	0x0	**DefCarveOutID**: Default MC Carveout ID. Specifies the carveout ID for default accesses. This field specifies the state output on ast_master_a[w,r]user[15,11] for requests that do not match a region.
2	RWGL	0x0	**DefSnoop**: Snoop. Specifies if default accesses snoop the Main CPU caches. This bit controls the state output on ast_master_a[w,r]user[8] for requests that do not match a region. 0 = Do not snoop request. 1 = Snoop request. **0** = DISABLE **1** = ENABLE
1	RW	0x0	**MatchErrCtl**: Match Error Control: Specifies how transactions that do not match a region are handled. 0 = Transactions that do not match a region are forwarded untranslated with the default attributes. 1 = Transactions that do not match a region return a decode error on the AXI slave interface. **0** = NO_DECERR **1** = DECERR
0	RW1	0x0	**Lock**: Security lock. This bit prevents writing to all RWGL and RWGL_Region bits. **0** = FALSE **1** = TRUE

APS_AST_ERROR_STATUS_0

OverFlow: This bit is set to 1 by Hardware when (Valid == 1) and a decode error response is generated by the AST

VMIndxErr: This bit is set to 1 by Hardware when (Valid == 0) and a decode error response is generated by the AST because a disabled VMIndx was used

Valid: This bit is set to 1 by Hardware when (Valid == 0) and a decode error response is generated by the AST Software can write this bit to 0 to clear the logged errors (Clears valid, overflow, error-address bits)

Offset: 0x4
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_GBL_SEC_CONTROL_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	R/W Attribute	Reset	Description
2	RO	0x0	**Overflow**: Error Overflow. This bit is set to 1 by Hardware when (AST_ERROR_STATUS[Valid] == 1) (as result of a previous error) and a new DEC_ERR occurs.
1	RO	0x0	**VMIndxErr**: VM Index Error. This bit is set to 1 by Hardware when (AST_ERROR_STATUS[Valid] == 0) and when a DEC_ERR response is returned because a disabled VMIndx was programmed.
0	RW	0x0	**Valid**: Error valid. This bit is set to 1 by Hardware when a DEC_ERR response is returned. **1** = Error Valid. Software can write this bit to 0 to clear the logged errors (Clears valid, overflow, error-address bits).

APS_AST_ERROR_ADDR_LO_0

ErrAddrLo: Logs lower 32 bits of the request that caused a decode error. This field is not updated if (AST_ERROR_STATUS[Valid] == 1).

Offset: 0x8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_GBL_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ErrAddrLo**: Error Address Low. When a DEC_ERR response is returned and (AST_ERROR_STATUS[Valid] == 0), then the lower 32 bits of the error address are latched into this register. This field is not updated if (AST_ERROR_STATUS[Valid] == 1).

APS_AST_ERROR_ADDR_HI_0

ErrAddrhi: Logs upper 32 bits of the request that caused a decode error. This field is not updated if (AST_ERROR_STATUS[Valid] == 1).

Offset: 0xc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_GBL_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ErrAddrHi**: Error Address high. When a DEC_ERR response is returned and (AST_ERROR_STATUS[Valid] == 0), the upper 32 bits of the error address are latched into this register. This field is not updated if (AST_ERROR_STATUS[Valid] == 1).

APS_AST_STREAMID_CTL_0

This is an array of 16 identical register entries; the register fields below apply to each entry.
Full register list is: APS_AST_STREAMID_CTL_[i], among which [i] belongs to [0..15].
Offset: 0x20,..,0x5c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_HYP_SEC_CONTROL_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,xxxx,xxx0)

Bit	Reset	Description
15:8	0x0	**StreamID**: This specifies the StreamID output when the VMIndx field is programmed to N in a region control register. N is the STREAMID_CTL register number.

Bit	Reset	Description
0	0x0	**Enable**: VM Index Enable. When this bit is set StreamID[N] can be selected by the VMIndx N. 0 = VM Index disabled. 1 = VM Index enabled. **0** = DISABLE **1** = ENABLE

APS_AST_REGION_0_SLAVE_BASE_LO_0

Offset: 0x100
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_0_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	R/W Attribute	Reset	Description
31:12	RWRL	0x0	**SlvBase**: Region Slave Base Address. This field specifies bits 31:12 of the Base address for the region on the AXI slave interface. This address is compared with the incoming slave address to determine if a region match occurs.
0	RWRL	0x0	**Enable**: Region Enable. This enables the translation region. 0 = Translation region disabled. 1 = Translation region enabled, **0** = FALSE **1** = TRUE

APS_AST_REGION_0_SLAVE_BASE_HI_0

Offset: 0x104
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_0_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W Attribute	Reset	Description
31:0	RWRL	0x0	**SlvBase**: Region Slave Base Address. This field specifies bits 63:32 of the Base address for the region on the AXI slave interface. This address is compared with the incoming slave address to determine if a region match occurs.

APS_AST_REGION_0_MASK_LO_0

Offset: 0x108
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_0_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxxx)

Bit	R/W Attribute	Reset	Description
31:12	RWRL	0x0	**Mask**: Region Mask Address[31:12]. This field is used to mask incoming address bits when performing the region compare. This field is also used to mask untranslated address bits when generating the output address.

APS_AST_REGION_0_MASK_HI_0

Offset: 0x10c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_0_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W Attribute	Reset	Description
31:0	RWRL	0x0	**Mask**: Region Mask Address[63:32]. This field is used to mask incoming address bits when performing the region compare. This field is also used to mask untranslated address bits when generating the output address.

APS_AST_REGION_0_MASTER_BASE_LO_0

Offset: 0x110
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_0_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxxx)

Bit	R/W Attribute	Reset	Description
31:12	RWRL	0x0	**MastBase**: Region Master Base Address. This field specifies bits 31:12 of the Base address for the region on the AXI Master interface. The Output address is generated using the following equation. OutputAddress[63:12] = (InputAddress[63:12] & Mask[N]) | (MastBase[N] & !Mask[N]).

APS_AST_REGION_0_MASTER_BASE_HI_0

Offset: 0x114
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_0_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W Attribute	Reset	Description
31:0	RWRL	0x0	**MastBase**: Region Master Base Address. This field specifies bits 63:32 of the Base address for the region on the AXI Master interface. The Output address is generated using the following equation. OutputAddress[63:12] = (InputAddress[63:12] & Mask[N]) | (MastBase[N] & !Mask[N]).

APS_AST_REGION_0_CONTROL_0

Offset: 0x118
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_0_SEC_CONTROL_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0xxx,xx00,000x,x0x0)

Bit	R/W Attribute	Reset	Description
19	RWGL_Region	0x0	**Physical**: Specifies how the StreamID is selected for a region match. **0** = VMIndx is used to select the StreamID **1** = PhysStreamID is used
18:15	RWRL	0x0	**VMIndex**: Specifies the VM Index used to select the Stream ID when (Physical == 0).

Bit	R/W Attribute	Reset	Description
9:5	RWCL	0x0	**CarveOutID**: Specifies the carveout ID for the region. his field specifies the state output on ast_master_a[w,r]user[15,11] for requests that matches the region.
2	RWRL	0x0	**Snoop**: Specifies if region accesses snoop the Main CPU caches. This bit controls the state output on ast_master_a[w,r]user[8] for requests that matches the region. 0 = Do not snoop request. 1 = Snoop request. **0** = DISABLE **1** = ENABLE
0	RW1	0x0	**Lock**: This bit prevents writes to this region registers. **0** = FALSE **1** = TRUE

APS_AST_REGION_1_SLAVE_BASE_LO_0

Offset: 0x120
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_1_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	R/W Attribute	Reset	Description
31:12	RWRL	0x0	**SlvBase**: Region Slave Base Address. This field specifies bits 31:12 of the Base address for the region on the AXI slave interface. This address is compared with the incoming slave address to determine if a region match occurs.
0	RWRL	0x0	**Enable**: Region Enable. This enables the translation region. 0 = Translation region disabled. 1 = Translation region enabled. **0** = FALSE **1** = TRUE

APS_AST_REGION_1_SLAVE_BASE_HI_0

Offset: 0x124
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: APS_AST_SCR_AST_REG_1_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W Attribute	Reset	Description
31:0	RWRL	0x0	**SlvBase**: Region Slave Base Address. This field specifies bits 63:32 of the Base address for the region on the AXI slave interface. This address is compared with the incoming slave address to determine if a region match occurs.

APS_AST_REGION_1_MASK_LO_0

Offset: 0x128
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_1_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxxx)

Bit	R/W Attribute	Reset	Description
31:12	RWRL	0x0	**Mask**: Region Mask Address[31:12]. This field is used to mask incoming address bits when performing the region compare. This field is also used to mask untranslated address bits when generating the output address.

APS_AST_REGION_1_MASK_HI_0

Offset: 0x12c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_1_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W Attribute	Reset	Description
31:0	RWRL	0x0	**Mask**: Region Mask Address[63:32]. This field is used to mask incoming address bits when performing the region compare. This field is also used to mask untranslated address bits when generating the output address.

APS_AST_REGION_1_MASTER_BASE_LO_0

Offset: 0x130
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_1_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxxx)

Bit	R/W Attribute	Reset	Description
31:12	RWRL	0x0	**MastBase**: Region Master Base Address. This field specifies bits 31:12 of the Base address for the region on the AXI Master interface. The Output address is generated using the following equation. OutputAddress[63:12] = (InputAddress[63:12] & Mask[N]) | (MastBase[N] & !Mask[N]).

APS_AST_REGION_1_MASTER_BASE_HI_0

Offset: 0x134
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_1_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W Attribute	Reset	Description
31:0	RWRL	0x0	**MastBase**: Region Master Base Address. This field specifies bits 63:32 of the Base address for the region on the AXI Master interface. The Output address is generated using the following equation. OutputAddress[63:12] = (InputAddress[63:12] & Mask[N]) | (MastBase[N] & !Mask[N]).

APS_AST_REGION_1_CONTROL_0

Offset: 0x138
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_1_SEC_CONTROL_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0xxx,xx00,000x,x0x0)

Bit	R/W Attribute	Reset	Description
19	RWGL_Region	0x0	**Physical**: Specifies how the StreamID is selected for a region match. **0** = VMIndx is used to select the StreamID **1** = PhysStreamID is used
18:15	RWRL	0x0	**VMIndex**: Specifies the VM Index used to select the Stream ID when (Physical == 0).

Bit	R/W Attribute	Reset	Description
9:5	RWCL	0x0	**CarveOutID**: Specifies the carveout ID for the region. This field specifies the state output on ast_master_a[w,r]user[15,11] for requests that matches the region.
2	RWRL	0x0	**Snoop**: Specifies if region accesses snoop the Main CPU caches. This bit controls the state output on ast_master_a[w,r]user[8] for requests that matches the region. 0 = Do not snoop request. 1 = Snoop request. **0** = DISABLE **1** = ENABLE
0	RW1	0x0	**Lock**: This bit prevents writes to this region registers. **0** = FALSE **1** = TRUE

APS_AST_REGION_2_SLAVE_BASE_LO_0

Offset: 0x140
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_2_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	R/W Attribute	Reset	Description
31:12	RWRL	0x0	**SlvBase**: Region Slave Base Address. This field specifies bits 31:12 of the Base address for the region on the AXI slave interface. This address is compared with the incoming slave address to determine if a region match occurs.
0	RWRL	0x0	**Enable**: Region Enable. This enables the translation region. 0 = Translation region disabled. 1 = Translation region enabled. **0** = FALSE **1** = TRUE

APS_AST_REGION_2_SLAVE_BASE_HI_0

Offset: 0x144
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: APS_AST_SCR_AST_REG_2_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W Attribute	Reset	Description
31:0	RWRL	0x0	**SlvBase**: Region Slave Base Address. This field specifies bits 63:32 of the Base address for the region on the AXI slave interface. This address is compared with the incoming slave address to determine if a region match occurs.

APS_AST_REGION_2_MASK_LO_0

Offset: 0x148
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_2_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxxx)

Bit	R/W Attribute	Reset	Description
31:12	RWRL	0x0	**Mask**: Region Mask Address[31:12]. This field is used to mask incoming address bits when performing the region compare. This field is also used to mask untranslated address bits when generating the output address.

APS_AST_REGION_2_MASK_HI_0

Offset: 0x14c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_2_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W Attribute	Reset	Description
31:0	RWRL	0x0	**Mask**: Region Mask Address[63:32]. This field is used to mask incoming address bits when performing the region compare. This field is also used to mask untranslated address bits when generating the output address.

APS_AST_REGION_2_MASTER_BASE_LO_0

Offset: 0x150
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_2_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxxx)

Bit	R/W Attribute	Reset	Description
31:12	RWRL	0x0	**MastBase**: Region Master Base Address. This field specifies bits 31:12 of the Base address for the region on the AXI Master interface. The Output address is generated using the following equation. OutputAddress[63:12] = (InputAddress[63:12] & Mask[N]) | (MastBase[N] & !Mask[N]).

APS_AST_REGION_2_MASTER_BASE_HI_0

Offset: 0x154
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_2_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W Attribute	Reset	Description
31:0	RWRL	0x0	**MastBase**: Region Master Base Address. This field specifies bits 63:32 of the Base address for the region on the AXI Master interface. The Output address is generated using the following equation. OutputAddress[63:12] = (InputAddress[63:12] & Mask[N]) | (MastBase[N] & !Mask[N]).

APS_AST_REGION_2_CONTROL_0

Offset: 0x158
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_2_SEC_CONTROL_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0xxx,xx00,000x,x0x0)

Bit	R/W Attribute	Reset	Description
19	RWGL_Region	0x0	**Physical**: Specifies how the StreamID is selected for a region match. **0** = VMIndx is used to select the StreamID **1** = PhysStreamID is used
18:15	RWRL	0x0	**VMIndex**: Specifies the VM Index used to select the Stream ID when (Physical == 0).

Bit	R/W Attribute	Reset	Description
9:5	RWCL	0x0	**CarveOutID**: Specifies the carveout ID for the region. This field specifies the state output on ast_master_a[w,r]user[15,11] for requests that matches the region
2	RWRL	0x0	**Snoop**: Specifies if region accesses snoop the Main CPU caches. This bit controls the state output on ast_master_a[w,r]user[8] for requests that matches the region. 0 = Do not snoop request. 1 = Snoop request. **0** = DISABLE **1** = ENABLE
0	RW1	0x0	**Lock**: This bit prevents writes to this region registers. **0** = FALSE **1** = TRUE

APS_AST_REGION_3_SLAVE_BASE_LO_0

Offset: 0x160
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_3_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	R/W Attribute	Reset	Description
31:12	RWRL	0x0	**SlvBase**: Region Slave Base Address. This field specifies bits 31:12 of the Base address for the region on the AXI slave interface. This address is compared with the incoming slave address to determine if a region match occurs.
0	RWRL	0x0	**Enable**: Region Enable: This enables the translation region. 0 = Translation region disabled. 1 = Translation region enabled. **0** = FALSE **1** = TRUE

APS_AST_REGION_3_SLAVE_BASE_HI_0

Offset: 0x164
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: APS_AST_SCR_AST_REG_3_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W Attribute	Reset	Description
31:0	RWRL	0x0	**SlvBase**: Region Slave Base Address. This field specifies bits 63:32 of the Base address for the region on the AXI slave interface. This address is compared with the incoming slave address to determine if a region match occurs.

APS_AST_REGION_3_MASK_LO_0

Offset: 0x168
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_3_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxxx)

Bit	R/W Attribute	Reset	Description
31:12	RWRL	0x0	**Mask**: Region Mask Address[31:12]. This field is used to mask incoming address bits when performing the region compare. This field is also used to mask untranslated address bits when generating the output address.

APS_AST_REGION_3_MASK_HI_0

Offset: 0x16c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_3_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W Attribute	Reset	Description
31:0	RWRL	0x0	**Mask**: Region Mask Address[63:32]. This field is used to mask incoming address bits when performing the region compare. This field is also used to mask untranslated address bits when generating the output address.

APS_AST_REGION_3_MASTER_BASE_LO_0

Offset: 0x170
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_3_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxxx)

Bit	R/W Attribute	Reset	Description
31:12	RWRL	0x0	**MastBase**: Region Master Base Address. This field specifies bits 31:12 of the Base address for the region on the AXI Master interface. The Output address is generated using the following equation. OutputAddress[63:12] = (InputAddress[63:12] & Mask[N]) | (MastBase[N] & !Mask[N]).

APS_AST_REGION_3_MASTER_BASE_HI_0

Offset: 0x174
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_3_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W Attribute	Reset	Description
31:0	RWRL	0x0	**MastBase**: Region Master Base Address: This field specifies bits 63:32 of the Base address for the region on the AXI Master interface. The Output address is generated using the following equation. OutputAddress[63:12] = (InputAddress[63:12] & Mask[N]) | (MastBase[N] & !Mask[N]).

APS_AST_REGION_3_CONTROL_0

Offset: 0x178
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_3_SEC_CONTROL_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0xxx,xx00,000x,x0x0)

Bit	R/W Attribute	Reset	Description
19	RWGL_Region	0x0	**Physical**: Specifies how the StreamID is selected for a region match. **0** = VMIndx is used to select the StreamID **1** = PhysStreamID is used
18:15	RWRL	0x0	**VMIndex**: Specifies the VM Index used to select the Stream ID when (Physical == 0).

Bit	R/W Attribute	Reset	Description
9:5	RWCL	0x0	**CarveOutID**: Specifies the carveout ID for the region. This field specifies the state output on ast_master_a[w,r]user[15,11] for requests that matches the region.
2	RWRL	0x0	**Snoop**: Specifies if region accesses snoop the Main CPU caches. This bit controls the state output on ast_master_a[w,r]user[8] for requests that matches the region. 0 = Do not snoop request. 1 = Snoop request. **0** = DISABLE **1** = ENABLE
0	RW1	0x0	**Lock**: This bit prevents writes to this region registers. **0** = FALSE **1** = TRUE

APS_AST_REGION_4_SLAVE_BASE_LO_0

Offset: 0x180
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_4_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	R/W Attribute	Reset	Description
31:12	RWRL	0x0	**SlvBase**: Region Slave Base Address. This field specifies bits 31:12 of the Base address for the region on the AXI slave interface. This address is compared with the incoming slave address to determine if a region match occurs.
0	RWRL	0x0	**Enable**: Region Enable. This enables the translation region. 0=Translation region disabled. 1=Translation region enabled. **0** = FALSE **1** = TRUE

APS_AST_REGION_4_SLAVE_BASE_HI_0

Offset: 0x184
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: APS_AST_SCR_AST_REG_4_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W Attribute	Reset	Description
31:0	RWRL	0x0	**SlvBase**: Region Slave Base Address. This field specifies bits 63:32 of the Base address for the region on the AXI slave interface. This address is compared with the incoming slave address to determine if a region match occurs.

APS_AST_REGION_4_MASK_LO_0

Offset: 0x188
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_4_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxxx)

Bit	R/W Attribute	Reset	Description
31:12	RWRL	0x0	**Mask**: Region Mask Address[31:12]. This field is used to mask incoming address bits when performing the region compare. This field is also used to mask untranslated address bits when generating the output address.

APS_AST_REGION_4_MASK_HI_0

Offset: 0x18c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_4_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W Attribute	Reset	Description
31:0	RWRL	0x0	**Mask**: Region Mask Address[63:32]. This field is used to mask incoming address bits when performing the region compare. This field is also used to mask untranslated address bits when generating the output address.

APS_AST_REGION_4_MASTER_BASE_LO_0

Offset: 0x190
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_4_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxxx)

Bit	R/W Attribute	Reset	Description
31:12	RWRL	0x0	**MastBase**: Region Master Base Address. This field specifies bits 31:12 of the Base address for the region on the AXI Master interface. The Output address is generated using the following equation. OutputAddress[63:12] = (InputAddress[63:12] & Mask[N]) | (MastBase[N] & !Mask[N]).

APS_AST_REGION_4_MASTER_BASE_HI_0

Offset: 0x194
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_4_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W Attribute	Reset	Description
31:0	RWRL	0x0	**MastBase**: Region Master Base Address. This field specifies bits 63:32 of the Base address for the region on the AXI Master interface. The Output address is generated using the following equation. OutputAddress[63:12] = (InputAddress[63:12] & Mask[N]) | (MastBase[N] & !Mask[N]).

APS_AST_REGION_4_CONTROL_0

Offset: 0x198
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_4_SEC_CONTROL_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0xxx,xx00,000x,x0x0)

Bit	R/W Attribute	Reset	Description
19	RWGL_Region	0x0	**Physical**: Specifies how the StreamID is selected for a region match. **0** = VMIndx is used to select the StreamID **1** = PhysStreamID is used
18:15	RWRL	0x0	**VMIndex**: Specifies the VM Index used to select the Stream ID when Physical=0.

Bit	R/W Attribute	Reset	Description
9:5	RWCL	0x0	**CarveOutID**: Specifies the carveout ID for the region. This field specifies the state output on ast_master_a[w,r]user[15,11] for requests that matches the region.
2	RWRL	0x0	**Snoop**: Specifies if region accesses snoop the Main CPU caches. This bit controls the state output on ast_master_a[w,r]user[8] for requests that matches the region. 0 = Do not snoop request. 1 = Snoop request. **0** = DISABLE **1** = ENABLE
0	RW1	0x0	**Lock**: This bit prevents writes to this region registers. **0** = FALSE **1** = TRUE

APS_AST_REGION_5_SLAVE_BASE_LO_0

Offset: 0x1a0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_5_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	R/W Attribute	Reset	Description
31:12	RWRL	0x0	**SlvBase**: Region Slave Base Address. This field specifies bits 31:12 of the Base address for the region on the AXI slave interface. This address is compared with the incoming slave address to determine if a region match occurs.
0	RWRL	0x0	**Enable**: Region Enable: This enables the translation region. 0 = Translation region disabled. 1 = Translation region enabled. **0** = FALSE **1** = TRUE

APS_AST_REGION_5_SLAVE_BASE_HI_0

Offset: 0x1a4
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: APS_AST_SCR_AST_REG_5_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W Attribute	Reset	Description
31:0	RWRL	0x0	**SlvBase**: Region Slave Base Address. This field specifies bits 63:32 of the Base address for the region on the AXI slave interface. This address is compared with the incoming slave address to determine if a region match occurs.

APS_AST_REGION_5_MASK_LO_0

Offset: 0x1a8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_5_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxxx)

Bit	R/W Attribute	Reset	Description
31:12	RWRL	0x0	**Mask**: Region Mask Address[31:12]. This field is used to mask incoming address bits when performing the region compare. This field is also used to mask untranslated address bits when generating the output address.

APS_AST_REGION_5_MASK_HI_0

Offset: 0x1ac
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_5_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W Attribute	Reset	Description
31:0	RWRL	0x0	**Mask**: Region Mask Address[63:32]. This field is used to mask incoming address bits when performing the region compare. This field is also used to mask untranslated address bits when generating the output address.

APS_AST_REGION_5_MASTER_BASE_LO_0

Offset: 0x1b0
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_5_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxxx)

Bit	R/W Attribute	Reset	Description
31:12	RWRL	0x0	**MastBase**: Region Master Base Address. This field specifies bits 31:12 of the Base address for the region on the AXI Master interface. The Output address is generated using the following equation. OutputAddress[63:12] = (InputAddress[63:12] & Mask[N]) | (MastBase[N] & !Mask[N]).

APS_AST_REGION_5_MASTER_BASE_HI_0

Offset: 0x1b4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_5_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W Attribute	Reset	Description
31:0	RWRL	0x0	**MastBase**: Region Master Base Address. This field specifies bits 63:32 of the Base address for the region on the AXI Master interface. The Output address is generated using the following equation. OutputAddress[63:12] = (InputAddress[63:12] & Mask[N]) | (MastBase[N] & !Mask[N]).

APS_AST_REGION_5_CONTROL_0

Offset: 0x1b8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_5_SEC_CONTROL_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0xxx,xx00,000x,x0x0)

Bit	R/W Attribute	Reset	Description
19	RWGL_Region	0x0	**Physical**: Specifies how the StreamID is selected for a region match. **0** = VMIndx is used to select the StreamID **1** = PhysStreamID is used

Bit	R/W Attribute	Reset	Description
18:15	RWRL	0x0	**VMIndex**: Specifies the VM Index used to select the Stream ID when Physical=0
9:5	RWCL	0x0	**CarveOutID**: Specifies the carveout ID for the region. This field specifies the state output on ast_master_a[w,r]user[15,11] for requests that matches the region.
2	RWRL	0x0	**Snoop**: Specifies if region accesses snoop the Main CPU caches. This bit controls the state output on ast_master_a[w,r]user[8] for requests that matches the region. 0 = Do not snoop request. 1 = Snoop request. **0** = DISABLE **1** = ENABLE
0	RW1	0x0	**Lock**: This bit prevents writes to this region registers. **0** = FALSE **1** = TRUE

APS_AST_REGION_6_SLAVE_BASE_LO_0

Offset: 0x1c0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_6_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	R/W Attribute	Reset	Description
31:12	RWRL	0x0	**SlvBase**: Region Slave Base Address. This field specifies bits 31:12 of the Base address for the region on the AXI slave interface. This address is compared with the incoming slave address to determine if a region match occurs.
0	RWRL	0x0	**Enable**: Region Enable. This enables the translation region. 0 = Translation region disabled. 1 = Translation region enabled. **0** = FALSE **1** = TRUE

APS_AST_REGION_6_SLAVE_BASE_HI_0

Offset: 0x1c4
Read/Write: R/W

Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_6_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W Attribute	Reset	Description
31:0	RWRL	0x0	**SlvBase**: Region Slave Base Address. This field specifies bits 63:32 of the Base address for the region on the AXI slave interface. This address is compared with the incoming slave address to determine if a region match occurs.

APS_AST_REGION_6_MASK_LO_0

Offset: 0x1c8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_6_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxxx)

Bit	R/W Attribute	Reset	Description
31:12	RWRL	0x0	**Mask**: Region Mask Address[31:12]. This field is used to mask incoming address bits when performing the region compare. This field is also used to mask untranslated address bits when generating the output address.

APS_AST_REGION_6_MASK_HI_0

Offset: 0x1cc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_6_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W Attribute	Reset	Description
31:0	RWRL	0x0	**Mask**: Region Mask Address[63:32]. This field is used to mask incoming address bits when performing the region compare. This field is also used to mask untranslated address bits when generating the output address.

APS_AST_REGION_6_MASTER_BASE_LO_0

Offset: 0x1d0

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_6_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxxx)

Bit	R/W Attribute	Reset	Description
31:12	RWRL	0x0	**MastBase**: Region Master Base Address. This field specifies bits 31:12 of the Base address for the region on the AXI Master interface. The Output address is generated using the following equation. OutputAddress[63:12] = (InputAddress[63:12] & Mask[N]) | (MastBase[N] & !Mask[N]).

APS_AST_REGION_6_MASTER_BASE_HI_0

Offset: 0x1d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_6_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W Attribute	Reset	Description
31:0	RWRL	0x0	**MastBase**: Region Master Base Address. This field specifies bits 63:32 of the Base address for the region on the AXI Master interface. The Output address is generated using the following equation. OutputAddress[63:12] = (InputAddress[63:12] & Mask[N]) | (MastBase[N] & !Mask[N]).

APS_AST_REGION_6_CONTROL_0

Offset: 0x1d8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_6_SEC_CONTROL_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0xxx,xx00,000x,x0x0)

Bit	R/W Attribute	Reset	Description
19	RWGL_Region	0x0	**Physical**: Specifies how the StreamID is selected for a region match. **0** = VMIndx is used to select the StreamID **1** = PhysStreamID is used

Bit	R/W Attribute	Reset	Description
18:15	RWRL	0x0	**VMIndex**: Specifies the VM Index used to select the Stream ID when (Physical = 0).
9:5	RWCL	0x0	**CarveOutID**: Specifies the carveout ID for the region. This field specifies the state output on ast_master_a[w,r]user[15,11] for requests that matches the region.
2	RWRL	0x0	**Snoop**: Specifies if region accesses snoop the Main CPU caches. This bit controls the state output on ast_master_a[w,r]user[8] for requests that matches the region. 0 = Do not snoop request. 1 = Snoop request. **0** = DISABLE **1** = ENABLE
0	RW1	0x0	**Lock**: This bit prevents writes to this region registers. **0** = FALSE **1** = TRUE

APS_AST_REGION_7_SLAVE_BASE_LO_0

Offset: 0x1e0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_7_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	R/W Attribute	Reset	Description
31:12	RWRL	0x0	**SlvBase**: Region Slave Base Address. This field specifies bits 31:12 of the Base address for the region on the AXI slave interface. This address is compared with the incoming slave address to determine if a region match occurs.
0	RWRL	0x0	**Enable**: Region Enable. This enables the translation region. 0 = Translation region disabled. 1 = Translation region enabled. **0** = FALSE **1** = TRUE

APS_AST_REGION_7_SLAVE_BASE_HI_0

Offset: 0x1e4
Read/Write: R/W

Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_7_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W Attribute	Reset	Description
31:0	RWRL	0x0	**SlvBase**: Region Slave Base Address. This field specifies bits 63:32 of the Base address for the region on the AXI slave interface. This address is compared with the incoming slave address to determine if a region match occurs.

APS_AST_REGION_7_MASK_LO_0

Offset: 0x1e8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_7_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxxx)

Bit	R/W Attribute	Reset	Description
31:12	RWRL	0x0	**Mask**: Region Mask Address[31:12]. This field is used to mask incoming address bits when performing the region compare. This field is also used to mask untranslated address bits when generating the output address.

APS_AST_REGION_7_MASK_HI_0

Offset: 0x1ec
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_7_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)0)

Bit	R/W Attribute	Reset	Description
31:0	RWRL	0x0	**Mask**: Region Mask Address[63:32]. This field is used to mask incoming address bits when performing the region compare. This field is also used to mask untranslated address bits when generating the output address.

APS_AST_REGION_7_MASTER_BASE_LO_0

Offset: 0x1f0

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_7_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxxx)

Bit	R/W Attribute	Reset	Description
31:12	RWRL	0x0	**MastBase**: Region Master Base Address. This field specifies bits 31:12 of the Base address for the region on the AXI Master interface. The Output address is generated using the following equation. OutputAddress[63:12] = (InputAddress[63:12] & Mask[N]) | (MastBase[N] & !Mask[N]).

APS_AST_REGION_7_MASTER_BASE_HI_0

Offset: 0x1f4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_7_SEC_CONTROL_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W Attribute	Reset	Description
31:0	RWRL	0x0	**MastBase**: Region Master Base Address: This field specifies bits 63:32 of the Base address for the region on the AXI Master interface. The Output address is generated using the following equation. OutputAddress[63:12] = (InputAddress[63:12] & Mask[N]) | (MastBase[N] & !Mask[N]).

APS_AST_REGION_7_CONTROL_0

Offset: 0x1f8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: APS_AST_SCR_AST_REG_7_SEC_CONTROL_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0xxx,xx00,000x,x0x0)

Bit	R/W Attribute	Reset	Description
19	RWGL_Region	0x0	**Physical**: Specifies how the StreamID is selected for a region match. **0** = VMIndx is used to select the StreamID **1** = PhysStreamID is used

Bit	R/W Attribute	Reset	Description
18:15	RWRL	0x0	**VMIndex**: Specifies the VM Index used to select the Stream ID when Physical=0
9:5	RWCL	0x0	**CarveOutID**: Specifies the carveout ID for the region. This field specifies the state output on ast_master_a[w,r]user[15,11] for requests that matches the region.
2	RWRL	0x0	**Snoop**: Specifies if region accesses snoop the Main CPU caches. This bit controls the state output on ast_master_a[w,r]user[8] for requests that matches the region. 0 = Do not snoop request. 1 = Snoop request. **0** = DISABLE **1** = ENABLE
0	RW1	0x0	**Lock**: This bit prevents writes to this region registers. **0** = FALSE **1** = TRUE

3.4 General Purpose Direct Memory Access (DMA) Engines

3.4.1 Overview

The system-on-chip (SoC) has eight instances of a General-Purpose DMA engine, all based on the same architecture, as follows:

1. General-Purpose Central DMA controller (GPC-DMA, also referred to as GPCDMA) placed on the Control Fabric. It has 32 fully programmable independent channels which can be programmed by different masters.
2. DMA engine as part of the BPMP subsystem (BPMP-DMA). It has four fully programmable independent channels.
3. DMA engine as part of the SCE subsystem (SCE-DMA). It has eight fully programmable independent channels.
4. DMA engine as part of the RCE subsystem (RCE-DMA). It has eight fully programmable independent channels.
5. DMA engine as part of the DCE subsystem (DCE-DMA). It has eight fully programmable independent channels.

6. DMA engine as part of the AO subsystem (AON-DMA). It has eight fully programmable independent channels. Note that AON-DMA has no safety support.
7. DMA engine as part of the PSC subsystem (PSC-DMA). It has four fully programmable independent channels.
8. DMA engine as part of the FSI subsystem (FSI-DMA). It has eight fully programmable independent channels.

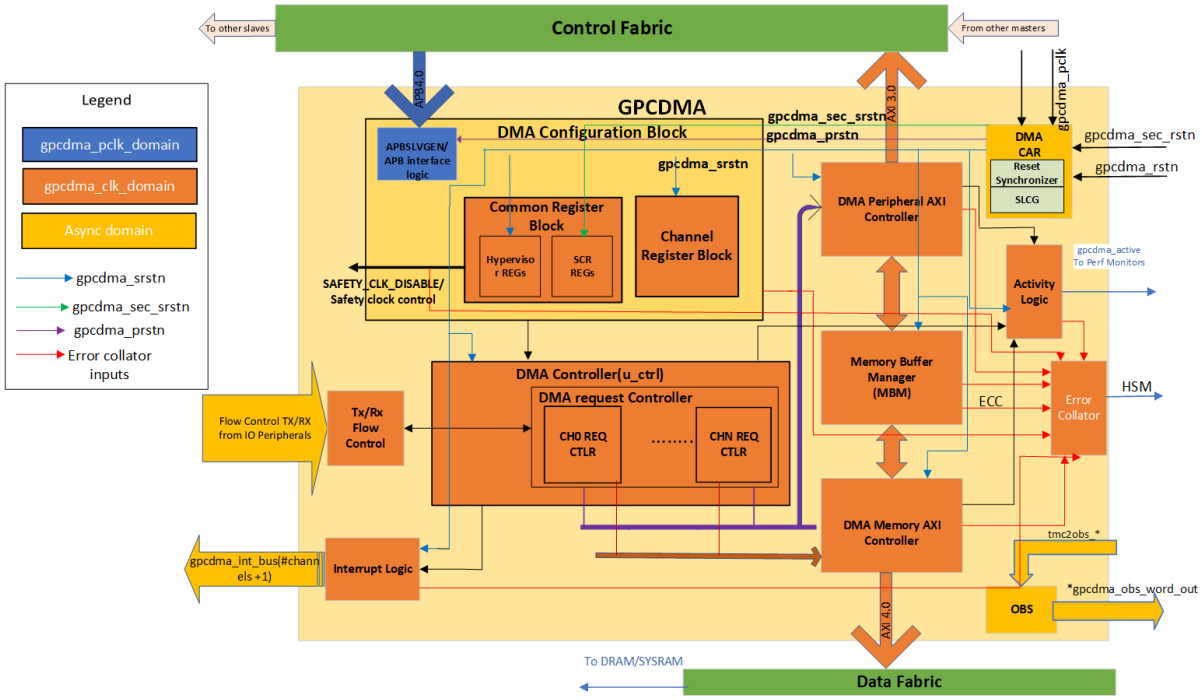
Each DMA engine provides the capability of either being used to write a block of data from Memory Mapped IO (MMIO) devices, to DRAM or SysRAM system memory via the Memory Subsystem (MSS), or to read a block of data from DRAM or SysRAM system memory to MMIO devices without any processor intervention. In addition, GPC-DMA can copy data to and from any memory mapped peripheral to system memory with and without flow control.

Note that APB is sometimes used in this document to refer to MMIO. The APB bus is still used in parts of the control fabric, but the General-Purpose DMA engine is not restricted to only APB devices.

This document generally describes the central GPC-DMA controller block, but the instances of the DMA engine are functionally alike except where noted.

GPC-DMA connections to the reset of the SoC are shown below.

Figure 3.8 GPC-DMA Top Level Connection Block Diagram



3.4.1.1 Features

- GPC-DMA engine serves APB clients that require memory access.
- GPC-DMA engine supports APB devices on any Fabric AXI2APB bridge.
- GPC-DMA engine can copy data from any addressable memory to/from DRAM/SysRAM.
- Firmware backwards-compatibility with legacy APBDMA drivers.
- Legacy flow-control support.
- Removes the dependency on APB bus by issuing pipelined AXI requests.
- Direct access to memory via AXICIF.
- DMA engine has a standard interface.
- Channels are independent from each other.
- CCPLEX to control different channels and cluster DMAs. They are managed by corresponding cluster Cortex-R5s or Cortex-R52 (in case of FSI).
- Virtualization support: each channel's registers are in an independent 64K aperture.
- Functional safety by duplication, ECC for shared buffer and error collator.

3.4.1.2 New Hardware Features

- Any pending transactions (requests or responses) are allowed to make forward progress and complete. The Busy bit indicates if there is any pending transaction still in flight.
- A lock bit per channel in the Hypervisor controlled Channel to block any malicious GuestOS from changing key registers that could end up in HOL blocking scenarios and potentially blocking another GuestOS channels from making forward progress.
- For Cortex-R5 DMAs only, the RDRSP queue (common response buffer for all channels) on the MSS read interface has been increased from a depth of 16 bytes (2 x 8 transactions) to a depth of 256 bytes (32 x 8 transactions). The increase in queue depth matches the maximum number of outstanding read responses in order to solve the DMA deadlock scenario of DRAM access and TCM access at the same time.

3.4.1.3 Support for Legacy MMIO DMA Hardware Features

The GPC-DMA includes the following capabilities from the MMIO DMA hardware in prior NVIDIA SoCs.

- Two modes of operation: single transfer (once) or continuous.
- Enable bit for each channel.
- Programmable burst sizes of one, four, eight, and 16 words.
- For memory, support for two and 16 word burst.
- Maximum transfer size is 1 GB per channel, with the minimum size being one word.
- Trigger and flow control mechanism support per channel. These are additional controls apart from channel enable on which the transfer depends. Trigger starts a channel to start the

transfer, and flow is used to proceed with every new burst transfer. These events are under software control or hardware control.
- Channel to channel trigger support, which is the ability to link up channels to start at the end of another channel's transfer, allowing scattering/gathering of physical memory.
- Interrupt enable at the end of transfer with the ability to mask or route to a desired processor.
- Wrap mode supported for all channels in Once mode.
- Round robin arbitration among channels at burst granularity.
- Direction bit to determine the direction of transfer MSS to MMIO or MMIO to MSS.
- Separate source address and destination address. MSS addresses are 40-bit wide.
- Wrap feature: enables the address to wrap back to starting address after N words of transfer. For example, if the address starts at 0x4 with a burst of four words, then the address would increment as 0x4, 0x8, 0xC, 0x10, 0x4, …. If disabled, it would be 0x4, 0x8, 0xC, 0x10, 0x14. Note that WRAP setting must be multiple of MMIO burst.

3.4.1.4 Supported Legacy Software Features

The GPC-DMA also includes these hardware features:

- AXI/PCLK with 1:1 ratio synchronous clock
- Three directional bits to determine the direction of transfer
  - SysRAM-DRAM to SDRAM-SysRAM
  - Fixed pattern write to SysRAM-SDRAM
- Programmable burst sizes of one, two, four, eight, 16 words on MMIO, two and 16 words on memory.
- Transfer size in words
- Byte enable support
- Interrupts per channel can be routed to CCPLEX, BPMP, SCE, or SPE. An additional interrupt for common space.
- Interrupt generation per channel after last burst write response (MMIO or Memory).
- Support TrustZone® and NV security privileges per channel for a given master.
- Error handling and DMA engine termination upon errors.
- Virtualization support by placing channel registers in a 64K aperture.
- Interrupt is generated when the last response data is received from the MSS for Rx mode, or once the last response data is received from the MMIO bus while reading from MSS (Tx mode).
- Memory to Memory DMA transfer
- The FIFO trigger levels (in the modules) need to be programmed so they do not lead to an overflow/underflow of the FIFO.
- Software programs all the registers of channel ensuring that the channel enable bit is disabled. Set the channel enable bit last.

- If channel is disabled while transfer is in progress, the transfer ends after ongoing burst is completed and an interrupt is generated (if that interrupt is enabled).
- Busy bit gets set as soon as the DMA channel is enabled and gets cleared after transfer is completed.
- Interrupts write 1 to clear.
- The default wraparound on the MMIO side is wrapping on one word. It prevents unnecessary address switching on the MMIO.
- The parameters of the channel must be programmed first (base address, wrap-around, etc.), then the control register of that channel is programmed. If the control register is programmed first, the current parameters of the DMA would be considered as the programmed values.

3.4.2 Functional Description

There are 32 channels in GPC-DMA. A DMA channel can transfer a specified range of data between a memory address space (SysRAM or external memory) and an MMIO address space. A DMA channel can also transfer data between a memory address space and another memory address space (Mem-to-Mem copy). The DMA controller follows a simple round robin arbitration scheme between the channels, starting with channel 0.

Each channel can have an independent burst transfer size programmed to one word, two words, four words, eight words, or 16 words. For Memory transfers, we only support two word and 16 word bursts. There is a corresponding read/write buffer in the memory buffer manager for each channel. There is also a corresponding buffer for each channel on the peripheral side.

3.4.2.1 DMA Functionality: Mem-MMIO

After programming the DMA channel with the starting MMIO and MSS address, the burst size and the total transfer byte count, the DMA engine can be enabled by setting the channel enable bit.

3.4.2.2 Peripheral Rx Mode: DMA Read from MMIO Peripheral to Memory

The burst size and the FIFO trigger levels in the MMIO slave need to be programmed such that they do not lead to an overflow/underflow of the FIFOs in the peripheral. This means that the controller FIFO thresholds need to be set correctly based on DMA request size. In flow control mode, the DMA engine waits for the flow control request from the peripheral controller to trigger a transfer to the DMA. Then DMA transfers that burst for the given channel.

If the burst size is eight, then a 32-byte request can be initiated on the memory interface. For some clients that have support like QSPI, a burst size of 16 words can be initiated to fill the 64-byte buffer in the buffer manager. The buffer control uses byte enable control for unaligned transfers and for the residual bytes if the remaining transfer is less than 32/64 bytes. A burst size of four is also supported, but in this case multiple bursts are needed to initiate the 32/64 byte transfers on

the memory interface. Once an Rx request has been initiated to the DMA by the peripheral, the DMA engine initiates a read request from the peripheral FIFO.

In case the peripheral is fast enough to fill in the second burst and to increase the DMA engine performance, the Peripheral Control block has the option of initiating two outstanding read requests to transfer bursts from the peripheral FIFO to system memory. The number of MMIO outstanding requests is set in channel register space.

3.4.2.3 Peripheral Tx Mode: DMA Write from Memory to MMIO

The burst size and the FIFO trigger levels in the MMIO slave need to be programmed so they do not lead to an overflow/underflow of the FIFOs in the MMIO client. By default, the DMA engine always initiates a 64-byte read from the MSS for the best system memory utilization. Upon receiving the read data in the buffer, and in case the burst size programmed is eight, then a 32-byte data transfer can occur to the MMIO slave. For clients that support a burst size of 16 words, then all 64 bytes can be transferred.

Once a Tx request has been initiated to the DMA by the peripheral, the DMA engine initiates a read request from memory. If the peripheral is fast enough to read in the current burst request and to increase the DMA engine performance, the Peripheral Control block has the option of initiating two outstanding write requests to transfer bursts from the system memory to the peripheral FIFO. In this case, the second write request is queued in the bridge until the peripheral is ready to consume the second burst. In this mode, the Peripheral Control block has to keep track of the outstanding requests and their responses relative to the total transfer byte count needed. The number of MMIO outstanding requests is set in channel register space.

3.4.2.4 MEM-MEM DMA Mode

MEM-MEM mode can read and write channels that are used to copy data from one memory location to another. The read channel keeps track of the total transfer size. The RD_MEM engine initiates multiple outstanding read requests to the MSS for higher bandwidth. Once the read data arrives, the RD_MEM block copies this data to the corresponding write buffer in the WR_MEM block. The WR_MEM block in return sends the write data to system memory with a different address. This process continues until all data is copied.

3.4.3 Programming Guidelines

3.4.3.1 Main Programming Steps

All the registers of a channel need to be programmed before the Channel Enable bit is set.

- Program the MSS Starting Address and MMIO Starting address in GPC-DMA-X Source Address Pointer and GPC-DMA-X Destination Address Pointer registers.

- Program the required BURST size, WRAP word window size, and the GPC-DMA-X MSS Address Sequencer register. The MSS BUS width is fixed to 64-bit bus.
- Program the required MMIO_BUS_WIDTH (as the peripheral) and WRAP word window size in the MMIO Address Sequencer register.
- Program the number of words to be transferred in the GPCDMA_CHANNEL_CH<X>_WCOUNT_0 register. This register needs to be programmed with number of words to be transferred -1.
- Program the Trigger in GPC-DMA-X Control-Extended register.
- Program the Interrupt option, FC mode, DMA transfer direction, Transfer mode, and Flow Enable in GPCDMA_CHANNEL_CHx_CSR_0 register. Write the channel Enable bit in the same register.
- Whenever the Channel ENB bit is enabled, the DMA starts the data transfer. The security attributes after writing Channel Enable are latched and used by the current DMA engine for peripheral access. MSS security is programmed under GPC-DMA-X MSS Address Sequencer.
- Each channel's status is observed by polling the GPCDMA_CHANNEL_CH<X>_STA register. The number of words remaining to be transferred are in the GPCDMA_CHANNEL_CH<X>_DMA_WORD_TRA register.
- Tx/Rx Flow/Trigger requesters are programmed in the GPCDMA_CHANNEL_CHx_CSR_0 register.

3.4.3.2 GPC-DMA Address Space

For different DMA register programming, use the following address rules for IP base and IP register:

- For GPC-DMA: NV_ADDRESS_MAP_GPCDMA_BASE + GPCDMA_CHANNEL_XXXXXXX
- For BPMP-DMA: NV_ADDRESS_MAP_BPMP_DMA_BASE + GPCDMA_FLV_4CH_XXXXXXX
- For SCE-DMA: NV_ADDRESS_MAP_SCE_DMA_BASE + GPCDMA_FLV_8CH_XXXXXXX
- For RCE-DMA: NV_ADDRESS_MAP_RCE_DMA_BASE + GPCDMA_FLV_8CH_XXXXXXX
- For DCE-DMA: NV_ADDRESS_MAP_DCE_DMA_BASE + GPCDMA_FLV_8CH_XXXXXXX
- For AO-DMA: NV_ADDRESS_MAP_AON_DMA_BASE + GPCDMA_FLV_8CH_NON_SAFE_XXXXXXX
- For PSC-DMA: NV_ADDRESS_MAP_PSC_DMA_BASE + GPCDMA_FLV_4CH_XXXXXXX
- For FSI-DMA: NV_ADDRESS_MAP_FSI_DMA_BASE + GPCDMA_FLV_8CH_XXXXXXX

3.4.3.3 Pause Mode

- Setting the CHANNEL_PAUSE bit in GPCDMA_CHANNEL_CHx_CSR_0 register blocks any controller requests from being serviced. All transfers that are in progress are allowed to continue and make forward progress. Clearing CHANNEL_PAUSE resumes the transfers.
- Do not disable a channel when a channel busy bit is asserted (BSY0-31). Pause mode flushes out all pending requests from GPC-DMA and any inflight data on MSS and MMIO bus.

- Upon pausing the channel, the programming sequence is:
  - Software waits for the DMA channel to be idle (busy bit deasserted).
  - Software now can disable the paused channel or un-pause the channel.
  - To restart the channel after disable, the pause bit must be cleared first.
  - In unaligned source/destination address cases, the pause mode does not flush out the data correctly.
  - If there was a mismatch in burst sizes between MMIO and MSS (MMIO burst > MSS burst), the pause mode does not flush out data correctly.
  - If an error occurred during pause mode, then an error interrupt is asserted and software shall reset the DMA engine.
- Maximum latency for waiting for MSS traffic from DMA to be committed is 100 µs.

Typically, all transactions have been committed by this time in pause mode.

3.4.3.4 Abrupt Channel Disable

If channel ENB is disabled abruptly while a transfer is in progress, the transfer ends after completing any burst sequence that is in progress. GPC-DMA does not flush out remaining data in local buffers.

3.4.3.5 Busy Bit

The Busy bit is read only and gets set when there is any inflight transaction either internally to DMA, on the MMIO interface, or on the MSS interface. The bit is cleared by Hardware after all pending transfers are completed.

3.4.3.6 Enable (ENB) Bit

- GPC-DMA channel enable bit ENB in GPCDMA_CHANNEL_CHX_CSR_0 is auto cleared after receiving an in-band error. In that case, the enable bit is cleared after all pending transactions are completed.
- GPC-DMA channel enable bit ENB in GPCDMA_CHANNEL_CHX_CSR_0 is auto cleared after all DMA transactions are completed in single mode. In that case, the enable bit is cleared after all pending transactions are completed. For re-use cases, software must enable the bit after reprogramming DMA.

3.4.3.7 DMA Interrupt

Interrupts are write-1-to-clear, i.e., interrupt bit is cleared when the value of write data corresponding to the bit position of the interrupt bit is 1.

3.4.3.8 Controller FIFO Size

The APB burst size and the FIFO trigger levels (in the peripheral controller) need to be programmed so they do not lead to an overflow/underflow of the FIFOs in the APB client. One limitation is that they need to be the same.

3.4.3.9 Transfer Alignment

- MSS requests are always aligned to 4 Bytes (lower 2 bits are zeros).
- MMIO requests are always aligned to 4 Bytes (lower 2 bits are zeros).
- When flow control is enabled, the number of words to be transferred must always be a multiple of the burst size/MMIO trigger level.
- The DMA wraparound needs to be programmed keeping in mind the address that is programmed in the BM start address and the burst size.

3.4.3.10 WCOUNT Register

If the DMA channel is enabled, then source/destination addresses BCOUNT can be reprogrammed during data transfer and after enabling the channel. WCOUNT can be a multiple of burst sizes:

- In case of Mem2Mem copy, Word Count could be any nonzero number within 1 GB.
- In case of mem2mmio or mmio2mem copy, Word Count should be a multiple of I/O burst size.

3.4.3.11 Continuous Mode

In continuous mode single buffer mode, software has two separate buffers that are maintained by software to emulate the hardware ping pong buffer. In this mode, software enables the DMA with the ping-buffer address and then reprograms the DMA with pong buffer after enabling the DMA. The DMA registers are shadowed (latched) every time upon entering the continuous cycle. The register programming can be done for the pong buffer either after enabling the channel (for the first reprogramming) or receiving an interrupt (for any subsequent reprogramming). Software has requested 15 ms interrupt latency as the worst case requirement.

If interrupt latency is not guaranteed, then pausing the DMA channel is needed before reprogramming the new address/word count. The pause mode must be done before receiving an end of transfer (EoT) interrupt that guarantees it does not re-transfer to the same buffer.

If DMA channel in Rx mode is abruptly disabled in a continuous mode during a transfer, then:

- For this mode, use FC_mode = 0
- Set MMIO burst size = 16W
- If the I/O controller generates an interrupt, the Rx request line should not be asserted.
- The last MSS request should be committed to MSS before disabling the channel.

- As a precautionary measure, the CPU should disable this channel after ensuring there is no DMA activity on MSS or MMIO by reading the DMA_IO_MC_ACTIVE status register under GPCDMA_COMMON_DMA_ACTIVE_0.
- Any residual words in the I/O controller FIFO should be cleared by host.

3.4.3.12 HOL Blocking

Since DBB does not support OOO for all GPC-DMA channels, and if there is more than one channel running in either Tx or Rx mode, a HOL blocking can occur on DBB if the outstanding request size is more than two for any of these channels. For BPMP-DMA/GPC-DMA only, if more than two channels are used, then FC_MODE = 0x3 and MMIO_BURST = 64B should never be programmed.

3.4.3.13 Non-continuous SPI Mode

For non-continuous SPI mode:

- The external device might stop sending data after some logical boundary, so software programs GPCDMA for the maximum possible data transfer size based upon use case.
- SPI is programmed for 4B/1W Rx mode as trigger level.
- GPCDMA channel is programmed for 4B/1W MMIO burst size.
- GPCDMA channel is programmed for 64B/16W MSS burst size (default value in spec, so no need to program explicitly).
- SPI generates an interrupt to the CPU when the external device stops sending data.
- Based upon this interrupt, DMA CH_BSY = 1 *&& SPI RX_FIFO is empty,* write DMA Channel Pause Bit and wait for CH_BSY status to go low.
- When CH_BSY bit is 0, it is expected that the SPI Transfer Count Status is equal to DMA_BYTE Count Status register.
- If SPI Transfer count status < DMA_BYTE Count status by 1B/2B/3B, discard that data before providing it to consumer.

3.4.4 GPC-DMA Registers

3.4.4.1 COMMON Registers

GPCDMA_COMMON_DMA_CHAN_STA_0
Offset: 0x0
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_DMA_RO_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**CH31**
30	0x0	**CH30**
29	0x0	**CH29**
28	0x0	**CH28**
27	0x0	**CH27**
26	0x0	**CH26**
25	0x0	**CH25**
24	0x0	**CH24**
23	0x0	**CH23**
22	0x0	**CH22**
21	0x0	**CH21**
20	0x0	**CH20**
19	0x0	**CH19**
18	0x0	**CH18**
17	0x0	**CH17**
16	0x0	**CH16**
15	0x0	**CH15**
14	0x0	**CH14**
13	0x0	**CH13**
12	0x0	**CH12**
11	0x0	**CH11**
10	0x0	**CH10**
9	0x0	**CH9**
8	0x0	**CH8**
7	0x0	**CH7**
6	0x0	**CH6**
5	0x0	**CH5**
4	0x0	**CH4**
3	0x0	**CH3**

Bit	Reset	Description
2	0x0	**CH2**
1	0x0	**CH1**
0	0x0	**CH0**

GPCDMA_COMMON_REQUESTORS_TX_0

Offset: 0x4
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_DMA_RO_0
Reset: 0x00000000 (0b00xx,00x0,0000,0x00,0x00,xx00,x00x,0000)

Bit	Reset	Description
31	0x0	**I2C9**
30	0x0	**I2C6**
27	0x0	**I2C7**
26	0x0	**I2C4**
24	0x0	**I2C5**
23	0x0	**I2C3**
22	0x0	**I2C2**
21	0x0	**I2C**
20	0x0	**UARTE**
19	0x0	**UARTD**
17	0x0	**SPI3**
16	0x0	**SPI2**
15	0x0	**SPI1**
13	0x0	**UARTH**
12	0x0	**UARTF**
9	0x0	**UARTB**
8	0x0	**UARTA**
6	0x0	**QSPI1**
5	0x0	**QSPI0**

Bit	Reset	Description
3	0x0	**UARTC**
2	0x0	**UARTG**
1	0x0	**I2C10**
0	0x0	**I2C8**

GPCDMA_COMMON_REQUESTORS_RX_0

Offset: 0x8
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_DMA_RO_0
Reset: 0x00000000 (0b00xx,00x0,0000,0x00,0x00,xx00,x00x,0000)

Bit	Reset	Description
31	0x0	**I2C9**
30	0x0	**I2C6**
27	0x0	**I2C7**
26	0x0	**I2C4**
24	0x0	**I2C5**
23	0x0	**I2C3**
22	0x0	**I2C2**
21	0x0	**I2C**
20	0x0	**UARTE**
19	0x0	**UARTD**
17	0x0	**SPI3**
16	0x0	**SPI2**
15	0x0	**SPI1**
13	0x0	**UARTH**
12	0x0	**UARTF**
9	0x0	**UARTB**
8	0x0	**UARTA**
6	0x0	**QSPI1**

Bit	Reset	Description
5	0x0	**QSPI0**
3	0x0	**UARTC**
2	0x0	**UARTG**
1	0x0	**I2C10**
0	0x0	**I2C8**

GPCDMA_COMMON_COMMON_ERROR_STA_0

Offset: 0xc
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_COMMON_CHANNEL_ERROR_STA_0

Offset: 0x10
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_COMMON_CHANNEL_TRIG_REG_0

Offset: 0x18
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_DMA_RO_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**CH31**

Bit	Reset	Description
30	0x0	**CH30**
29	0x0	**CH29**
28	0x0	**CH28**
27	0x0	**CH27**
26	0x0	**CH26**
25	0x0	**CH25**
24	0x0	**CH24**
23	0x0	**CH23**
22	0x0	**CH22**
21	0x0	**CH21**
20	0x0	**CH20**
19	0x0	**CH19**
18	0x0	**CH18**
17	0x0	**CH17**
16	0x0	**CH16**
15	0x0	**CH15**
14	0x0	**CH14**
13	0x0	**CH13**
12	0x0	**CH12**
11	0x0	**CH11**
10	0x0	**CH10**
9	0x0	**CH9**
8	0x0	**CH8**
7	0x0	**CH7**
6	0x0	**CH6**
5	0x0	**CH5**
4	0x0	**CH4**
3	0x0	**CH3**
2	0x0	**CH2**

Bit	Reset	Description
1	0x0	**CH1**
0	0x0	**CH0**

GPCDMA_COMMON_MASKED_INTR_REG_0

Offset: 0x1c
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_DMA_RO_0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31	0x1	**CH31**
30	0x1	**CH30**
29	0x1	**CH29**
28	0x1	**CH28**
27	0x1	**CH27**
26	0x1	**CH26**
25	0x1	**CH25**
24	0x1	**CH24**
23	0x1	**CH23**
22	0x1	**CH22**
21	0x1	**CH21**
20	0x1	**CH20**
19	0x1	**CH19**
18	0x1	**CH18**
17	0x1	**CH17**
16	0x1	**CH16**
15	0x1	**CH15**
14	0x1	**CH14**
13	0x1	**CH13**
12	0x1	**CH12**

Bit	Reset	Description
11	0x1	**CH11**
10	0x1	**CH10**
9	0x1	**CH9**
8	0x1	**CH8**
7	0x1	**CH7**
6	0x1	**CH6**
5	0x1	**CH5**
4	0x1	**CH4**
3	0x1	**CH3**
2	0x1	**CH2**
1	0x1	**CH1**
0	0x1	**CH0**

GPCDMA_COMMON_CHANNEL_INTR_STA_0

Offset: 0x20
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_DMA_RO_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**CH31**
30	0x0	**CH30**
29	0x0	**CH29**
28	0x0	**CH28**
27	0x0	**CH27**
26	0x0	**CH26**
25	0x0	**CH25**
24	0x0	**CH24**
23	0x0	**CH23**
22	0x0	**CH22**

Bit	Reset	Description
21	0x0	**CH21**
20	0x0	**CH20**
19	0x0	**CH19**
18	0x0	**CH18**
17	0x0	**CH17**
16	0x0	**CH16**
15	0x0	**CH15**
14	0x0	**CH14**
13	0x0	**CH13**
12	0x0	**CH12**
11	0x0	**CH11**
10	0x0	**CH10**
9	0x0	**CH9**
8	0x0	**CH8**
7	0x0	**CH7**
6	0x0	**CH6**
5	0x0	**CH5**
4	0x0	**CH4**
3	0x0	**CH3**
2	0x0	**CH2**
1	0x0	**CH1**
0	0x0	**CH0**

GPCDMA_COMMON_COMMON_INTR_0

Offset: 0x24
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	R/W	Reset	Description
4	RO	0x0	**RAW_INTR_STATUS**
3	RO	0x0	**IRQ_INTR_STATUS**
2	RW	0x0	**IS_EOC**
1	RW	0x0	**INTR_MASK**
0	RW	0x0	**IE_EOC**

GPCDMA_COMMON_CHANNEL_REG_LOCK_0

Offset: 0x28
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**REGLOCK**

GPCDMA_COMMON_CH0_PERI_ID_MASK_0

Offset: 0x80
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH0_PERI_ID_MASK**

GPCDMA_COMMON_CH1_PERI_ID_MASK_0

Offset: 0x84
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH1_PERI_ID_MASK**

GPCDMA_COMMON_CH2_PERI_ID_MASK_0

Offset: 0x88
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH2_PERI_ID_MASK**

GPCDMA_COMMON_CH3_PERI_ID_MASK_0

Offset: 0x8c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH3_PERI_ID_MASK**

GPCDMA_COMMON_CH4_PERI_ID_MASK_0

Offset: 0x90
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH4_PERI_ID_MASK**

GPCDMA_COMMON_CH5_PERI_ID_MASK_0

Offset: 0x94
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH5_PERI_ID_MASK**

GPCDMA_COMMON_CH6_PERI_ID_MASK_0

Offset: 0x98
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH6_PERI_ID_MASK**

GPCDMA_COMMON_CH7_PERI_ID_MASK_0

Offset: 0x9c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH7_PERI_ID_MASK**

GPCDMA_COMMON_CH8_PERI_ID_MASK_0

Offset: 0xa0
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH8_PERI_ID_MASK**

GPCDMA_COMMON_CH9_PERI_ID_MASK_0

Offset: 0xa4
Read/Write: R/W

Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH9_PERI_ID_MASK**

GPCDMA_COMMON_CH10_PERI_ID_MASK_0

Offset: 0xa8
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH10_PERI_ID_MASK**

GPCDMA_COMMON_CH11_PERI_ID_MASK_0

Offset: 0xac
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH11_PERI_ID_MASK**

GPCDMA_COMMON_CH12_PERI_ID_MASK_0

Offset: 0xb0
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH12_PERI_ID_MASK**

GPCDMA_COMMON_CH13_PERI_ID_MASK_0

Offset: 0xb4
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH13_PERI_ID_MASK**

GPCDMA_COMMON_CH14_PERI_ID_MASK_0

Offset: 0xb8
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH14_PERI_ID_MASK**

GPCDMA_COMMON_CH15_PERI_ID_MASK_0

Offset: 0xbc
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH15_PERI_ID_MASK**

GPCDMA_COMMON_CH16_PERI_ID_MASK_0

Offset: 0xc0
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH16_PERI_ID_MASK**

GPCDMA_COMMON_CH17_PERI_ID_MASK_0

Offset: 0xc4
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH17_PERI_ID_MASK**

GPCDMA_COMMON_CH18_PERI_ID_MASK_0

Offset: 0xc8
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH18_PERI_ID_MASK**

GPCDMA_COMMON_CH19_PERI_ID_MASK_0

Offset: 0xcc
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH19_PERI_ID_MASK**

GPCDMA_COMMON_CH20_PERI_ID_MASK_0

Offset: 0xd0
Read/Write: R/W

Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH20_PERI_ID_MASK**

GPCDMA_COMMON_CH21_PERI_ID_MASK_0

Offset: 0xd4
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH21_PERI_ID_MASK**

GPCDMA_COMMON_CH22_PERI_ID_MASK_0

Offset: 0xd8
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH22_PERI_ID_MASK**

GPCDMA_COMMON_CH23_PERI_ID_MASK_0

Offset: 0xdc
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH23_PERI_ID_MASK**

GPCDMA_COMMON_CH24_PERI_ID_MASK_0

Offset: 0xe0
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH24_PERI_ID_MASK**

GPCDMA_COMMON_CH25_PERI_ID_MASK_0

Offset: 0xe4
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH25_PERI_ID_MASK**

GPCDMA_COMMON_CH26_PERI_ID_MASK_0

Offset: 0xe8
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH26_PERI_ID_MASK**

GPCDMA_COMMON_CH27_PERI_ID_MASK_0

Offset: 0xec
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH27_PERI_ID_MASK**

GPCDMA_COMMON_CH28_PERI_ID_MASK_0

Offset: 0xf0
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH28_PERI_ID_MASK**

GPCDMA_COMMON_CH29_PERI_ID_MASK_0

Offset: 0xf4
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH29_PERI_ID_MASK**

GPCDMA_COMMON_CH30_PERI_ID_MASK_0

Offset: 0xf8
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH30_PERI_ID_MASK**

GPCDMA_COMMON_CH31_PERI_ID_MASK_0

Offset: 0xfc
Read/Write: R/W

Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH31_PERI_ID_MASK**

GPCDMA_COMMON_PER0_PERI_ADDR_0
Offset: 0x100
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER0_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done
15:0	0x0	**PER0_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER1_PERI_ADDR_0
Offset: 0x104
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER1_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done
15:0	0x0	**PER1_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER2_PERI_ADDR_0
Offset: 0x108

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER2_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done
15:0	0x0	**PER2_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER3_PERI_ADDR_0

Offset: 0x10c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER3_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done
15:0	0x0	**PER3_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER4_PERI_ADDR_0

Offset: 0x110
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER4_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done

Bit	Reset	Description
15:0	0x0	**PER4_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER5_PERI_ADDR_0

Offset: 0x114
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER5_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done
15:0	0x0	**PER5_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER6_PERI_ADDR_0

Offset: 0x118
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER6_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done
15:0	0x0	**PER6_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER7_PERI_ADDR_0

Offset: 0x11c
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER7_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done
15:0	0x0	**PER7_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER8_PERI_ADDR_0

Offset: 0x120
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER8_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done
15:0	0x0	**PER8_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER9_PERI_ADDR_0

Offset: 0x124
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER9_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done
15:0	0x0	**PER9_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER10_PERI_ADDR_0

Offset: 0x128
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER10_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done
15:0	0x0	**PER10_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER11_PERI_ADDR_0

Offset: 0x12c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER11_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done
15:0	0x0	**PER11_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER12_PERI_ADDR_0

Offset: 0x130
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER12_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done
15:0	0x0	**PER12_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER13_PERI_ADDR_0

Offset: 0x134
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER13_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done
15:0	0x0	**PER13_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER14_PERI_ADDR_0

Offset: 0x138
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER14_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done
15:0	0x0	**PER14_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER15_PERI_ADDR_0

Offset: 0x13c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER15_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done
15:0	0x0	**PER15_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER16_PERI_ADDR_0

Offset: 0x140
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER16_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done
15:0	0x0	**PER16_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER17_PERI_ADDR_0

Offset: 0x144
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER17_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done

Bit	Reset	Description
15:0	0x0	**PER17_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER18_PERI_ADDR_0

Offset: 0x148
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER18_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done
15:0	0x0	**PER18_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER19_PERI_ADDR_0

Offset: 0x14c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER19_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done
15:0	0x0	**PER19_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER20_PERI_ADDR_0

Offset: 0x150
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER20_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done
15:0	0x0	**PER20_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER21_PERI_ADDR_0

Offset: 0x154
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER21_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done
15:0	0x0	**PER21_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER22_PERI_ADDR_0

Offset: 0x158
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER22_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done
15:0	0x0	**PER22_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER23_PERI_ADDR_0

Offset: 0x15c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER23_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done
15:0	0x0	**PER23_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER24_PERI_ADDR_0

Offset: 0x160
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER24_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done
15:0	0x0	**PER24_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER25_PERI_ADDR_0

Offset: 0x164
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER25_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done
15:0	0x0	**PER25_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER26_PERI_ADDR_0

Offset: 0x168
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER26_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done
15:0	0x0	**PER26_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER27_PERI_ADDR_0

Offset: 0x16c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER27_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done
15:0	0x0	**PER27_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER28_PERI_ADDR_0

Offset: 0x170
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER28_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done
15:0	0x0	**PER28_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER29_PERI_ADDR_0

Offset: 0x174
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER29_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done
15:0	0x0	**PER29_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER30_PERI_ADDR_0

Offset: 0x178
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER30_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done

Bit	Reset	Description
15:0	0x0	**PER30_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_PER31_PERI_ADDR_0

Offset: 0x17c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPCDMA_SCR_SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PER31_PERI_START_ADDR**: PeripheralX start address set by Hypervisor (upper 16 address bits @ 64K boundary). If base address is set to zero then no address checking is done
15:0	0x0	**PER31_PERI_OFFSET**: PeripheralX range size set by Hypervisor (address offset)

GPCDMA_COMMON_CH0_STREAM_ID0_MASK_0

Offset: 0x180
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH0_STREAM_ID0_MASK**

GPCDMA_COMMON_CH1_STREAM_ID0_MASK_0

Offset: 0x184
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH1_STREAM_ID0_MASK**

GPCDMA_COMMON_CH2_STREAM_ID0_MASK_0

Offset: 0x188
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH2_STREAM_ID0_MASK**

GPCDMA_COMMON_CH3_STREAM_ID0_MASK_0

Offset: 0x18c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH3_STREAM_ID0_MASK**

GPCDMA_COMMON_CH4_STREAM_ID0_MASK_0

Offset: 0x190
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH4_STREAM_ID0_MASK**

GPCDMA_COMMON_CH5_STREAM_ID0_MASK_0

Offset: 0x194
Read/Write: R/W

Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH5_STREAM_ID0_MASK**

GPCDMA_COMMON_CH6_STREAM_ID0_MASK_0

Offset: 0x198
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH6_STREAM_ID0_MASK**

GPCDMA_COMMON_CH7_STREAM_ID0_MASK_0

Offset: 0x19c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH7_STREAM_ID0_MASK**

GPCDMA_COMMON_CH8_STREAM_ID0_MASK_0

Offset: 0x1a0
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH8_STREAM_ID0_MASK**

GPCDMA_COMMON_CH9_STREAM_ID0_MASK_0

Offset: 0x1a4
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH9_STREAM_ID0_MASK**

GPCDMA_COMMON_CH10_STREAM_ID0_MASK_0

Offset: 0x1a8
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH10_STREAM_ID0_MASK**

GPCDMA_COMMON_CH11_STREAM_ID0_MASK_0

Offset: 0x1ac
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH11_STREAM_ID0_MASK**

GPCDMA_COMMON_CH12_STREAM_ID0_MASK_0

Offset: 0x1b0
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH12_STREAM_ID0_MASK**

GPCDMA_COMMON_CH13_STREAM_ID0_MASK_0

Offset: 0x1b4
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH13_STREAM_ID0_MASK**

GPCDMA_COMMON_CH14_STREAM_ID0_MASK_0

Offset: 0x1b8
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH14_STREAM_ID0_MASK**

GPCDMA_COMMON_CH15_STREAM_ID0_MASK_0

Offset: 0x1bc
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH15_STREAM_ID0_MASK**

GPCDMA_COMMON_CH16_STREAM_ID0_MASK_0

Offset: 0x1c0
Read/Write: R/W

Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH16_STREAM_ID0_MASK**

GPCDMA_COMMON_CH17_STREAM_ID0_MASK_0

Offset: 0x1c4
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH17_STREAM_ID0_MASK**

GPCDMA_COMMON_CH18_STREAM_ID0_MASK_0

Offset: 0x1c8
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH18_STREAM_ID0_MASK**

GPCDMA_COMMON_CH19_STREAM_ID0_MASK_0

Offset: 0x1cc
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH19_STREAM_ID0_MASK**

GPCDMA_COMMON_CH20_STREAM_ID0_MASK_0

Offset: 0x1d0
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH20_STREAM_ID0_MASK**

GPCDMA_COMMON_CH21_STREAM_ID0_MASK_0

Offset: 0x1d4
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH21_STREAM_ID0_MASK**

GPCDMA_COMMON_CH22_STREAM_ID0_MASK_0

Offset: 0x1d8
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH22_STREAM_ID0_MASK**

GPCDMA_COMMON_CH23_STREAM_ID0_MASK_0

Offset: 0x1dc
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH23_STREAM_ID0_MASK**

GPCDMA_COMMON_CH24_STREAM_ID0_MASK_0

Offset: 0x1e0
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH24_STREAM_ID0_MASK**

GPCDMA_COMMON_CH25_STREAM_ID0_MASK_0

Offset: 0x1e4
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH25_STREAM_ID0_MASK**

GPCDMA_COMMON_CH26_STREAM_ID0_MASK_0

Offset: 0x1e8
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH26_STREAM_ID0_MASK**

GPCDMA_COMMON_CH27_STREAM_ID0_MASK_0

Offset: 0x1ec
Read/Write: R/W

Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH27_STREAM_ID0_MASK**

GPCDMA_COMMON_CH28_STREAM_ID0_MASK_0

Offset: 0x1f0
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH28_STREAM_ID0_MASK**

GPCDMA_COMMON_CH29_STREAM_ID0_MASK_0

Offset: 0x1f4
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH29_STREAM_ID0_MASK**

GPCDMA_COMMON_CH30_STREAM_ID0_MASK_0

Offset: 0x1f8
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH30_STREAM_ID0_MASK**

GPCDMA_COMMON_CH31_STREAM_ID0_MASK_0

Offset: 0x1fc
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH31_STREAM_ID0_MASK**

GPCDMA_COMMON_CH0_STREAM_ID1_MASK_0

Offset: 0x200
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH0_STREAM_ID1_MASK**

GPCDMA_COMMON_CH1_STREAM_ID1_MASK_0

Offset: 0x204
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH1_STREAM_ID1_MASK**

GPCDMA_COMMON_CH2_STREAM_ID1_MASK_0

Offset: 0x208
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH2_STREAM_ID1_MASK**

GPCDMA_COMMON_CH3_STREAM_ID1_MASK_0

Offset: 0x20c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH3_STREAM_ID1_MASK**

GPCDMA_COMMON_CH4_STREAM_ID1_MASK_0

Offset: 0x210
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH4_STREAM_ID1_MASK**

GPCDMA_COMMON_CH5_STREAM_ID1_MASK_0

Offset: 0x214
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH5_STREAM_ID1_MASK**

GPCDMA_COMMON_CH6_STREAM_ID1_MASK_0

Offset: 0x218
Read/Write: R/W

Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH6_STREAM_ID1_MASK**

GPCDMA_COMMON_CH7_STREAM_ID1_MASK_0

Offset: 0x21c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH7_STREAM_ID1_MASK**

GPCDMA_COMMON_CH8_STREAM_ID1_MASK_0

Offset: 0x220
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH8_STREAM_ID1_MASK**

GPCDMA_COMMON_CH9_STREAM_ID1_MASK_0

Offset: 0x224
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH9_STREAM_ID1_MASK**

GPCDMA_COMMON_CH10_STREAM_ID1_MASK_0

Offset: 0x228
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH10_STREAM_ID1_MASK**

GPCDMA_COMMON_CH11_STREAM_ID1_MASK_0

Offset: 0x22c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH11_STREAM_ID1_MASK**

GPCDMA_COMMON_CH12_STREAM_ID1_MASK_0

Offset: 0x230
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH12_STREAM_ID1_MASK**

GPCDMA_COMMON_CH13_STREAM_ID1_MASK_0

Offset: 0x234
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH13_STREAM_ID1_MASK**

GPCDMA_COMMON_CH14_STREAM_ID1_MASK_0

Offset: 0x238
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH14_STREAM_ID1_MASK**

GPCDMA_COMMON_CH15_STREAM_ID1_MASK_0

Offset: 0x23c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH15_STREAM_ID1_MASK**

GPCDMA_COMMON_CH16_STREAM_ID1_MASK_0

Offset: 0x240
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH16_STREAM_ID1_MASK**

GPCDMA_COMMON_CH17_STREAM_ID1_MASK_0

Offset: 0x244
Read/Write: R/W

Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH17_STREAM_ID1_MASK**

GPCDMA_COMMON_CH18_STREAM_ID1_MASK_0

Offset: 0x248
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH18_STREAM_ID1_MASK**

GPCDMA_COMMON_CH19_STREAM_ID1_MASK_0

Offset: 0x24c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH19_STREAM_ID1_MASK**

GPCDMA_COMMON_CH20_STREAM_ID1_MASK_0

Offset: 0x250
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH20_STREAM_ID1_MASK**

GPCDMA_COMMON_CH21_STREAM_ID1_MASK_0

Offset: 0x254
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH21_STREAM_ID1_MASK**

GPCDMA_COMMON_CH22_STREAM_ID1_MASK_0

Offset: 0x258
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH22_STREAM_ID1_MASK**

GPCDMA_COMMON_CH23_STREAM_ID1_MASK_0

Offset: 0x25c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH23_STREAM_ID1_MASK**

GPCDMA_COMMON_CH24_STREAM_ID1_MASK_0

Offset: 0x260
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH24_STREAM_ID1_MASK**

GPCDMA_COMMON_CH25_STREAM_ID1_MASK_0

Offset: 0x264
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH25_STREAM_ID1_MASK**

GPCDMA_COMMON_CH26_STREAM_ID1_MASK_0

Offset: 0x268
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH26_STREAM_ID1_MASK**

GPCDMA_COMMON_CH27_STREAM_ID1_MASK_0

Offset: 0x26c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH27_STREAM_ID1_MASK**

GPCDMA_COMMON_CH28_STREAM_ID1_MASK_0

Offset: 0x270
Read/Write: R/W

Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH28_STREAM_ID1_MASK**

GPCDMA_COMMON_CH29_STREAM_ID1_MASK_0

Offset: 0x274
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH29_STREAM_ID1_MASK**

GPCDMA_COMMON_CH30_STREAM_ID1_MASK_0

Offset: 0x278
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH30_STREAM_ID1_MASK**

GPCDMA_COMMON_CH31_STREAM_ID1_MASK_0

Offset: 0x27c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH31_STREAM_ID1_MASK**

GPCDMA_COMMON_CH0_STREAM_ID2_MASK_0

Offset: 0x280
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH0_STREAM_ID2_MASK**

GPCDMA_COMMON_CH1_STREAM_ID2_MASK_0

Offset: 0x284
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH1_STREAM_ID2_MASK**

GPCDMA_COMMON_CH2_STREAM_ID2_MASK_0

Offset: 0x288
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH2_STREAM_ID2_MASK**

GPCDMA_COMMON_CH3_STREAM_ID2_MASK_0

Offset: 0x28c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH3_STREAM_ID2_MASK**

GPCDMA_COMMON_CH4_STREAM_ID2_MASK_0

Offset: 0x290
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH4_STREAM_ID2_MASK**

GPCDMA_COMMON_CH5_STREAM_ID2_MASK_0

Offset: 0x294
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH5_STREAM_ID2_MASK**

GPCDMA_COMMON_CH6_STREAM_ID2_MASK_0

Offset: 0x298
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH6_STREAM_ID2_MASK**

GPCDMA_COMMON_CH7_STREAM_ID2_MASK_0

Offset: 0x29c
Read/Write: R/W

Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH7_STREAM_ID2_MASK**

GPCDMA_COMMON_CH8_STREAM_ID2_MASK_0

Offset: 0x2a0
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH8_STREAM_ID2_MASK**

GPCDMA_COMMON_CH9_STREAM_ID2_MASK_0

Offset: 0x2a4
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH9_STREAM_ID2_MASK**

GPCDMA_COMMON_CH10_STREAM_ID2_MASK_0

Offset: 0x2a8
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH10_STREAM_ID2_MASK**

GPCDMA_COMMON_CH11_STREAM_ID2_MASK_0

Offset: 0x2ac
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH11_STREAM_ID2_MASK**

GPCDMA_COMMON_CH12_STREAM_ID2_MASK_0

Offset: 0x2b0
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH12_STREAM_ID2_MASK**

GPCDMA_COMMON_CH13_STREAM_ID2_MASK_0

Offset: 0x2b4
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH13_STREAM_ID2_MASK**

GPCDMA_COMMON_CH14_STREAM_ID2_MASK_0

Offset: 0x2b8
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH14_STREAM_ID2_MASK**

GPCDMA_COMMON_CH15_STREAM_ID2_MASK_0

Offset: 0x2bc
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH15_STREAM_ID2_MASK**

GPCDMA_COMMON_CH16_STREAM_ID2_MASK_0

Offset: 0x2c0
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH16_STREAM_ID2_MASK**

GPCDMA_COMMON_CH17_STREAM_ID2_MASK_0

Offset: 0x2c4
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH17_STREAM_ID2_MASK**

GPCDMA_COMMON_CH18_STREAM_ID2_MASK_0

Offset: 0x2c8
Read/Write: R/W

Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH18_STREAM_ID2_MASK**

GPCDMA_COMMON_CH19_STREAM_ID2_MASK_0

Offset: 0x2cc
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH19_STREAM_ID2_MASK**

GPCDMA_COMMON_CH20_STREAM_ID2_MASK_0

Offset: 0x2d0
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH20_STREAM_ID2_MASK**

GPCDMA_COMMON_CH21_STREAM_ID2_MASK_0

Offset: 0x2d4
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH21_STREAM_ID2_MASK**

GPCDMA_COMMON_CH22_STREAM_ID2_MASK_0

Offset: 0x2d8
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH22_STREAM_ID2_MASK**

GPCDMA_COMMON_CH23_STREAM_ID2_MASK_0

Offset: 0x2dc
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH23_STREAM_ID2_MASK**

GPCDMA_COMMON_CH24_STREAM_ID2_MASK_0

Offset: 0x2e0
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH24_STREAM_ID2_MASK**

GPCDMA_COMMON_CH25_STREAM_ID2_MASK_0

Offset: 0x2e4
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH25_STREAM_ID2_MASK**

GPCDMA_COMMON_CH26_STREAM_ID2_MASK_0

Offset: 0x2e8
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH26_STREAM_ID2_MASK**

GPCDMA_COMMON_CH27_STREAM_ID2_MASK_0

Offset: 0x2ec
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH27_STREAM_ID2_MASK**

GPCDMA_COMMON_CH28_STREAM_ID2_MASK_0

Offset: 0x2f0
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH28_STREAM_ID2_MASK**

GPCDMA_COMMON_CH29_STREAM_ID2_MASK_0

Offset: 0x2f4
Read/Write: R/W

Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH29_STREAM_ID2_MASK**

GPCDMA_COMMON_CH30_STREAM_ID2_MASK_0

Offset: 0x2f8
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH30_STREAM_ID2_MASK**

GPCDMA_COMMON_CH31_STREAM_ID2_MASK_0

Offset: 0x2fc
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH31_STREAM_ID2_MASK**

GPCDMA_COMMON_CH0_STREAM_ID3_MASK_0

Offset: 0x300
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH0_STREAM_ID3_MASK**

GPCDMA_COMMON_CH1_STREAM_ID3_MASK_0

Offset: 0x304
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH1_STREAM_ID3_MASK**

GPCDMA_COMMON_CH2_STREAM_ID3_MASK_0

Offset: 0x308
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH2_STREAM_ID3_MASK**

GPCDMA_COMMON_CH3_STREAM_ID3_MASK_0

Offset: 0x30c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH3_STREAM_ID3_MASK**

GPCDMA_COMMON_CH4_STREAM_ID3_MASK_0

Offset: 0x310
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH4_STREAM_ID3_MASK**

GPCDMA_COMMON_CH5_STREAM_ID3_MASK_0

Offset: 0x314
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH5_STREAM_ID3_MASK**

GPCDMA_COMMON_CH6_STREAM_ID3_MASK_0

Offset: 0x318
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH6_STREAM_ID3_MASK**

GPCDMA_COMMON_CH7_STREAM_ID3_MASK_0

Offset: 0x31c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH7_STREAM_ID3_MASK**

GPCDMA_COMMON_CH8_STREAM_ID3_MASK_0

Offset: 0x320
Read/Write: R/W

Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH8_STREAM_ID3_MASK**

GPCDMA_COMMON_CH9_STREAM_ID3_MASK_0

Offset: 0x324
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH9_STREAM_ID3_MASK**

GPCDMA_COMMON_CH10_STREAM_ID3_MASK_0

Offset: 0x328
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH10_STREAM_ID3_MASK**

GPCDMA_COMMON_CH11_STREAM_ID3_MASK_0

Offset: 0x32c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH11_STREAM_ID3_MASK**

GPCDMA_COMMON_CH12_STREAM_ID3_MASK_0

Offset: 0x330
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH12_STREAM_ID3_MASK**

GPCDMA_COMMON_CH13_STREAM_ID3_MASK_0

Offset: 0x334
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH13_STREAM_ID3_MASK**

GPCDMA_COMMON_CH14_STREAM_ID3_MASK_0

Offset: 0x338
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH14_STREAM_ID3_MASK**

GPCDMA_COMMON_CH15_STREAM_ID3_MASK_0

Offset: 0x33c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH15_STREAM_ID3_MASK**

GPCDMA_COMMON_CH16_STREAM_ID3_MASK_0

Offset: 0x340
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH16_STREAM_ID3_MASK**

GPCDMA_COMMON_CH17_STREAM_ID3_MASK_0

Offset: 0x344
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH17_STREAM_ID3_MASK**

GPCDMA_COMMON_CH18_STREAM_ID3_MASK_0

Offset: 0x348
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH18_STREAM_ID3_MASK**

GPCDMA_COMMON_CH19_STREAM_ID3_MASK_0

Offset: 0x34c
Read/Write: R/W

Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH19_STREAM_ID3_MASK**

GPCDMA_COMMON_CH20_STREAM_ID3_MASK_0

Offset: 0x350
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH20_STREAM_ID3_MASK**

GPCDMA_COMMON_CH21_STREAM_ID3_MASK_0

Offset: 0x354
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH21_STREAM_ID3_MASK**

GPCDMA_COMMON_CH22_STREAM_ID3_MASK_0

Offset: 0x358
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH22_STREAM_ID3_MASK**

GPCDMA_COMMON_CH23_STREAM_ID3_MASK_0

Offset: 0x35c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH23_STREAM_ID3_MASK**

GPCDMA_COMMON_CH24_STREAM_ID3_MASK_0

Offset: 0x360
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH24_STREAM_ID3_MASK**

GPCDMA_COMMON_CH25_STREAM_ID3_MASK_0

Offset: 0x364
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH25_STREAM_ID3_MASK**

GPCDMA_COMMON_CH26_STREAM_ID3_MASK_0

Offset: 0x368
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH26_STREAM_ID3_MASK**

GPCDMA_COMMON_CH27_STREAM_ID3_MASK_0

Offset: 0x36c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH27_STREAM_ID3_MASK**

GPCDMA_COMMON_CH28_STREAM_ID3_MASK_0

Offset: 0x370
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH28_STREAM_ID3_MASK**

GPCDMA_COMMON_CH29_STREAM_ID3_MASK_0

Offset: 0x374
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH29_STREAM_ID3_MASK**

GPCDMA_COMMON_CH30_STREAM_ID3_MASK_0

Offset: 0x378
Read/Write: R/W

Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH30_STREAM_ID3_MASK**

GPCDMA_COMMON_CH31_STREAM_ID3_MASK_0

Offset: 0x37c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CH31_STREAM_ID3_MASK**

GPCDMA_COMMON_DMA_CHAN_VIRTUALIZATION_ENABLE_0

Offset: 0x380
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_HYPER_0
Reset: 0x0 (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31	0x1	**CH31**
30	0x1	**CH30**
29	0x1	**CH29**
28	0x1	**CH28**
27	0x1	**CH27**
26	0x1	**CH26**
25	0x1	**CH25**
24	0x1	**CH24**
23	0x1	**CH23**
22	0x1	**CH22**
21	0x1	**CH21**

Bit	Reset	Description
20	0x1	**CH20**
19	0x1	**CH19**
18	0x1	**CH18**
17	0x1	**CH17**
16	0x1	**CH16**
15	0x1	**CH15**
14	0x1	**CH14**
13	0x1	**CH13**
12	0x1	**CH12**
11	0x1	**CH11**
10	0x1	**CH10**
9	0x1	**CH9**
8	0x1	**CH8**
7	0x1	**CH7**
6	0x1	**CH6**
5	0x1	**CH5**
4	0x1	**CH4**
3	0x1	**CH3**
2	0x1	**CH2**
1	0x1	**CH1**
0	0x1	**CH0**

GPCDMA_COMMON_DMA_ICG_EN_OVERRIDE_0

Offset: 0x384
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**DMA_ICG_EN_OVERRIDE**

GPCDMA_COMMON_DMA_ACTIVE_0

Offset: 0x388
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	0x0	**DMA_IO_ACTIVE**
2	0x0	**DMA_MC_ACTIVE**
1	0x0	**DMA_IO_MC_ACTIVE**
0	0x0	**DMA_ACTIVE**

GPCDMA_COMMON_SAFETY_LOGIC_CLK_DISABLE_0

Offset: 0x390
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_SAFETY_0
Reset: 0x00000002 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10)

Bit	Reset	Description
1:0	0x2	**SAFETY_LOGIC_CLK_DISABLE**

GPCDMA_ERRCOLLATOR_FEATURE_0

Offset: 0xff00
Read/Write: RO
Parity Protection: N
SCR Protection: EC_SCR_0
Reset: 0x000c0001 (0b0000,0000,0000,1100,xxxx,xxxx,xx00,0001)

Bit	Reset	Description
31:16	0xc	**NUM_ERR**: Number of errors connected to this collator. This is passed as a build time option to the plugin

Bit	Reset	Description
5:0	0x1	**NUM_ERR_SLICES**: Number of error slices supported by this error collator, does not include the GlobalSpace and is derived by CEIL (NUM_ERR/32). Software shall first read this register to determine the number of slices and read the required number of Error_Status registers .

GPCDMA_ERRCOLLATOR_SWRESET_0

Offset: 0xff04
Read/Write: WO
Parity Protection: N
SCR Protection: EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SWRST**: 1'b1 : Issue a Software reset to the Error Collator. This will reset all the registers (Except SCR), counters and logic of the Error Collator. Software can use this bit to flush errors logged into the error collator for example, after Boot, SC7/8 exit. 1'b0 : Do nothing, reset value. This bit is auto-cleared.

GPCDMA_ERRCOLLATOR_MISSIONERR_TYPE_0

Offset: 0xff08
Read/Write: RO
Parity Protection: N
SCR Protection: EC_SCR_0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0101)

Bit	Reset	Description
5:0	0x5	**CODE**: This register indicates the fault code of the error line based on the value of MISSIONERR_INDEX Register. This can be used by a fault handling agent to triage an error without requiring device-specific code. The possible values of this field are: 6'd0 : None 6'd1 : Parity Error on internal data path 6'd2 : ECC SEC Error on internal data path 6'd3 : ECC DED Error on internal data path 6'd4 : Comparator Error 6'd5 : Register Parity Error 6'd6 : Parity Error from on-chip SRAM/FIFO 6'd7 : ECC SEC Error from on-chip SRAM/FIFO 6'd8 : ECC DED Error from on-chip SRAM/FIFO 6'd9 : Clock Monitor Error 6'd10 : Voltage Error 6'd11 : Temperature Error 6'd16 : Software Correctable Error 6'd17 : Software Uncorrectable Error 6'd32 : Other Hardware Correctable Error 6'd33 : Other Hardware Uncorrectable Error All other values : Reserved for future use.

GPCDMA_ERRCOLLATOR_CURRENT_COUNTER_VALUE_0

Offset: 0xff0c
Read/Write: RO
Parity Protection: N
SCR Protection: EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
8:0	0x0	**VALUE**: Provides the current value of the counter corresponding to the error in MissionErr_Index Register. Default provides the value of error 0 counter. Bit[8] is the overflow bit post which the counter saturates and does not counter further.

GPCDMA_ERRCOLLATOR_MISSIONERR_INDEX_0

Offset: 0xff14
Read/Write: R/W
Parity Protection: Y
SCR Protection: EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	0x0	**IDX**: BINARY Encoded. For error number 32, register should be programmed with value 0x20. Write to this register with Error number will update: - MISSIONERR_TYPE Register with the Error-Code for the Error. - CURRENT_COUNTER_VALUE Register with the error's SEC/DED Counter. - MISSIONERR_USERVALUE with value of the first error_<i>_user signal. Software can use this to triage the error. The number shall update the MISSIONERR_TYPE register with the error code and the Current_Counter_Value register with the value of the errors SEC/DED counter. Software can use this register to triage the error.

GPCDMA_ERRCOLLATOR_CORRECTABLE_THRESHOLD_0

Offset: 0xff18
Read/Write: R/W
Parity Protection: Y
SCR Protection: EC_SCR_0
Reset: 0x000000ff (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1111,1111)

Bit	Reset	Description
7:0	0xff	**COUNT**: Threshold value for all SEC Fault Reporting Units connected to this error collator. SEC Errors are logged once the threshold is reached and the overflow bit is set. 7'b0 : Log SEC error after receiving 1 Error. 7'b1 : Log SEC error after receiving 2 Errors. ... 7'bFF : Log SEC error after receiving 256 Errors.

GPCDMA_ERRCOLLATOR_MISSIONERR_INJECT_UNLOCK_0

Offset: 0xff1c
Read/Write: R/W
Parity Protection: N
SCR Protection: EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Writes to ERRSLICE_XXX_MISSIONERR_INJECT registers are disabled until this register is written with a value of 0xE1. This is to prevent an inadvertent safety error injection in the safety plugin due to: 1. A fault on ERRSLICE_XXX_MISSIONERR_INJECT register itself. 2. Erroneous Software. The register shall be written with a value of 0x0 to reestablish the lock after user has completed the error injection testing. 0xE1: Unlock the MISSIONERR_INJECT Register 0x0: Lock the MISSIONERR_INJECT Register **0** = LOCK **225** = UNLOCK

GPCDMA_ERRCOLLATOR_ERRSLICE0_MISSIONERR_ENABLE_0

Offset: 0xff30
Read/Write: R/W
Parity Protection: Y
SCR Protection: EC_SCR_0
Reset: 0x00001fff (0bxxxx,xxxx,xxxx,xxxx,xxx1,1111,1111,1111)

Bit	Reset	Description
12	0x1	**ERR12**: 1'b1 -> Enable Mission Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.clkdis 1'b0 -> Disable Mission Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.clkdis **0** = DISABLE **1** = ENABLE
11	0x1	**ERR11**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from sys0_0.u_NV_gpcdma.u_safety_plugins.u_err_collator_scrwrap 1'b0 -> Disable Mission Error Reporting for Register Parity Error from sys0_0.u_NV_gpcdma.u_safety_plugins.u_err_collator_scrwrap **0** = DISABLE **1** = ENABLE
10	0x1	**ERR10**: 1'b1 -> Enable Mission Error Reporting for ECC DED Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap 1'b0 -> Disable Mission Error Reporting for ECC DED Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
9	0x1	**ERR9**: 1'b1 -> Enable Mission Error Reporting for ECC DED Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap 1'b0 -> Disable Mission Error Reporting for ECC DED Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap **0** = DISABLE **1** = ENABLE
8	0x1	**ERR8**: 1'b1 -> Enable Mission Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap 1'b0 -> Disable Mission Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap **0** = DISABLE **1** = ENABLE
7	0x1	**ERR7**: 1'b1 -> Enable Mission Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap 1'b0 -> Disable Mission Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap **0** = DISABLE **1** = ENABLE
6	0x1	**ERR6**: 1'b1 -> Enable Mission Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_ramintf 1'b0 -> Disable Mission Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_ramintf **0** = DISABLE **1** = ENABLE
5	0x1	**ERR5**: 1'b1 -> Enable Mission Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.activity 1'b0 -> Disable Mission Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.activity **0** = DISABLE **1** = ENABLE
4	0x1	**ERR4**: 1'b1 -> Enable Mission Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.intbus 1'b0 -> Disable Mission Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.intbus **0** = DISABLE **1** = ENABLE
3	0x1	**ERR3**: 1'b1 -> Enable Mission Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap 1'b0 -> Disable Mission Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Mission Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_io_axi 1'b0 -> Disable Mission Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_io_axi **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Mission Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_mc_axi 1'b0 -> Disable Mission Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_mc_axi **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from NV_GPCDMA_err_collator 1'b0 -> Disable Mission Error Reporting for Register Parity Error from NV_GPCDMA_err_collator **0** = DISABLE **1** = ENABLE

GPCDMA_ERRCOLLATOR_ERRSLICE0_MISSIONERR_FORCE_0

Offset: 0xff34
Read/Write: WO
Parity Protection: N
SCR Protection: EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12	0x0	**ERR12**: 1'b1 -> Force Assertion of Mission Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.clkdis 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
11	0x0	**ERR11**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from sys0_0.u_NV_gpcdma.u_safety_plugins.u_err_collator_scrwrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
10	0x0	**ERR10**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC DED Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
9	0x0	**ERR9**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC DED Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
8	0x0	**ERR8**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
7	0x0	**ERR7**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
6	0x0	**ERR6**: 1'b1 -> Force Assertion of Mission Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_ramintf 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
5	0x0	**ERR5**: 1'b1 -> Force Assertion of Mission Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.activity 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
4	0x0	**ERR4**: 1'b1 -> Force Assertion of Mission Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.intbus 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
3	0x0	**ERR3**: 1'b1 -> Force Assertion of Mission Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Mission Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_io_axi 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Mission Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_mc_axi 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from NV_GPCDMA_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

GPCDMA_ERRCOLLATOR_ERRSLICE0_MISSIONERR_STATUS_0

Software must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of MissionError_Enable register, to avoid silent dropping of errors.
Offset: 0xff38
Read/Write: R/W
Parity Protection: Y
SCR Protection: EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12	0x0	**ERR12**: 1'b1 -> Error_12_pulse[1:0] for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.clkdis was equal to 2'b10. 1'b0 -> Error_12_pulse[1:0] for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.clkdis was equal to 2'b01.
11	0x0	**ERR11**: 1'b1 -> Error_11_pulse[1:0] for Register Parity Error from sys0_0.u_NV_gpcdma.u_safety_plugins.u_err_collator_scrwrap was equal to 2'b10. 1'b0 -> Error_11_pulse[1:0] for Register Parity Error from sys0_0.u_NV_gpcdma.u_safety_plugins.u_err_collator_scrwrap was equal to 2'b01.
10	0x0	**ERR10**: 1'b1 -> Error_10_pulse[1:0] for ECC DED Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap was equal to 2'b10. 1'b0 -> Error_10_pulse[1:0] for ECC DED Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap was equal to 2'b01.

Bit	Reset	Description
9	0x0	**ERR9**: 1'b1 -> Error_9_pulse[1:0] for ECC DED Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap was equal to 2'b10. 1'b0 -> Error_9_pulse[1:0] for ECC DED Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap was equal to 2'b01.
8	0x0	**ERR8**: 1'b1 -> Error_8_pulse[1:0] for ECC SEC Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap was equal to 2'b10. 1'b0 -> Error_8_pulse[1:0] for ECC SEC Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap was equal to 2'b01.
7	0x0	**ERR7**: 1'b1 -> Error_7_pulse[1:0] for ECC SEC Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap was equal to 2'b10. 1'b0 -> Error_7_pulse[1:0] for ECC SEC Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap was equal to 2'b01.
6	0x0	**ERR6**: 1'b1 -> Error_6_pulse[1:0] for Comparator Error from sys0_0.u_NV_gpcdma.u_ramintf was equal to 2'b10. 1'b0 -> Error_6_pulse[1:0] for Comparator Error from sys0_0.u_NV_gpcdma.u_ramintf was equal to 2'b01.
5	0x0	**ERR5**: 1'b1 -> Error_5_pulse[1:0] for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.activity was equal to 2'b10. 1'b0 -> Error_5_pulse[1:0] for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.activity was equal to 2'b01.
4	0x0	**ERR4**: 1'b1 -> Error_4_pulse[1:0] for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.intbus was equal to 2'b10. 1'b0 -> Error_4_pulse[1:0] for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.intbus was equal to 2'b01.
3	0x0	**ERR3**: 1'b1 -> Error_3_pulse[1:0] for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap was equal to 2'b10. 1'b0 -> Error_3_pulse[1:0] for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap was equal to 2'b01.
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Comparator Error from sys0_0.u_NV_gpcdma.u_io_axi was equal to 2'b10. 1'b0 -> Error_2_pulse[1:0] for Comparator Error from sys0_0.u_NV_gpcdma.u_io_axi was equal to 2'b01.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Comparator Error from sys0_0.u_NV_gpcdma.u_mc_axi was equal to 2'b10. 1'b0 -> Error_1_pulse[1:0] for Comparator Error from sys0_0.u_NV_gpcdma.u_mc_axi was equal to 2'b01.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from NV_GPCDMA_err_collator was equal to 2'b10. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from NV_GPCDMA_err_collator was equal to 2'b01.

GPCDMA_ERRCOLLATOR_ERRSLICE0_MISSIONERR_INJECT_0

Offset: 0xff3c
Read/Write: R/W
Parity Protection: N
SCR Protection: EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12	0x0	**ERR12**: 1'b1 -> Assert the inject_error_12 output for Comparator Error to sys0_0.u_NV_gpcdma.u_regwrap.clkdis to allow for error injection. 1'b0 -> De-Assert inject_error_12 output. **0** = DISABLE **1** = ENABLE
11	0x0	**ERR11**: 1'b1 -> Assert the inject_error_11 output for Register Parity Error to sys0_0.u_NV_gpcdma.u_safety_plugins.u_err_collator_scrwrap to allow for error injection. 1'b0 -> De-Assert inject_error_11 output. **0** = DISABLE **1** = ENABLE
10	0x0	**ERR10**: 1'b1 -> Assert the inject_error_10 output for ECC DED Error from on-chip SRAM/FIFO to sys0_0.u_NV_gpcdma.u_ramwrap to allow for error injection. 1'b0 -> De-Assert inject_error_10 output. **0** = DISABLE **1** = ENABLE
9	0x0	**ERR9**: 1'b1 -> Assert the inject_error_9 output for ECC DED Error from on-chip SRAM/FIFO to sys0_0.u_NV_gpcdma.u_ramwrap to allow for error injection. 1'b0 -> De-Assert inject_error_9 output. **0** = DISABLE **1** = ENABLE
8	0x0	**ERR8**: 1'b1 -> Assert the inject_error_8 output for ECC SEC Error from on-chip SRAM/FIFO to sys0_0.u_NV_gpcdma.u_ramwrap to allow for error injection. 1'b0 -> De-Assert inject_error_8 output. **0** = DISABLE **1** = ENABLE
7	0x0	**ERR7**: 1'b1 -> Assert the inject_error_7 output for ECC SEC Error from on-chip SRAM/FIFO to sys0_0.u_NV_gpcdma.u_ramwrap to allow for error injection. 1'b0 -> De-Assert inject_error_7 output. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	0x0	**ERR6**: 1'b1 -> Assert the inject_error_6 output for Comparator Error to sys0_0.u_NV_gpcdma.u_ramintf to allow for error injection. 1'b0 -> De-Assert inject_error_6 output. **0** = DISABLE **1** = ENABLE
5	0x0	**ERR5**: 1'b1 -> Assert the inject_error_5 output for Comparator Error to sys0_0.u_NV_gpcdma.u_regwrap.activity to allow for error injection. 1'b0 -> De-Assert inject_error_5 output. **0** = DISABLE **1** = ENABLE
4	0x0	**ERR4**: 1'b1 -> Assert the inject_error_4 output for Comparator Error to sys0_0.u_NV_gpcdma.u_regwrap.intbus to allow for error injection. 1'b0 -> De-Assert inject_error_4 output. **0** = DISABLE **1** = ENABLE
3	0x0	**ERR3**: 1'b1 -> Assert the inject_error_3 output for Comparator Error to sys0_0.u_NV_gpcdma.u_regwrap to allow for error injection. 1'b0 -> De-Assert inject_error_3 output. **0** = DISABLE **1** = ENABLE
2	0x0	**ERR2**: 1'b1 -> Assert the inject_error_2 output for Comparator Error to sys0_0.u_NV_gpcdma.u_io_axi to allow for error injection. 1'b0 -> De-Assert inject_error_2 output. **0** = DISABLE **1** = ENABLE
1	0x0	**ERR1**: 1'b1 -> Assert the inject_error_1 output for Comparator Error to sys0_0.u_NV_gpcdma.u_mc_axi to allow for error injection. 1'b0 -> De-Assert inject_error_1 output. **0** = DISABLE **1** = ENABLE
0	0x0	**ERR0**: 1'b1 -> Assert the inject_error_0 output for Register Parity Error to NV_GPCDMA_err_collator to allow for error injection. 1'b0 -> De-Assert inject_error_0 output. **0** = DISABLE **1** = ENABLE

GPCDMA_ERRCOLLATOR_ERRSLICE0_LATENTERR_ENABLE_0

Offset: 0xff40
Read/Write: R/W

Parity Protection: Y
SCR Protection: EC_SCR_0
Reset: 0x00001fff (0bxxxx,xxxx,xxxx,xxxx,xxx1,1111,1111,1111)

Bit	Reset	Description
12	0x1	**ERR12**: 1'b1 -> Enable Latent Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.clkdis 1'b0 -> Disable Latent Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.clkdis **0** = DISABLE **1** = ENABLE
11	0x1	**ERR11**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from sys0_0.u_NV_gpcdma.u_safety_plugins.u_err_collator_scrwrap 1'b0 -> Disable Latent Error Reporting for Register Parity Error from sys0_0.u_NV_gpcdma.u_safety_plugins.u_err_collator_scrwrap **0** = DISABLE **1** = ENABLE
10	0x1	**ERR10**: 1'b1 -> Enable Latent Error Reporting for ECC DED Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap 1'b0 -> Disable Latent Error Reporting for ECC DED Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap **0** = DISABLE **1** = ENABLE
9	0x1	**ERR9**: 1'b1 -> Enable Latent Error Reporting for ECC DED Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap 1'b0 -> Disable Latent Error Reporting for ECC DED Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap **0** = DISABLE **1** = ENABLE
8	0x1	**ERR8**: 1'b1 -> Enable Latent Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap 1'b0 -> Disable Latent Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap **0** = DISABLE **1** = ENABLE
7	0x1	**ERR7**: 1'b1 -> Enable Latent Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap 1'b0 -> Disable Latent Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	0x1	**ERR6**: 1'b1 -> Enable Latent Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_ramintf 1'b0 -> Disable Latent Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_ramintf **0** = DISABLE **1** = ENABLE
5	0x1	**ERR5**: 1'b1 -> Enable Latent Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.activity 1'b0 -> Disable Latent Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.activity **0** = DISABLE **1** = ENABLE
4	0x1	**ERR4**: 1'b1 -> Enable Latent Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.intbus 1'b0 -> Disable Latent Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.intbus **0** = DISABLE **1** = ENABLE
3	0x1	**ERR3**: 1'b1 -> Enable Latent Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap 1'b0 -> Disable Latent Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap **0** = DISABLE **1** = ENABLE
2	0x1	**ERR2**: 1'b1 -> Enable Latent Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_io_axi 1'b0 -> Disable Latent Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_io_axi **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Latent Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_mc_axi 1'b0 -> Disable Latent Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_mc_axi **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from NV_GPCDMA_err_collator 1'b0 -> Disable Latent Error Reporting for Register Parity Error from NV_GPCDMA_err_collator **0** = DISABLE **1** = ENABLE

GPCDMA_ERRCOLLATOR_ERRSLICE0_LATENTERR_FORCE_0

Offset: 0xff44
Read/Write: WO
Parity Protection: N
SCR Protection: EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12	0x0	**ERR12**: 1'b1 -> Force Assertion of Latent Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.clkdis 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
11	0x0	**ERR11**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from sys0_0.u_NV_gpcdma.u_safety_plugins.u_err_collator_scrwrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
10	0x0	**ERR10**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC DED Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
9	0x0	**ERR9**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC DED Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
8	0x0	**ERR8**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
7	0x0	**ERR7**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
6	0x0	**ERR6**: 1'b1 -> Force Assertion of Latent Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_ramintf 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
5	0x0	**ERR5**: 1'b1 -> Force Assertion of Latent Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.activity 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
4	0x0	**ERR4**: 1'b1 -> Force Assertion of Latent Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.intbus 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
3	0x0	**ERR3**: 1'b1 -> Force Assertion of Latent Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Latent Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_io_axi 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Latent Error Reporting for Comparator Error from sys0_0.u_NV_gpcdma.u_mc_axi 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from NV_GPCDMA_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

GPCDMA_ERRCOLLATOR_ERRSLICE0_LATENTERR_STATUS_0

Software must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of LatentError_Enable register,

to avoid silent dropping of errors.
Offset: 0xff48
Read/Write: R/W
Parity Protection: Y
SCR Protection: EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12	0x0	**ERR12**: 1'b1 -> Error_12_pulse[1:0] for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.clkdis was equal to 2'b00 or 2'b11. 1'b0 -> Error_12_pulse[1:0] for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.clkdis was equal to 2'b01 or 2'b10, but no latent error.
11	0x0	**ERR11**: 1'b1 -> Error_11_pulse[1:0] for Register Parity Error from sys0_0.u_NV_gpcdma.u_safety_plugins.u_err_collator_scrwrap was equal to 2'b00 or 2'b11. 1'b0 -> Error_11_pulse[1:0] for Register Parity Error from sys0_0.u_NV_gpcdma.u_safety_plugins.u_err_collator_scrwrap was equal to 2'b01 or 2'b10, but no latent error.
10	0x0	**ERR10**: 1'b1 -> Error_10_pulse[1:0] for ECC DED Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap was equal to 2'b00 or 2'b11. 1'b0 -> Error_10_pulse[1:0] for ECC DED Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap was equal to 2'b01 or 2'b10, but no latent error.
9	0x0	**ERR9**: 1'b1 -> Error_9_pulse[1:0] for ECC DED Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap was equal to 2'b00 or 2'b11. 1'b0 -> Error_9_pulse[1:0] for ECC DED Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap was equal to 2'b01 or 2'b10, but no latent error.
8	0x0	**ERR8**: 1'b1 -> Error_8_pulse[1:0] for ECC SEC Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap was equal to 2'b00 or 2'b11. 1'b0 -> Error_8_pulse[1:0] for ECC SEC Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap was equal to 2'b01 or 2'b10, but no latent error.
7	0x0	**ERR7**: 1'b1 -> Error_7_pulse[1:0] for ECC SEC Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap was equal to 2'b00 or 2'b11. 1'b0 -> Error_7_pulse[1:0] for ECC SEC Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap was equal to 2'b01 or 2'b10, but no latent error.
6	0x0	**ERR6**: 1'b1 -> Error_6_pulse[1:0] for Comparator Error from sys0_0.u_NV_gpcdma.u_ramintf was equal to 2'b00 or 2'b11. 1'b0 -> Error_6_pulse[1:0] for Comparator Error from sys0_0.u_NV_gpcdma.u_ramintf was equal to 2'b01 or 2'b10, but no latent error.

Bit	Reset	Description
5	0x0	**ERR5**: 1'b1 -> Error_5_pulse[1:0] for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.activity was equal to 2'b00 or 2'b11. 1'b0 -> Error_5_pulse[1:0] for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.activity was equal to 2'b01 or 2'b10, but no latent error.
4	0x0	**ERR4**: 1'b1 -> Error_4_pulse[1:0] for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.intbus was equal to 2'b00 or 2'b11. 1'b0 -> Error_4_pulse[1:0] for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.intbus was equal to 2'b01 or 2'b10, but no latent error.
3	0x0	**ERR3**: 1'b1 -> Error_3_pulse[1:0] for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap was equal to 2'b00 or 2'b11. 1'b0 -> Error_3_pulse[1:0] for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap was equal to 2'b01 or 2'b10, but no latent error.
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Comparator Error from sys0_0.u_NV_gpcdma.u_io_axi was equal to 2'b00 or 2'b11. 1'b0 -> Error_2_pulse[1:0] for Comparator Error from sys0_0.u_NV_gpcdma.u_io_axi was equal to 2'b01 or 2'b10, but no latent error.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Comparator Error from sys0_0.u_NV_gpcdma.u_mc_axi was equal to 2'b00 or 2'b11. 1'b0 -> Error_1_pulse[1:0] for Comparator Error from sys0_0.u_NV_gpcdma.u_mc_axi was equal to 2'b01 or 2'b10, but no latent error.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from NV_GPCDMA_err_collator was equal to 2'b00 or 2'b11. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from NV_GPCDMA_err_collator was equal to 2'b01 or 2'b10, but no latent error.

GPCDMA_ERRCOLLATOR_ERRSLICE0_COUNTER_RELOAD_0

Offset: 0xff50
Read/Write: WO
Parity Protection: N
SCR Protection: EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12	0x0	**ERR12**: 1'b1 -> Reload Error Counter for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.clkdis 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
11	0x0	**ERR11**: 1'b1 -> Reload Error Counter for Register Parity Error from sys0_0.u_NV_gpcdma.u_safety_plugins.u_err_collator_scrwrap 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
10	0x0	**ERR10**: 1'b1 -> Reload Error Counter for ECC DED Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
9	0x0	**ERR9**: 1'b1 -> Reload Error Counter for ECC DED Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
8	0x0	**ERR8**: 1'b1 -> Reload Error Counter for ECC SEC Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
7	0x0	**ERR7**: 1'b1 -> Reload Error Counter for ECC SEC Error from on-chip SRAM/FIFO from sys0_0.u_NV_gpcdma.u_ramwrap 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
6	0x0	**ERR6**: 1'b1 -> Reload Error Counter for Comparator Error from sys0_0.u_NV_gpcdma.u_ramintf 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
5	0x0	**ERR5**: 1'b1 -> Reload Error Counter for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.activity 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

Bit	Reset	Description
4	0x0	**ERR4**: 1'b1 -> Reload Error Counter for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap.intbus 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
3	0x0	**ERR3**: 1'b1 -> Reload Error Counter for Comparator Error from sys0_0.u_NV_gpcdma.u_regwrap 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
2	0x0	**ERR2**: 1'b1 -> Reload Error Counter for Comparator Error from sys0_0.u_NV_gpcdma.u_io_axi 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
1	0x0	**ERR1**: 1'b1 -> Reload Error Counter for Comparator Error from sys0_0.u_NV_gpcdma.u_mc_axi 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
0	0x0	**ERR0**: 1'b1 -> Reload Error Counter for Register Parity Error from NV_GPCDMA_err_collator 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

3.4.4.2 CHANNEL Registers

GPCDMA_CHANNEL_CH0_CSR_0
Offset: 0x10000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH0_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **6** = FIXED_PAT
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH0_STA_0

Offset: 0x10004
Read/Write: See table below
Parity Protection: N

SCR Protection: SCR_CH0_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH0_CSRE_0

Offset: 0x10008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH0_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH0_SRC_PTR_0

Offset: 0x1000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH0_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH0_DST_PTR_0

Offset: 0x10010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH0_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH0_HI_ADR_PTR_0

Offset: 0x10014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH0_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH0_MC_SEQ_0

Offset: 0x10018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH0_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH0_MMIO_SEQ_0

Offset: 0x1001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH0_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH0_WCOUNT_0

Offset: 0x10020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH0_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH0_DMA_WORD_TRA_0

Offset: 0x10024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH0_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH0_DMA_WORD_STA_0

Offset: 0x10028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH0_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH0_ERR_STA_0

Offset: 0x10030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH0_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH0_FIXED_PAT_0

Offset: 0x10034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH0_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH0_TZ_0

Offset: 0x10038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH1_CSR_0

Offset: 0x20000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH1_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH1_STA_0

Offset: 0x20004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH1_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH1_CSRE_0

Offset: 0x20008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH1_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH1_SRC_PTR_0

Offset: 0x2000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH1_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH1_DST_PTR_0

Offset: 0x20010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH1_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH1_HI_ADR_PTR_0

Offset: 0x20014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH1_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH1_MC_SEQ_0

Offset: 0x20018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH1_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH1_MMIO_SEQ_0

Offset: 0x2001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH1_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH1_WCOUNT_0

Offset: 0x20020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH1_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH1_DMA_WORD_TRA_0

Offset: 0x20024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH1_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH1_DMA_WORD_STA_0

Offset: 0x20028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH1_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH1_ERR_STA_0

Offset: 0x20030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH1_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH1_FIXED_PAT_0

Offset: 0x20034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH1_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH1_TZ_0

Offset: 0x20038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH2_CSR_0

Offset: 0x30000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH2_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH2_STA_0

Offset: 0x30004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH2_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH2_CSRE_0

Offset: 0x30008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH2_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH2_SRC_PTR_0

Offset: 0x3000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH2_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH2_DST_PTR_0

Offset: 0x30010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH2_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH2_HI_ADR_PTR_0

Offset: 0x30014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH2_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH2_MC_SEQ_0

Offset: 0x30018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH2_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH2_MMIO_SEQ_0

Offset: 0x3001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH2_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH2_WCOUNT_0

Offset: 0x30020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH2_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH2_DMA_WORD_TRA_0

Offset: 0x30024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH2_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH2_DMA_WORD_STA_0

Offset: 0x30028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH2_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH2_ERR_STA_0

Offset: 0x30030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH2_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH2_FIXED_PAT_0

Offset: 0x30034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH2_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH2_TZ_0

Offset: 0x30038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH3_CSR_0

Offset: 0x40000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH3_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH3_STA_0

Offset: 0x40004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH3_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH3_CSRE_0

Offset: 0x40008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH3_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH3_SRC_PTR_0

Offset: 0x4000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH3_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH3_DST_PTR_0

Offset: 0x40010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH3_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH3_HI_ADR_PTR_0

Offset: 0x40014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH3_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH3_MC_SEQ_0

Offset: 0x40018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH3_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH3_MMIO_SEQ_0

Offset: 0x4001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH3_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH3_WCOUNT_0

Offset: 0x40020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH3_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH3_DMA_WORD_TRA_0

Offset: 0x40024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH3_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH3_DMA_WORD_STA_0

Offset: 0x40028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH3_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH3_ERR_STA_0

Offset: 0x40030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH3_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH3_FIXED_PAT_0

Offset: 0x40034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH3_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH3_TZ_0

Offset: 0x40038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH4_CSR_0

Offset: 0x50000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH4_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH4_STA_0

Offset: 0x50004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH4_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH4_CSRE_0

Offset: 0x50008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH4_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH4_SRC_PTR_0

Offset: 0x5000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH4_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH4_DST_PTR_0

Offset: 0x50010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH4_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH4_HI_ADR_PTR_0

Offset: 0x50014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH4_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH4_MC_SEQ_0

Offset: 0x50018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH4_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH4_MMIO_SEQ_0

Offset: 0x5001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH4_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH4_WCOUNT_0

Offset: 0x50020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH4_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH4_DMA_WORD_TRA_0

Offset: 0x50024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH4_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH4_DMA_WORD_STA_0

Offset: 0x50028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH4_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH4_ERR_STA_0

Offset: 0x50030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH4_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH4_FIXED_PAT_0

Offset: 0x50034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH4_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH4_TZ_0

Offset: 0x50038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH5_CSR_0

Offset: 0x60000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH5_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH5_STA_0

Offset: 0x60004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH5_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH5_CSRE_0

Offset: 0x60008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH5_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH5_SRC_PTR_0

Offset: 0x6000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH5_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH5_DST_PTR_0

Offset: 0x60010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH5_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH5_HI_ADR_PTR_0

Offset: 0x60014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH5_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH5_MC_SEQ_0

Offset: 0x60018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH5_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH5_MMIO_SEQ_0

Offset: 0x6001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH5_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH5_WCOUNT_0

Offset: 0x60020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH5_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH5_DMA_WORD_TRA_0

Offset: 0x60024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH5_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH5_DMA_WORD_STA_0

Offset: 0x60028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH5_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH5_ERR_STA_0

Offset: 0x60030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH5_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH5_FIXED_PAT_0

Offset: 0x60034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH5_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH5_TZ_0

Offset: 0x60038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH6_CSR_0

Offset: 0x70000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH6_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH6_STA_0

Offset: 0x70004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH6_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH6_CSRE_0

Offset: 0x70008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH6_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH6_SRC_PTR_0

Offset: 0x7000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH6_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH6_DST_PTR_0

Offset: 0x70010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH6_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH6_HI_ADR_PTR_0

Offset: 0x70014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH6_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH6_MC_SEQ_0

Offset: 0x70018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH6_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH6_MMIO_SEQ_0

Offset: 0x7001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH6_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH6_WCOUNT_0

Offset: 0x70020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH6_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH6_DMA_WORD_TRA_0

Offset: 0x70024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH6_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH6_DMA_WORD_STA_0

Offset: 0x70028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH6_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH6_ERR_STA_0

Offset: 0x70030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH6_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH6_FIXED_PAT_0

Offset: 0x70034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH6_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH6_TZ_0

Offset: 0x70038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH7_CSR_0

Offset: 0x80000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH7_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH7_STA_0

Offset: 0x80004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH7_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH7_CSRE_0

Offset: 0x80008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH7_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH7_SRC_PTR_0

Offset: 0x8000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH7_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH7_DST_PTR_0

Offset: 0x80010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH7_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH7_HI_ADR_PTR_0

Offset: 0x80014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH7_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH7_MC_SEQ_0

Offset: 0x80018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH7_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH7_MMIO_SEQ_0

Offset: 0x8001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH7_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH7_WCOUNT_0

Offset: 0x80020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH7_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH7_DMA_WORD_TRA_0

Offset: 0x80024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH7_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH7_DMA_WORD_STA_0

Offset: 0x80028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH7_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH7_ERR_STA_0

Offset: 0x80030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH7_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH7_FIXED_PAT_0

Offset: 0x80034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH7_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH7_TZ_0

Offset: 0x80038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH8_CSR_0

Offset: 0x90000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH8_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH8_STA_0

Offset: 0x90004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH8_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH8_CSRE_0

Offset: 0x90008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH8_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH8_SRC_PTR_0

Offset: 0x9000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH8_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH8_DST_PTR_0

Offset: 0x90010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH8_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH8_HI_ADR_PTR_0

Offset: 0x90014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH8_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH8_MC_SEQ_0

Offset: 0x90018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH8_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH8_MMIO_SEQ_0

Offset: 0x9001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH8_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH8_WCOUNT_0

Offset: 0x90020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH8_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH8_DMA_WORD_TRA_0

Offset: 0x90024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH8_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH8_DMA_WORD_STA_0

Offset: 0x90028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH8_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH8_ERR_STA_0

Offset: 0x90030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH8_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH8_FIXED_PAT_0

Offset: 0x90034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH8_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH8_TZ_0

Offset: 0x90038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH9_CSR_0

Offset: 0xa0000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH9_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH9_STA_0

Offset: 0xa0004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH9_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH9_CSRE_0

Offset: 0xa0008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH9_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH9_SRC_PTR_0

Offset: 0xa000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH9_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH9_DST_PTR_0

Offset: 0xa0010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH9_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH9_HI_ADR_PTR_0

Offset: 0xa0014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH9_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH9_MC_SEQ_0

Offset: 0xa0018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH9_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH9_MMIO_SEQ_0

Offset: 0xa001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH9_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH9_WCOUNT_0

Offset: 0xa0020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH9_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH9_DMA_WORD_TRA_0

Offset: 0xa0024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH9_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH9_DMA_WORD_STA_0

Offset: 0xa0028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH9_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH9_ERR_STA_0

Offset: 0xa0030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH9_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH9_FIXED_PAT_0

Offset: 0xa0034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH9_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH9_TZ_0

Offset: 0xa0038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH10_CSR_0

Offset: 0xb0000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH10_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH10_STA_0

Offset: 0xb0004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH10_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH10_CSRE_0

Offset: 0xb0008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH10_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH10_SRC_PTR_0

Offset: 0xb000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH10_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH10_DST_PTR_0

Offset: 0xb0010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH10_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH10_HI_ADR_PTR_0

Offset: 0xb0014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH10_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH10_MC_SEQ_0

Offset: 0xb0018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH10_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH10_MMIO_SEQ_0

Offset: 0xb001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH10_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH10_WCOUNT_0

Offset: 0xb0020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH10_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH10_DMA_WORD_TRA_0

Offset: 0xb0024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH10_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH10_DMA_WORD_STA_0

Offset: 0xb0028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH10_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH10_ERR_STA_0

Offset: 0xb0030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH10_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH10_FIXED_PAT_0

Offset: 0xb0034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH10_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH10_TZ_0

Offset: 0xb0038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH11_CSR_0

Offset: 0xc0000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH11_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH11_STA_0

Offset: 0xc0004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH11_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH11_CSRE_0

Offset: 0xc0008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH11_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH11_SRC_PTR_0

Offset: 0xc000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH11_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH11_DST_PTR_0

Offset: 0xc0010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH11_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH11_HI_ADR_PTR_0

Offset: 0xc0014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH11_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH11_MC_SEQ_0

Offset: 0xc0018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH11_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH11_MMIO_SEQ_0

Offset: 0xc001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH11_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH11_WCOUNT_0

Offset: 0xc0020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH11_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH11_DMA_WORD_TRA_0

Offset: 0xc0024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH11_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH11_DMA_WORD_STA_0

Offset: 0xc0028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH11_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH11_ERR_STA_0

Offset: 0xc0030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH11_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH11_FIXED_PAT_0

Offset: 0xc0034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH11_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH11_TZ_0

Offset: 0xc0038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH12_CSR_0

Offset: 0xd0000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH12_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH12_STA_0

Offset: 0xd0004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH12_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH12_CSRE_0

Offset: 0xd0008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH12_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH12_SRC_PTR_0

Offset: 0xd000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH12_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH12_DST_PTR_0

Offset: 0xd0010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH12_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH12_HI_ADR_PTR_0

Offset: 0xd0014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH12_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH12_MC_SEQ_0

Offset: 0xd0018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH12_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH12_MMIO_SEQ_0

Offset: 0xd001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH12_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH12_WCOUNT_0

Offset: 0xd0020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH12_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH12_DMA_WORD_TRA_0

Offset: 0xd0024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH12_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH12_DMA_WORD_STA_0

Offset: 0xd0028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH12_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH12_ERR_STA_0

Offset: 0xd0030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH12_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH12_FIXED_PAT_0

Offset: 0xd0034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH12_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH12_TZ_0

Offset: 0xd0038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH13_CSR_0

Offset: 0xe0000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH13_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH13_STA_0

Offset: 0xe0004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH13_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH13_CSRE_0

Offset: 0xe0008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH13_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH13_SRC_PTR_0

Offset: 0xe000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH13_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH13_DST_PTR_0

Offset: 0xe0010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH13_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH13_HI_ADR_PTR_0

Offset: 0xe0014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH13_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH13_MC_SEQ_0

Offset: 0xe0018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH13_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH13_MMIO_SEQ_0

Offset: 0xe001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH13_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH13_WCOUNT_0

Offset: 0xe0020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH13_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH13_DMA_WORD_TRA_0

Offset: 0xe0024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH13_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH13_DMA_WORD_STA_0

Offset: 0xe0028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH13_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH13_ERR_STA_0

Offset: 0xe0030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH13_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH13_FIXED_PAT_0

Offset: 0xe0034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH13_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH13_TZ_0

Offset: 0xe0038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH14_CSR_0

Offset: 0xf0000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH14_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH14_STA_0

Offset: 0xf0004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH14_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH14_CSRE_0

Offset: 0xf0008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH14_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH14_SRC_PTR_0

Offset: 0xf000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH14_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH14_DST_PTR_0

Offset: 0xf0010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH14_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH14_HI_ADR_PTR_0

Offset: 0xf0014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH14_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH14_MC_SEQ_0

Offset: 0xf0018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH14_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH14_MMIO_SEQ_0

Offset: 0xf001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH14_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH14_WCOUNT_0

Offset: 0xf0020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH14_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH14_DMA_WORD_TRA_0

Offset: 0xf0024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH14_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH14_DMA_WORD_STA_0

Offset: 0xf0028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH14_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH14_ERR_STA_0

Offset: 0xf0030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH14_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH14_FIXED_PAT_0

Offset: 0xf0034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH14_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH14_TZ_0

Offset: 0xf0038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH15_CSR_0

Offset: 0x100000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH15_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH15_STA_0

Offset: 0x100004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH15_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH15_CSRE_0

Offset: 0x100008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH15_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH15_SRC_PTR_0

Offset: 0x10000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH15_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH15_DST_PTR_0

Offset: 0x100010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH15_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH15_HI_ADR_PTR_0

Offset: 0x100014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH15_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH15_MC_SEQ_0

Offset: 0x100018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH15_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH15_MMIO_SEQ_0

Offset: 0x10001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH15_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH15_WCOUNT_0

Offset: 0x100020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH15_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH15_DMA_WORD_TRA_0

Offset: 0x100024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH15_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH15_DMA_WORD_STA_0

Offset: 0x100028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH15_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH15_ERR_STA_0

Offset: 0x100030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH15_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH15_FIXED_PAT_0

Offset: 0x100034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH15_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH15_TZ_0

Offset: 0x100038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH16_CSR_0

Offset: 0x110000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH16_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH16_STA_0

Offset: 0x110004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH16_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH16_CSRE_0

Offset: 0x110008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH16_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH16_SRC_PTR_0

Offset: 0x11000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH16_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH16_DST_PTR_0

Offset: 0x110010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH16_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH16_HI_ADR_PTR_0

Offset: 0x110014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH16_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH16_MC_SEQ_0

Offset: 0x110018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH16_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH16_MMIO_SEQ_0

Offset: 0x11001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH16_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH16_WCOUNT_0

Offset: 0x110020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH16_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH16_DMA_WORD_TRA_0

Offset: 0x110024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH16_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH16_DMA_WORD_STA_0

Offset: 0x110028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH16_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH16_ERR_STA_0

Offset: 0x110030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH16_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH16_FIXED_PAT_0

Offset: 0x110034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH16_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH16_TZ_0

Offset: 0x110038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH17_CSR_0

Offset: 0x120000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH17_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH17_STA_0

Offset: 0x120004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH17_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH17_CSRE_0

Offset: 0x120008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH17_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH17_SRC_PTR_0

Offset: 0x12000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH17_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH17_DST_PTR_0

Offset: 0x120010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH17_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH17_HI_ADR_PTR_0

Offset: 0x120014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH17_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH17_MC_SEQ_0

Offset: 0x120018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH17_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH17_MMIO_SEQ_0

Offset: 0x12001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH17_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH17_WCOUNT_0

Offset: 0x120020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH17_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH17_DMA_WORD_TRA_0

Offset: 0x120024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH17_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH17_DMA_WORD_STA_0

Offset: 0x120028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH17_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH17_ERR_STA_0

Offset: 0x120030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH17_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH17_FIXED_PAT_0

Offset: 0x120034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH17_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH17_TZ_0

Offset: 0x120038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH18_CSR_0

Offset: 0x130000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH18_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH18_STA_0

Offset: 0x130004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH18_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH18_CSRE_0

Offset: 0x130008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH18_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH18_SRC_PTR_0

Offset: 0x13000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH18_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH18_DST_PTR_0

Offset: 0x130010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH18_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH18_HI_ADR_PTR_0

Offset: 0x130014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH18_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH18_MC_SEQ_0

Offset: 0x130018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH18_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH18_MMIO_SEQ_0

Offset: 0x13001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH18_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH18_WCOUNT_0

Offset: 0x130020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH18_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH18_DMA_WORD_TRA_0

Offset: 0x130024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH18_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH18_DMA_WORD_STA_0

Offset: 0x130028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH18_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH18_ERR_STA_0

Offset: 0x130030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH18_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH18_FIXED_PAT_0

Offset: 0x130034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH18_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH18_TZ_0

Offset: 0x130038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH19_CSR_0

Offset: 0x140000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH19_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH19_STA_0

Offset: 0x140004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH19_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH19_CSRE_0

Offset: 0x140008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH19_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH19_SRC_PTR_0

Offset: 0x14000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH19_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH19_DST_PTR_0

Offset: 0x140010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH19_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH19_HI_ADR_PTR_0

Offset: 0x140014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH19_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH19_MC_SEQ_0

Offset: 0x140018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH19_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH19_MMIO_SEQ_0

Offset: 0x14001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH19_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH19_WCOUNT_0

Offset: 0x140020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH19_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH19_DMA_WORD_TRA_0

Offset: 0x140024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH19_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH19_DMA_WORD_STA_0

Offset: 0x140028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH19_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH19_ERR_STA_0

Offset: 0x140030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH19_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH19_FIXED_PAT_0

Offset: 0x140034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH19_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH19_TZ_0

Offset: 0x140038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH20_CSR_0

Offset: 0x150000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH20_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH20_STA_0

Offset: 0x150004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH20_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH20_CSRE_0

Offset: 0x150008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH20_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH20_SRC_PTR_0

Offset: 0x15000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH20_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH20_DST_PTR_0

Offset: 0x150010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH20_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH20_HI_ADR_PTR_0

Offset: 0x150014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH20_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH20_MC_SEQ_0

Offset: 0x150018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH20_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH20_MMIO_SEQ_0

Offset: 0x15001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH20_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH20_WCOUNT_0

Offset: 0x150020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH20_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH20_DMA_WORD_TRA_0

Offset: 0x150024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH20_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH20_DMA_WORD_STA_0

Offset: 0x150028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH20_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH20_ERR_STA_0

Offset: 0x150030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH20_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH20_FIXED_PAT_0

Offset: 0x150034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH20_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH20_TZ_0

Offset: 0x150038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH21_CSR_0

Offset: 0x160000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH21_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH21_STA_0

Offset: 0x160004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH21_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH21_CSRE_0

Offset: 0x160008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH21_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH21_SRC_PTR_0

Offset: 0x16000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH21_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH21_DST_PTR_0

Offset: 0x160010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH21_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH21_HI_ADR_PTR_0

Offset: 0x160014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH21_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH21_MC_SEQ_0

Offset: 0x160018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH21_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH21_MMIO_SEQ_0

Offset: 0x16001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH21_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH21_WCOUNT_0

Offset: 0x160020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH21_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH21_DMA_WORD_TRA_0

Offset: 0x160024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH21_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH21_DMA_WORD_STA_0

Offset: 0x160028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH21_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH21_ERR_STA_0

Offset: 0x160030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH21_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH21_FIXED_PAT_0

Offset: 0x160034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH21_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH21_TZ_0

Offset: 0x160038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH22_CSR_0

Offset: 0x170000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH22_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH22_STA_0

Offset: 0x170004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH22_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH22_CSRE_0

Offset: 0x170008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH22_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH22_SRC_PTR_0

Offset: 0x17000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH22_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH22_DST_PTR_0

Offset: 0x170010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH22_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH22_HI_ADR_PTR_0

Offset: 0x170014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH22_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH22_MC_SEQ_0

Offset: 0x170018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH22_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH22_MMIO_SEQ_0

Offset: 0x17001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH22_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH22_WCOUNT_0

Offset: 0x170020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH22_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH22_DMA_WORD_TRA_0

Offset: 0x170024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH22_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH22_DMA_WORD_STA_0

Offset: 0x170028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH22_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH22_ERR_STA_0

Offset: 0x170030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH22_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH22_FIXED_PAT_0

Offset: 0x170034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH22_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH22_TZ_0

Offset: 0x170038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH23_CSR_0

Offset: 0x180000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH23_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH23_STA_0

Offset: 0x180004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH23_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH23_CSRE_0

Offset: 0x180008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH23_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH23_SRC_PTR_0

Offset: 0x18000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH23_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH23_DST_PTR_0

Offset: 0x180010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH23_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH23_HI_ADR_PTR_0

Offset: 0x180014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH23_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH23_MC_SEQ_0

Offset: 0x180018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH23_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH23_MMIO_SEQ_0

Offset: 0x18001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH23_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH23_WCOUNT_0

Offset: 0x180020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH23_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH23_DMA_WORD_TRA_0

Offset: 0x180024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH23_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH23_DMA_WORD_STA_0

Offset: 0x180028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH23_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH23_ERR_STA_0

Offset: 0x180030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH23_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH23_FIXED_PAT_0

Offset: 0x180034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH23_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH23_TZ_0

Offset: 0x180038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH24_CSR_0

Offset: 0x190000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH24_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH24_STA_0

Offset: 0x190004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH24_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH24_CSRE_0

Offset: 0x190008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH24_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH24_SRC_PTR_0

Offset: 0x19000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH24_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH24_DST_PTR_0

Offset: 0x190010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH24_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH24_HI_ADR_PTR_0

Offset: 0x190014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH24_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH24_MC_SEQ_0

Offset: 0x190018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH24_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH24_MMIO_SEQ_0

Offset: 0x19001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH24_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH24_WCOUNT_0

Offset: 0x190020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH24_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH24_DMA_WORD_TRA_0

Offset: 0x190024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH24_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH24_DMA_WORD_STA_0

Offset: 0x190028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH24_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH24_ERR_STA_0

Offset: 0x190030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH24_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH24_FIXED_PAT_0

Offset: 0x190034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH24_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH24_TZ_0

Offset: 0x190038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH25_CSR_0

Offset: 0x1a0000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH25_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH25_STA_0

Offset: 0x1a0004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH25_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH25_CSRE_0

Offset: 0x1a0008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH25_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH25_SRC_PTR_0

Offset: 0x1a000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH25_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH25_DST_PTR_0

Offset: 0x1a0010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH25_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH25_HI_ADR_PTR_0

Offset: 0x1a0014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH25_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH25_MC_SEQ_0

Offset: 0x1a0018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH25_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH25_MMIO_SEQ_0

Offset: 0x1a001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH25_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH25_WCOUNT_0

Offset: 0x1a0020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH25_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH25_DMA_WORD_TRA_0

Offset: 0x1a0024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH25_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH25_DMA_WORD_STA_0

Offset: 0x1a0028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH25_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH25_ERR_STA_0

Offset: 0x1a0030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH25_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH25_FIXED_PAT_0

Offset: 0x1a0034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH25_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH25_TZ_0

Offset: 0x1a0038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH26_CSR_0

Offset: 0x1b0000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH26_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH26_STA_0

Offset: 0x1b0004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH26_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH26_CSRE_0

Offset: 0x1b0008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH26_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH26_SRC_PTR_0

Offset: 0x1b000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH26_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH26_DST_PTR_0

Offset: 0x1b0010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH26_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH26_HI_ADR_PTR_0

Offset: 0x1b0014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH26_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH26_MC_SEQ_0

Offset: 0x1b0018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH26_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH26_MMIO_SEQ_0

Offset: 0x1b001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH26_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH26_WCOUNT_0

Offset: 0x1b0020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH26_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH26_DMA_WORD_TRA_0

Offset: 0x1b0024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH26_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH26_DMA_WORD_STA_0

Offset: 0x1b0028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH26_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH26_ERR_STA_0

Offset: 0x1b0030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH26_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH26_FIXED_PAT_0

Offset: 0x1b0034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH26_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH26_TZ_0

Offset: 0x1b0038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH27_CSR_0

Offset: 0x1c0000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH27_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH27_STA_0

Offset: 0x1c0004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH27_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH27_CSRE_0

Offset: 0x1c0008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH27_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH27_SRC_PTR_0

Offset: 0x1c000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH27_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH27_DST_PTR_0

Offset: 0x1c0010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH27_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH27_HI_ADR_PTR_0

Offset: 0x1c0014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH27_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH27_MC_SEQ_0

Offset: 0x1c0018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH27_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH27_MMIO_SEQ_0

Offset: 0x1c001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH27_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH27_WCOUNT_0

Offset: 0x1c0020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH27_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH27_DMA_WORD_TRA_0

Offset: 0x1c0024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH27_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH27_DMA_WORD_STA_0

Offset: 0x1c0028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH27_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH27_ERR_STA_0

Offset: 0x1c0030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH27_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH27_FIXED_PAT_0

Offset: 0x1c0034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH27_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH27_TZ_0

Offset: 0x1c0038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH28_CSR_0

Offset: 0x1d0000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH28_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH28_STA_0

Offset: 0x1d0004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH28_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH28_CSRE_0

Offset: 0x1d0008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH28_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH28_SRC_PTR_0

Offset: 0x1d000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH28_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH28_DST_PTR_0

Offset: 0x1d0010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH28_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH28_HI_ADR_PTR_0

Offset: 0x1d0014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH28_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH28_MC_SEQ_0

Offset: 0x1d0018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH28_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH28_MMIO_SEQ_0

Offset: 0x1d001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH28_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH28_WCOUNT_0

Offset: 0x1d0020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH28_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH28_DMA_WORD_TRA_0

Offset: 0x1d0024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH28_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH28_DMA_WORD_STA_0

Offset: 0x1d0028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH28_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH28_ERR_STA_0

Offset: 0x1d0030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH28_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH28_FIXED_PAT_0

Offset: 0x1d0034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH28_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH28_TZ_0

Offset: 0x1d0038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH29_CSR_0

Offset: 0x1e0000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH29_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH29_STA_0

Offset: 0x1e0004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH29_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH29_CSRE_0

Offset: 0x1e0008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH29_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH29_SRC_PTR_0

Offset: 0x1e000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH29_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH29_DST_PTR_0

Offset: 0x1e0010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH29_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH29_HI_ADR_PTR_0

Offset: 0x1e0014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH29_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH29_MC_SEQ_0

Offset: 0x1e0018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH29_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH29_MMIO_SEQ_0

Offset: 0x1e001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH29_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH29_WCOUNT_0

Offset: 0x1e0020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH29_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH29_DMA_WORD_TRA_0

Offset: 0x1e0024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH29_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH29_DMA_WORD_STA_0

Offset: 0x1e0028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH29_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH29_ERR_STA_0

Offset: 0x1e0030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH29_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH29_FIXED_PAT_0

Offset: 0x1e0034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH29_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH29_TZ_0

Offset: 0x1e0038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH30_CSR_0

Offset: 0x1f0000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH30_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH30_STA_0

Offset: 0x1f0004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH30_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH30_CSRE_0

Offset: 0x1f0008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH30_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH30_SRC_PTR_0

Offset: 0x1f000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH30_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH30_DST_PTR_0

Offset: 0x1f0010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH30_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH30_HI_ADR_PTR_0

Offset: 0x1f0014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH30_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH30_MC_SEQ_0

Offset: 0x1f0018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH30_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH30_MMIO_SEQ_0

Offset: 0x1f001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH30_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH30_WCOUNT_0

Offset: 0x1f0020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH30_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH30_DMA_WORD_TRA_0

Offset: 0x1f0024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH30_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH30_DMA_WORD_STA_0

Offset: 0x1f0028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH30_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH30_ERR_STA_0

Offset: 0x1f0030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH30_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH30_FIXED_PAT_0

Offset: 0x1f0034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH30_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH30_TZ_0

Offset: 0x1f0038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

GPCDMA_CHANNEL_CH31_CSR_0

Offset: 0x200000
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH31_0
Reset: 0x08008400 (0b00xx,1x00,0000,0000,1x00,01xx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ENB**: **0** = DISABLE **1** = ENABLE
30	0x0	**IE_EOC**: **0** = DISABLE **1** = ENABLE
27	0x1	**ONCE**: **0** = CYCLIC_MODE **1** = SINGLE_BLOCK
25:24	0x0	**FC_MODE**: **0** = NO_MMIO **1** = ONE_MMIO **2** = TWO_MMIO **3** = FOUR_MMIO
23:21	0x0	**DMA_MODE**: **0** = IO2MEM_NO_FC **1** = IO2MEM_FC **2** = MEM2IO_NO_FC **3** = MEM2IO_FC **4** = MEM2MEM **5** = RSVD **6** = FIXED_PAT

Bit	Reset	Description
20:16	0x0	**REQ_SEL**: **0** = I2C8 **1** = I2C10 **2** = UARTG **3** = UARTC **5** = QSPI0 **6** = QSPI1 **8** = UARTA **9** = UARTB **12** = UARTF **13** = UARTH **15** = SPI1 **16** = SPI2 **17** = SPI3 **19** = UARTD **20** = UARTE **21** = I2C **22** = I2C2 **23** = I2C3 **24** = I2C5 **26** = I2C4 **27** = I2C7 **30** = I2C6 **31** = I2C9 **29** = RSVD
15	0x1	**IRQ_MASK**: **0** = DISABLE **1** = ENABLE
13:10	0x1	**WEIGHT**

GPCDMA_CHANNEL_CH31_STA_0

Offset: 0x200004
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH31_0
Reset: 0x00000000 (0b00x0,0000,0x00,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RO	0x0	**BSY**: **0** = WAIT **1** = ACTIVE
30	RW	0x0	**ISE_EOC**: **0** = NO_INTR **1** = INTR
28	RO	0x0	**PING_PONG_STA**: **0** = PING_INTR_STA **1** = PONG_INTR_STA
27	RO	0x0	**DMA_ACTIVITY**: **0** = IDLE **1** = BUSY

Bit	R/W	Reset	Description
26	RO	0x0	**CHANNEL_PAUSE**: **0** = RESUME **1** = PAUSE
25	RO	0x0	**CHANNEL_RX**: **0** = NOT_ACTIVE **1** = ACTIVE
24	RO	0x0	**CHANNEL_TX**: **0** = NOT_ACTIVE **1** = ACTIVE
23	RO	0x0	**IRQ_INTR_STA**: **0** = DISABLE **1** = ENABLE
21	RO	0x0	**TRIG_STA**: **0** = NOT_ACTIVE **1** = ACTIVE
20	RO	0x0	**INTR_STA**: **0** = NOT_ACTIVE **1** = ACTIVE

GPCDMA_CHANNEL_CH31_CSRE_0

Offset: 0x200008
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH31_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,0000,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_ACTIVITY**: **0** = RESUME **1** = PAUSE

Bit	Reset	Description
19:14	0x0	**TRIG_SEL**: **1** = SMP_24 **2** = SMP_25 **3** = SMP_26 **4** = SMP_27 **5** = XRQ_A **6** = XRQ_B **7** = TMR1 **8** = TMR2 **9** = CH0 **10** = CH1 **11** = CH2 **12** = CH3 **13** = CH4 **14** = CH5 **15** = CH6 **16** = CH7 **17** = CH8 **18** = CH9 **19** = CH10 **20** = CH11 **21** = CH12 **22** = CH13 **23** = CH14 **24** = CH15 **25** = CH16 **26** = CH17 **27** = CH18 **28** = CH19 **29** = CH20 **30** = CH21 **31** = CH22 **32** = CH23 **33** = CH24 **34** = CH25 **35** = CH26 **36** = CH27 **37** = CH28 **38** = CH29 **39** = CH30 **40** = CH31 **0** = RSVD

GPCDMA_CHANNEL_CH31_SRC_PTR_0

Offset: 0x20000c
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH31_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_PTR**

GPCDMA_CHANNEL_CH31_DST_PTR_0

Offset: 0x200010
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH31_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_PTR**

GPCDMA_CHANNEL_CH31_HI_ADR_PTR_0

Offset: 0x200014
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH31_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x0	**HI_DST_PTR**
7:0	0x0	**HI_SRC_PTR**

GPCDMA_CHANNEL_CH31_MC_SEQ_0

Offset: 0x200018
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH31_0
Reset: 0x03800000 (0b0000,0011,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MC_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
30:25	0x1	**MC_REQ_CNT**
24:23	0x3	**MC_BURST**: **0** = DMA_BURST_2WORDS **3** = DMA_BURST_16WORDS

Bit	Reset	Description
22:20	0x0	**MC_ADDR_WRAP1**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
19:17	0x0	**MC_ADDR_WRAP0**: **0** = NO_WRAP **1** = WRAP_0N_32WORDS **2** = WRAP_ON_64WORDS **3** = WRAP_ON_128WORDS **4** = WRAP_ON_256WORDS **5** = WRAP_ON_512WORDS **6** = WRAP_ON_1024WORDS **7** = WRAP_ON_2048WORDS
16	0x0	**MC_AXIID**
15:14	0x0	**MC_PROT**
13:7	0x0	**STREAMID1**
6:0	0x0	**STREAMID0**

GPCDMA_CHANNEL_CH31_MMIO_SEQ_0

Offset: 0x20001c
Read/Write: See table below
Parity Protection: N
SCR Protection: SCR_CH31_0
Reset: 0x2381007f (0b0010,0011,1000,0001,xxxx,xxx0,0111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**DBL_BUF**: **0** = DISABLE **1** = ENABLE
30:28	RW	0x2	**MMIO_BUS_WIDTH**: **0** = BUS_WIDTH_8 **1** = BUS_WIDTH_16 **2** = BUS_WIDTH_32
27	RW	0x0	**MMIO_DATA_SWAP**: **0** = DISABLE **1** = ENABLE
26:23	RW	0x7	**MMIO_BURST**: **0** = DMA_BURST_1WORDS **1** = DMA_BURST_2WORDS **3** = DMA_BURST_4WORDS **7** = DMA_BURST_8WORDS **15** = DMA_BURST_16WORDS

Bit	R/W	Reset	Description
22:19	RO	0x0	**MMIO_MASTER_ID**: **0** = RSVD **1** = CCPLEX **2** = CCPLEX_DPMU **3** = BPMP **4** = SPE **5** = SCE **6** = DMA_PER **7** = TSECA **8** = TSECB **9** = JTAGM **10** = CSITE **11** = APE
18:16	RW	0x1	**MMIO_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_0N_1WORDS **2** = WRAP_ON_2WORDS **3** = WRAP_ON_4WORDS **4** = WRAP_ON_8WORDS **5** = WRAP_ON_16WORDS **6** = WRAP_ON_32WORDS **7** = WRAP_ON_64WORDS
8:7	RW	0x0	**MMIO_PROT**
6:0	RO	0x7f	**MMIO_CHANNEL_SECURITY**

GPCDMA_CHANNEL_CH31_WCOUNT_0

Offset: 0x200020
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH31_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WCOUNT**

GPCDMA_CHANNEL_CH31_DMA_WORD_TRA_0

Offset: 0x200024
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH31_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TRANSFER_COUNT**

GPCDMA_CHANNEL_CH31_DMA_WORD_STA_0

Offset: 0x200028
Read/Write: RO
Parity Protection: N
SCR Protection: SCR_CH31_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_COUNT**

GPCDMA_CHANNEL_CH31_ERR_STA_0

Offset: 0x200030
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH31_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERROR_STATUS**

GPCDMA_CHANNEL_CH31_FIXED_PAT_0

Offset: 0x200034
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_CH31_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIXED_PATTERN**

GPCDMA_CHANNEL_CH31_TZ_0

Offset: 0x200038
Read/Write: R/W
Parity Protection: N
SCR Protection: SCR_TZ_0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**MC_PROT_1**
0	0x1	**MMIO_PROT_1**

3.5 System Memory Management Unit (SMMU)

3.5.1 Overview

The System Memory Management Unit (SMMU) is part of the memory system in the SoC. The purpose of the SMMU is to allow the use of virtual addressing for memory accesses by the hardware devices. The SMMU provides address translation, supports a two-stage look-up for hypervisor support, and can apply various controls and protections for memory access. This function is sometimes also referred to as an IOMMU.

The SMMU is based on the Arm® MMU-500. This chapter should be read with the Arm documentation, in particular the *Arm CoreLink™ MMU-500 System Memory Management Unit Technical Reference Manual. T*hat document details the implementation defined features, and the *Arm System Memory Management Unit Architecture Specification,* which is necessary in understanding the programming model. Refer also to the Memory Subsystem (MSS) chapter of this TRM, where many SMMU related concepts are discussed.

For the latest version of the Arm documents, refer to the Arm website: http://infocenter.arm.com/help/index.jsp

- *Arm CoreLink™ MMU-500 System Memory Management Unit Technical Reference Manual:* Document ID Arm DDI 0517E (ID072715)
- *Arm System Memory Management Unit Architecture Specification:* Document ID Arm IHI 0062D.b (ID071415)

3.5.1.1 SMMU Configuration

The SoC has three instances of SMMU. One of them serves the real-time requests from display and VI, and is referred to as the ISO MMU (ISO is an abbreviation for isochronous). The other two serve the rest of the SoC client traffic, and are referred to as NISO MMUs. All the SMMU instances are configured to support the following:

- 64 contexts
- Stage 1 and Stage 2 translation
- 7-bit streamID

- 128 stream matching groups
- Eight TBUs in each NISO SMMU and four TBUs in ISO SMMU

The configuration parameters above may have an effect on registers defined in the *Arm System Memory Management Unit Architecture Specification*.

Revision of the Arm IP used:

- TBU: r2p2
- TCU: r2p4

For the memory subsystem (MSS) datapath to operate correctly, the following must be programmed:

- AArch32 short descriptors should not be used.
- The SoC supports only 7-bit StreamIDs, so SMR.MASK[14:7] must be programmed to 'hFF.
- Program SCTLR.CFRE / SCTLR.CFIE / CR0.GFRE / CR0.GFIE / CR0.GCFGFRE / CR0.GCFGFIE = 1.
- The stall-fault model is not supported. Program SCTLR.CFCFG = 0.
- Prefetch must be disabled by setting SMMU_CBn_ACTLR.CPRE = 0.

3.5.1.2 StreamID

The StreamID is used to select a context in SMMU that is used for translation. For information on how to map a StreamID to a context, refer to the *Arm System Memory Management Unit Architecture Specification*.

The StreamID can be set by a client on a per-transaction basis. However, this behavior can be overridden by programming the StreamID override registers. The override is applied on a per-client basis for all transactions issued by that client. Refer to the Memory Controller (MC) chapter for client details.

The StreamID override registers are part of a separate aperture aligned to 64 KB so that it can be controlled by the hypervisor in a virtualized system.

4. Boot and Power Management

4.1 Boot and Power Management Processor (BPMP)

4.1.1 Overview

The Boot and Power Management Processor (BPMP) complex provides a set of hardware functions that support the following tasks:

- Boot
  - Cold boot
  - Warm boot
  - Deep-sleep (SC7) entry and exit
- Power Management
  - DVFS and clock/voltage management
  - SoC power state management
  - Core rail power management (i.e., VDD_CPU, VDD_CORE, VDD_CV, VDD_SOC)
  - Process, Voltage, and Temperature Sensor management

The BPMP complex includes:

- Dual Arm Cortex-R5F cores running in delayed lock-step with 32 KiB of I-cache and 32 KiB of D-cache, both with ECC
- 128 KiB of tightly-coupled memory (TCM) with ECC
- Boot ROM
- Timers, a DMA controller, an interrupt controller, and a set of peripherals for controlling the functions listed above

Note: The BPMP processor runs NVIDIA supplied software and is controlled through a NVIDIA API. Details of the hardware programming interface are not supplied in this document.

5. CPU Complex (CCPLEX)

5.1 Overview

The CPU Complex in the NVIDIA® Orin™ series System-on-Chip (SoC) consists of up to three CPU Clusters, each containing four Arm® Cortex®-A78AE cores, sometimes referred to by the code-name Hercules-AE.

5.1.1 Reference Documentation

The majority of the documentation for this CPU is supplied by Arm, and the following Arm documents should be referred to.

All these documents are publicly available on the Arm Developer web site, and the links below are valid at the time of publishing this section, but might potentially change, in which case a search on the document name should locate it.

5.1.1.1 List of References

The CCPLEX chapter makes the implicit use of the following documents available from Arm, and assumes readers are familiar with the Arm Architecture, and have access to the documents for reference. Refer to the Arm website to download these documents. See the *Additional reading* section under *About this book* in each of the following for additional Arm publications containing other relevant information.

- *Arm® Cortex®-A78AE Core Technical Reference Manual*
  https://developer.arm.com/documentation/101779/latest
- *Arm® Cortex®-A78AE Core Cryptographic Extension Technical Reference Manual*
  https://developer.arm.com/documentation/101799/latest
- *Arm® DynamIQ$_{TM}$ Shared Unit-AE Technical Reference Manual*
  https://developer.arm.com/documentation/101322/latest
- *Arm® CoreLink™ GIC-600AE Generic Interrupt Controller Technical Reference Manual*
  https://developer.arm.com/documentation/101206/latest
- *Arm® Architecture Reference Manual Armv8, for Armv8-A architecture profile*
  https://developer.arm.com/documentation/ddi0487/latest/

Refer also to the most recent version of the errata notices.

- *Arm Cortex-A78AE (MP105) Software Developer Errata Notice*
  https://developer.arm.com/documentation/SDEN1707912/latest
- *Arm DSU-AE (MP092)Software Developer Errata Notice*
  https://developer.arm.com/documentation/SDEN1343188/latest
- *CoreLink GIC-600 Generic Interrupt Controller Software Developer Errata Notice*
  https://developer.arm.com/documentation/sden892601/latest

The floating-point unit conforms to the following two specifications, available from the IEEE:

- ANSI/IEEE, *IEEE Standard for Binary Floating-Point Arithmetic*, Std 754-1985.
- ANSI/IEEE, *IEEE Standard for Floating-Point Arithmetic*, Std 754-2008.

5.1.2 Glossary

This glossary defines a number of terms used in the context of the CPU Complex. See also the glossary in the Introduction section of this document for terms used across the SoC.

Term	Definition
ACDI	Arm Cluster Debug Interface. A design unit in SCF CMU that instantiates the Debug Block IP from Arm and other necessary Coresight debug related collateral to interface with the SoC.
ACE	AXI Coherency Extensions
ACI	Arm Cluster Interface. Arm Cluster's gateway to the external world is through the ACI design unit.
ADB	AMBA Domain Bridge
AE	Automotive Enhanced (as in Cortex®-A78AE, Hercules-AE, DSU-AE.)
AMU	Activity Monitoring Unit
ARI	Abstract Request Interface. An interface that enables Arm software to communicate with the MCE software to enable modification of CREGs that are part of the CCPLEX.
Arm Cluster	Cluster may sometimes be referred to as the Arm Cluster to explicitly indicating that the Cluster is made up of CPU cores from Arm (as opposed to NVIDIA CPUs).
AVFS	Automatic Voltage Frequency Scaling
CCPMU	CCPLEX Power Management Unit
CMU	CCPLEX Miscellaneous Unit
CMULA	CCPLEX Miscellaneous Unit Logic Analyzer
CPE	Copy Engine
CRAB	Control Register Access Bus

Term	Definition
CREG	Control Register
CTI	Cross Trigger Interface
DSU-AE	DynamIQ™ Shared Unit. Arm IP containing Cluster coherency logic and the Level-3 (L3) Cache.
DSU-AE Cache	The L3 Cache inside the DSU-AE is also referred to as the DSU-AE Cache. (L3 and DSU-AE Cache are synonymous and are used interchangeably in this document.)
DUT	Design Under Test. (Referring to the design unit that a test bench is attempting to validate.)
DVM	Distributed Virtual Memory. DVM transactions support the maintenance of a virtual memory system
ETM	Embedded Trace Macrocell
FIQ	Fast Interrupt Request
FMU	Fault Management Unit
IH	Interrupt Handling unit
IOB	Input-Output Bridge
IRI	Interrupt Routing Infrastructure
IRQ	Interrupt Request
ISM	In Silicon Measurement. (ISM circuits are sprinkled around the SoC to measure various parameters for silicon characterization.)
IST	In System Test. (IST is a mechanism akin to running ATPG vectors on a chip tester except that it is performed on chip by hardware similar to Memory Built In Self Test or MBIST.)
Lock mode	Mode of operation where a pair of CPU's in a CPU Cluster operate as one CPU and is recognized by the operating system also as one CPU.
MCA	Machine Check Architecture
MP2LS	2-core Multi-Processor in Lock Step mode. (An MP4 processor configuration becomes a 2-core configuration in Lock-Step Mode.)
MP4	4-core Multi-Processor. (Arm acronym denoting a multi-processor CPU configuration with four CPU cores.)
NAFLL	Noise Aware Frequency Locked Loop. (Noise aware clock generator that is part of the Automatic Voltage Frequency Scaling, or AVFS, sub-system.)
PMU	Power Management Unit. (CCPMU in NVIDIA nomenclature, and PMU is Arm's nomenclature, while NVIDIA refers to the Performance Monitoring Unit as PerfMon.)
POD	Power On Detector. (A circuit that monitors a voltage rail for glitches and causes a processor reset in case of one.)

Term	Definition
PPI	Private Peripheral Interrupt
RAS	Reliability, Availability, and Serviceability
SCF	System Coherency Fabric
SCF Cache	System Coherency Fabric has a Cache that is technically Level 4.
SCU	Snoop Control Unit
SKU	Stock Keeping Unit. (A generic term used to inventory and track products) (An SKU is also used to indicate a group of chips that have the same feature set or characteristics.)
SNOC	System Network-On-Chip
SPI	Shared Peripheral Interrupt
Split Mode	Mode of operation where each CPU in a CPU Cluster operates independently and is recognized by the operating system as a different CPU.
vGIC	Virtual Generic Interrupt Controller (Interrupt Controller for v8 Arm architecture.)

5.1.3 Conventions of Units

This chapter follows the IEEE and NIST conventions for multiplying prefixes. Among other things, this convention uses an 'i' to indicate the binary convention, and its absence to indicate decimal. So, 1 KiB is 2^{10} or 1,024 bytes, and 1 KB is 10^3 or 1,000 bytes. Similarly it uses:

- Mi for 2^{20} and M for 10^6
- Gi for 2^{30} and G for 10^9
- Ti for 2^{40} and T for 10^{12}

5.2 CCPLEX Functional Description

5.2.1 Architecture Overview

The Orin CPU Complex (CCPLEX) uses the Cortex-A78AE CPU core from ARM, that is also referred to as Hercules-AE. The Cortex-A78AE is an ARM Architecture v8-A CPU that has an integrated 256-KiB Level-2 (L2) Cache in each CPU.

Each set of four Cortex-A78AE cores is grouped into a quad-core Cluster. The four CPUs in each Cluster are connected to an ARM DynamIQ™ Shared Unit (DSU-AE) to share the 2-MiB Level-3 (L3)

Cache, also referred to as the DSU-AE Cache. The Orin CCPLEX has up to three of these quad-CPU Clusters.

An optional safety Lock-Step Mode is supported by the Cortex-A78AE to allow two Cortex-A78AE cores in a Cluster to execute the same code in tandem and their results compared at each clock cycle. Any divergence (mismatched result) is detected and reported as a Lock-Step Error. This mode can be disabled to allow the two cores to operate independently in Full Performance Mode.

5.2.1.1 Key Features

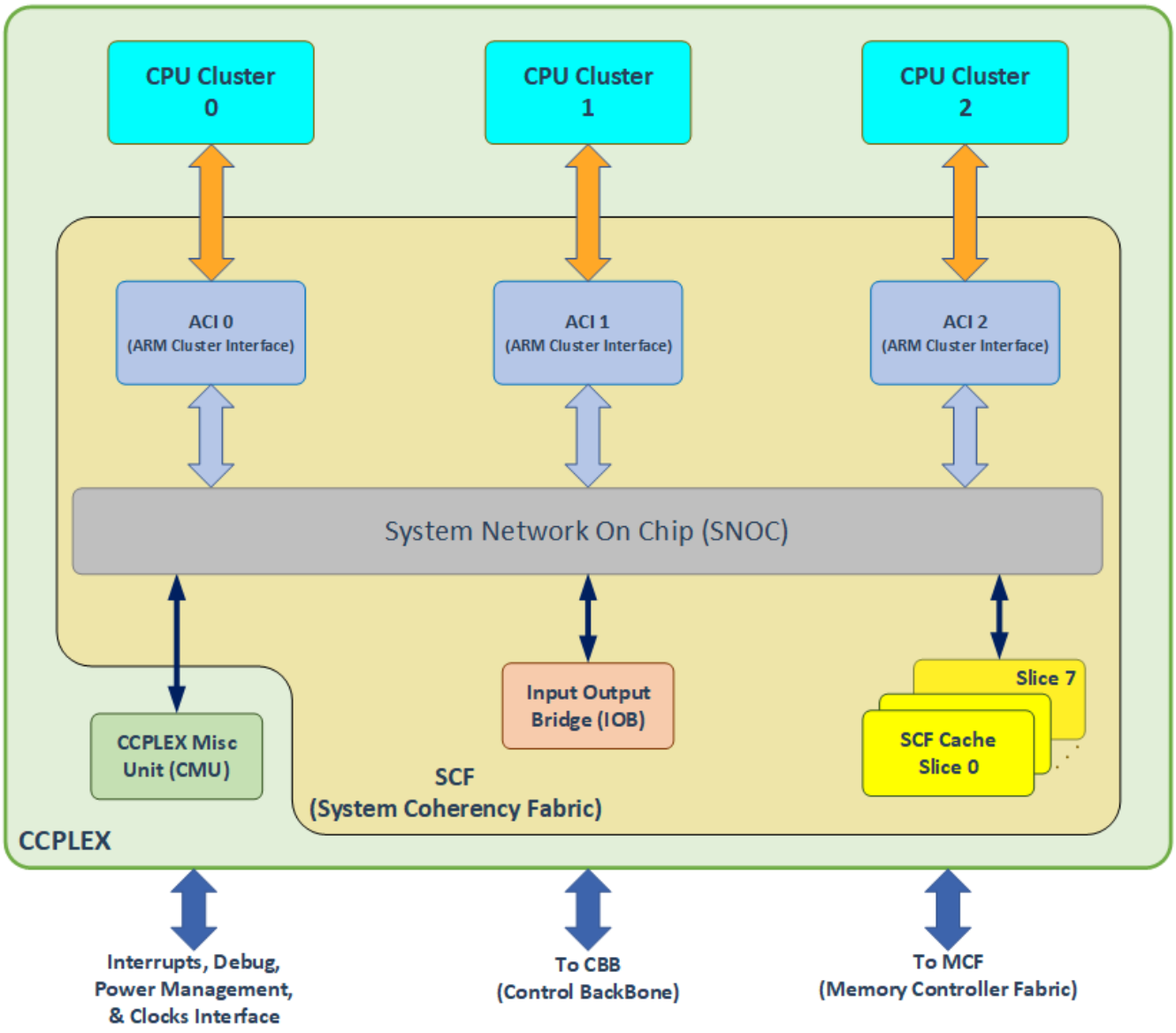
- Up to three CPU Clusters.
- All CPU Clusters share the same 4-MiB 16-way Set-Associative SCF (L4) Cache that is partitioned in eight 512-KiB slices.
- Split-Lock-Hybrid Safety Architecture for the Cortex-A78AE processors.
- Memory-mapped RAS architecture.
- Integration of Clusters with Virtual Generic Interrupt Controller (vGIC) based on ARM GIC-600AE.
- Fault Management Unit inside vGIC.
- AARCH32 Execution Exception Level: EL0 only.
- AARCH64 Execution Exception Levels: EL0 through EL3.

5.2.1.2 Architecture Summary

5.2.1.2.1 CPU Complex (CCPLEX)

The high-level overview of the CPU Complex showing its CPU Clusters is presented in the diagram below.

Figure 5.1 CPU Complex and Its CPU Clusters



Each CPU Cluster interfaces with an ARM Cluster Interface (ACI) module that is part of the System Coherency Fabric (SCF) and serves as the gateway for signals going in and out of the Cluster. The CPU memory Reads and Writes are routed to SCF Cache Slices then the Memory Controller Fabric (MCF) via ACI and SNOC, while Reads and Writes to the Memory-Mapped Input Output (MMIO) space are routed to the Input Output Bridge (IOB) then Control BackBone (CBB), or back into the units within CCPLEX. The 4-MiB SCF Cache is divided in eight slices, each of 512 KiB.

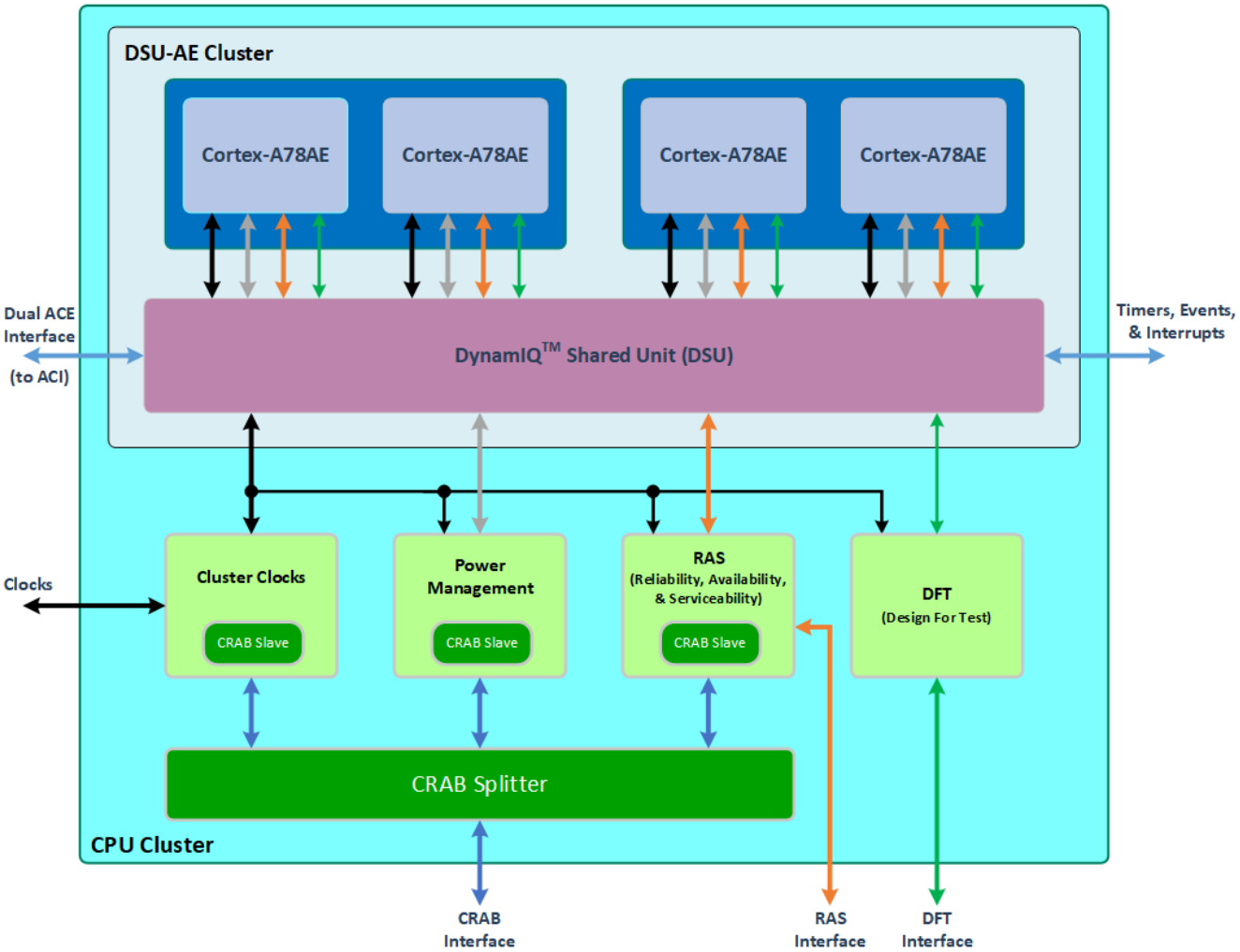
The CCPLEX Miscellaneous Unit (CMU) contains has the following units to handle power management, Interrupts, and Debug.

- The CPU Complex Power Management Unit (CCPMU) with micro-code programmable engines to execute power management sequences.
- The Interrupt Handler (IH) with vGIC (based on ARM GIC-600AE).
- The ARM Cluster Debug Interface (ACDI) module with the ARM Debug Block and several other components necessary to support the Coresight Debug architecture.

5.2.1.3 CPU Clusters

The high-level overview of the CPU Cluster showing its CPU (Cortex-A78AE) cores is presented in the diagram below.

Figure 5.2 CPU Cluster and Its CPU Cores



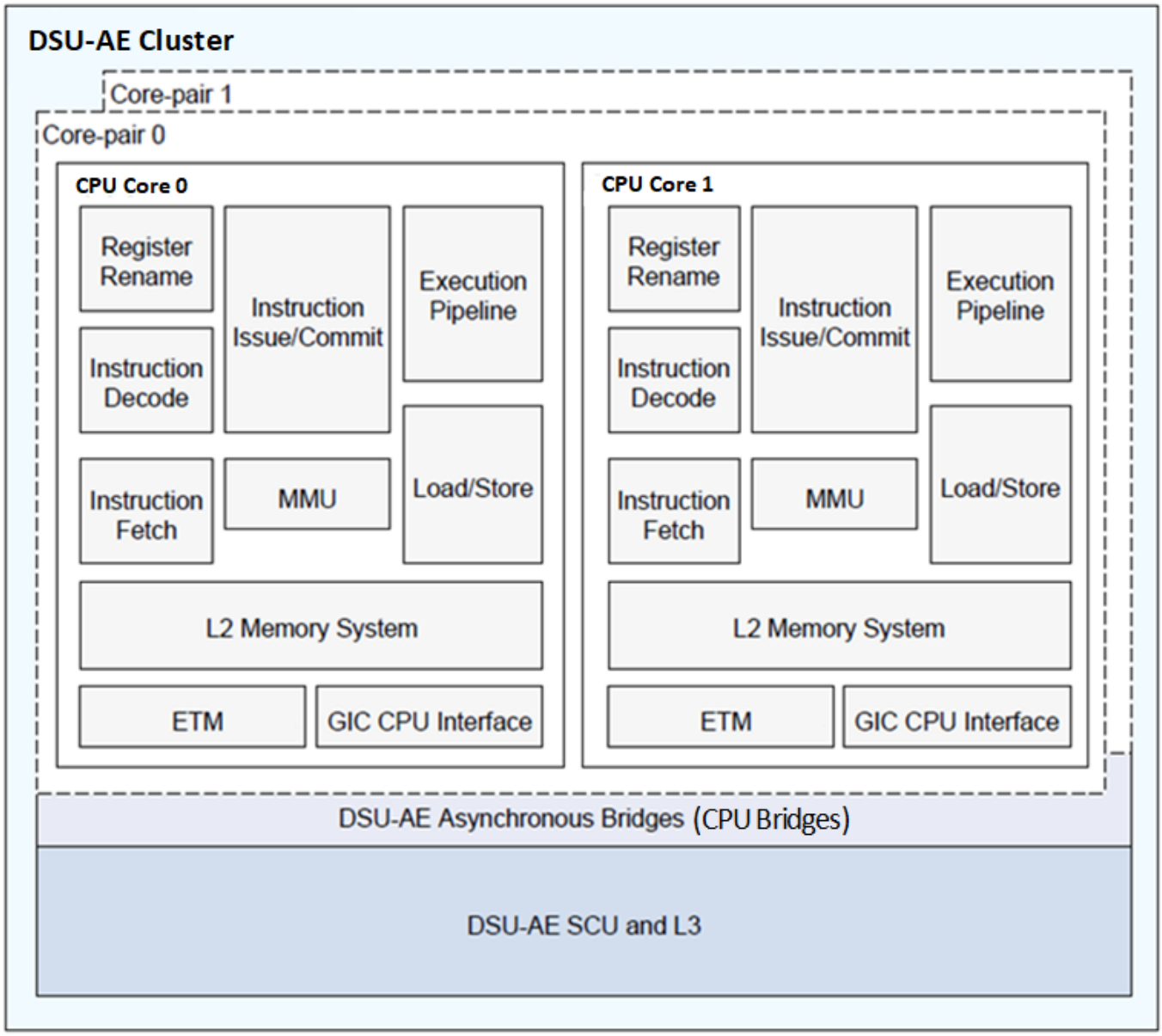
The CPU Cluster features are summarized as follows:

- Four Cortex-A78AE cores in each CPU Cluster.
- The DynamIQ™ Shared Unit (DSU-AE) with a 2-MiB L3 Cache is responsible for the coherency amongst the cores in the Cluster.
- Split Mode of operation with independent CPU in the Cluster where the number of physical CPUs equals to the number of CPU visible to software.
- Lock Mode of operation with pairs of CPU operate in Lock-Step as one CPU seen by software.
- Reliability, Availability, and Serviceability (RAS) logic to Interface with the DSU-AE and capture RAS records in the MMIO space.
- One common CPU power rail for all CPU Clusters in the CPU Complex.
- Logic for debug, trace, and performance monitoring logic.
- Noise Aware Frequency Locked Loop (NAFLL) clock source for CPU cores.
- Noise Aware Frequency Locked Loop (NAFLL) clock source for the DSU-AE.
- Support for Adaptive Voltage and Frequency Scaling.
- Independent power gating domains for each of the CPUs and the DSU-AE.
- Power management support logic to interact with the CCPMU.
- Interrupt-related support logic to interact with the vGIC.
- Control Register Access Bus (CRAB) slaves and support fabric.
- Design For Test (DFT) logic.

5.2.1.3.1 CPU Cores and DSU-AE

The architectural overview of the DSU-AE Cluster showing its one DSU-AE and four CPU cores with major functional blocks is presented in the diagram below.

Figure 5.3 DSU-AE Cluster of one DSU-AE Core and four CPU Cores with Architectural Details



CPU Core

The CPU (Cortex-A78AE) core implements the ARMv8-A architecture. Refer to the Arm Cortex-A78AE Core Technical Reference Manual for specifics of the ARM architecture extensions supported.

Each CPU core has an L1 memory Cache system and an integrated L2 Cache to work in concert with its superscalar, variable-length, out-of-order pipeline. The CPU core features are summarized as follows:

- Four Cortex-A78AE cores and one DSU-AE in each CPU Cluster.
  - Each Cortex-A78AE core has 64 KiB Instruction and Data Cache each, plus an integrated 256-KiB L2 Cache.
  - Four Cortex-A78AE cores in the Cluster share the 2-MiB L3 Cache inside the DSU-AE.
- Exception Levels.
  - AArch32 execution states at EL0 only.
  - AArch64 execution states at all exception levels, i.e. EL0 through EL3.
- 1.5 Ki 98-bit entries, 4-way Skewed-Associative L0 Macro-OP (MOP) Cache.
- Separate 64-KiB 4-way Set-Associative L1 Data and Instruction Cache.
  - ECC and parity protection on L1 Cache memory.
- Private unified 256-KiB 8-way Set-Associative Data and Instruction L2 Cache.
  - ECC and parity protection on L2 Cache memory.
  - L2 transaction queue size of 48 entries (24 entries per L2 bank).
- A Memory Management Unit (MMU).
- An integrated execution unit that implements the advanced SIMD and floating-point architecture support.
- Generic Interrupt Controller (GICv4) CPU interface to connect to an external distributor.
- Generic Timers interface supporting 64-bit count input from an external system counter.
- Split/Lock feature with the ability to choose amongst Split, Lock, and Hybrid mode at the Cluster level based on CEMODE input during Cluster's cold Reset.
- Performance Monitoring Unit (PMU).
- Activity Monitoring Unit (AMU).
- Embedded Trace Macrocell (ETM) supporting instruction trace only.

The following table lists the execution modes specifics supported by the Orin CCPLEX.

Table 5.1 Execution Modes

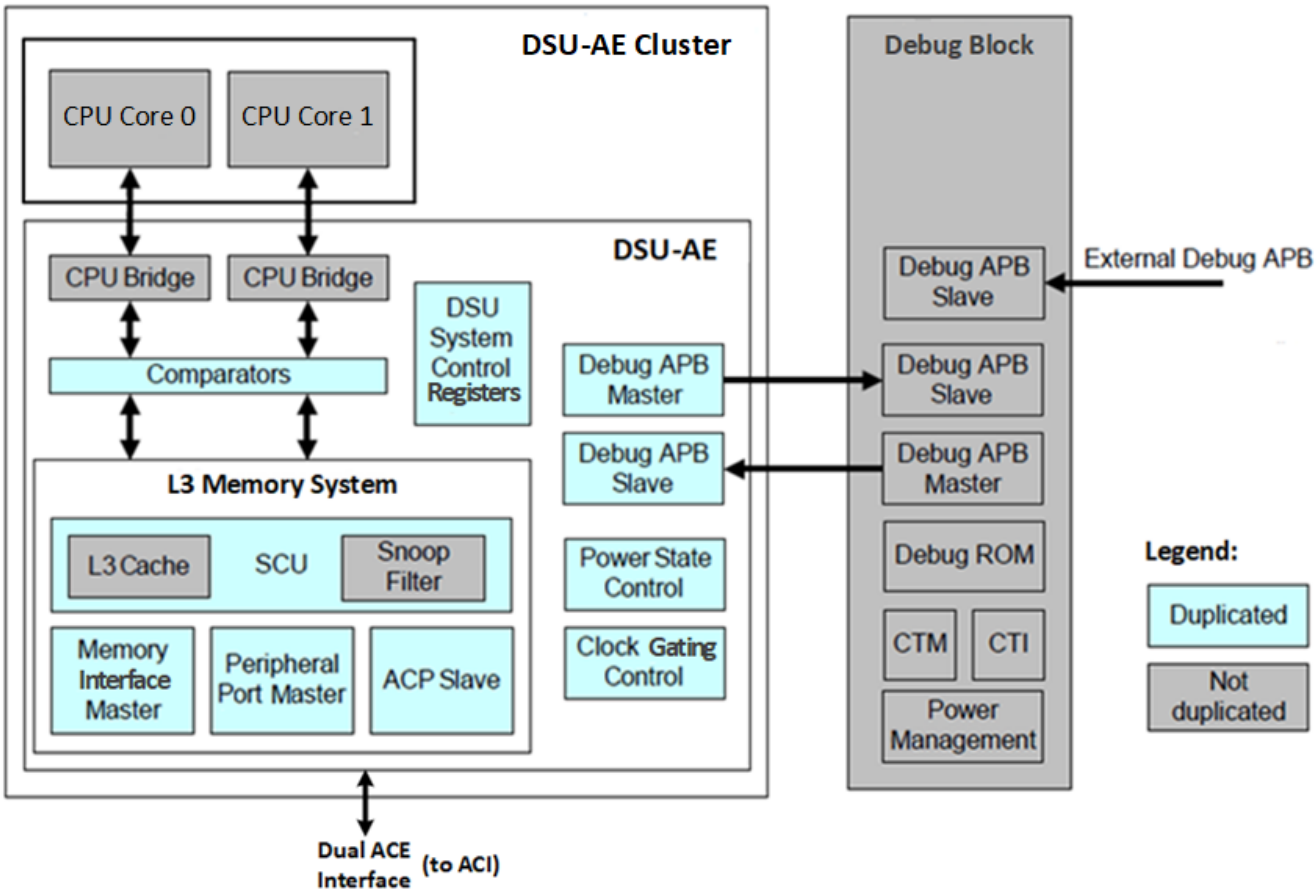
Cluster Mode	CPU Mode	DSU Mode	Description
Split	Split	Split	Each CPU core executes independently as appeared to the operating system, with four CPU cores in a Cluster.
Lock	Lock	Lock	Two CPU cores execute the same software concurrently. Their execution state is compared every clock cycle and any divergence (mismatched result) is reported. These two physical CPU cores appear as one logical CPU core to the operating system for a total of four CPU cores in a Cluster.

Cluster Mode	CPU Mode	DSU Mode	Description
Hybrid	Split	Lock	Cluster Hybrid mode means that CPUs operate in Split mode while the DSU operates in Lock mode. DSU logic is duplicated as a primary and redundant logic. Each CPU's outputs are fed into the primary and redundant DSU logic. Primary and redundant DSU logic are compared in Lock mode and mismatched results generate RAS errors. This Cluster Hybrid mode is useful for Safety applications where CPU cores can be in Split mode (thereby improving performance with more CPUs available) while the DSU is in Lock mode. Permanent fault coverage for the CPU cores is done using online IST (In-System Test) for coverage metrics.

DSU-AE

The architectural overview of the DSU-AE Cluster showing its one CPU core-pair and the DSU-AE with major functional blocks is presented in the diagram below.

Figure 5.4 DSU-AE Cluster Showing one CPU Core-pair and the DSU Core with Architectural Details



The DynamIQ™ Shared Unit (DSU-AE) comprises a Snoop Control Unit (SCU) with an L3 Memory System, DSU-AE Asynchronous Bridges (CPU Bridges), and other control logic as shown in the diagram above. The DSU-AE features are summarized as follows:

- Dual ACE (AXI Coherency Extensions) Interfaces (AMBA5 ACE main bus interfaces) to ACI.
- 48-bit Physical Address (PA) support, where only 40 bits are used in Orin, i.e. PA[47:40] = 8'b0 always.
- Reliability, Availability and Serviceability (RAS) support.
- Data poisoning on a 64-bit granule.
- 2-MiB 16-way Set-Associative L3 Cache.
  - 64-byte Cache line throughput.
  - L3 Cache partitioning - 2 slices.
- ECC support on Cache RAMs.

- SCU maintains coherency between Caches in the cores and the DSU Cache.
- Split, Lock, and Hybrid mode support.
- Interface Protection Safety feature through Parity for the external interfaces of the DSU-AE Cluster.

6. GPU

6.1 Overview

The Orin™ SoC has an NVIDIA Ampere GPU with two Graphics Processing Clusters (GPCs) responsible for all the compute/graphics processing including:

- Graphics-related computation
- Rasterization
- Rendering
- Ray Tracing
- Pixel generation

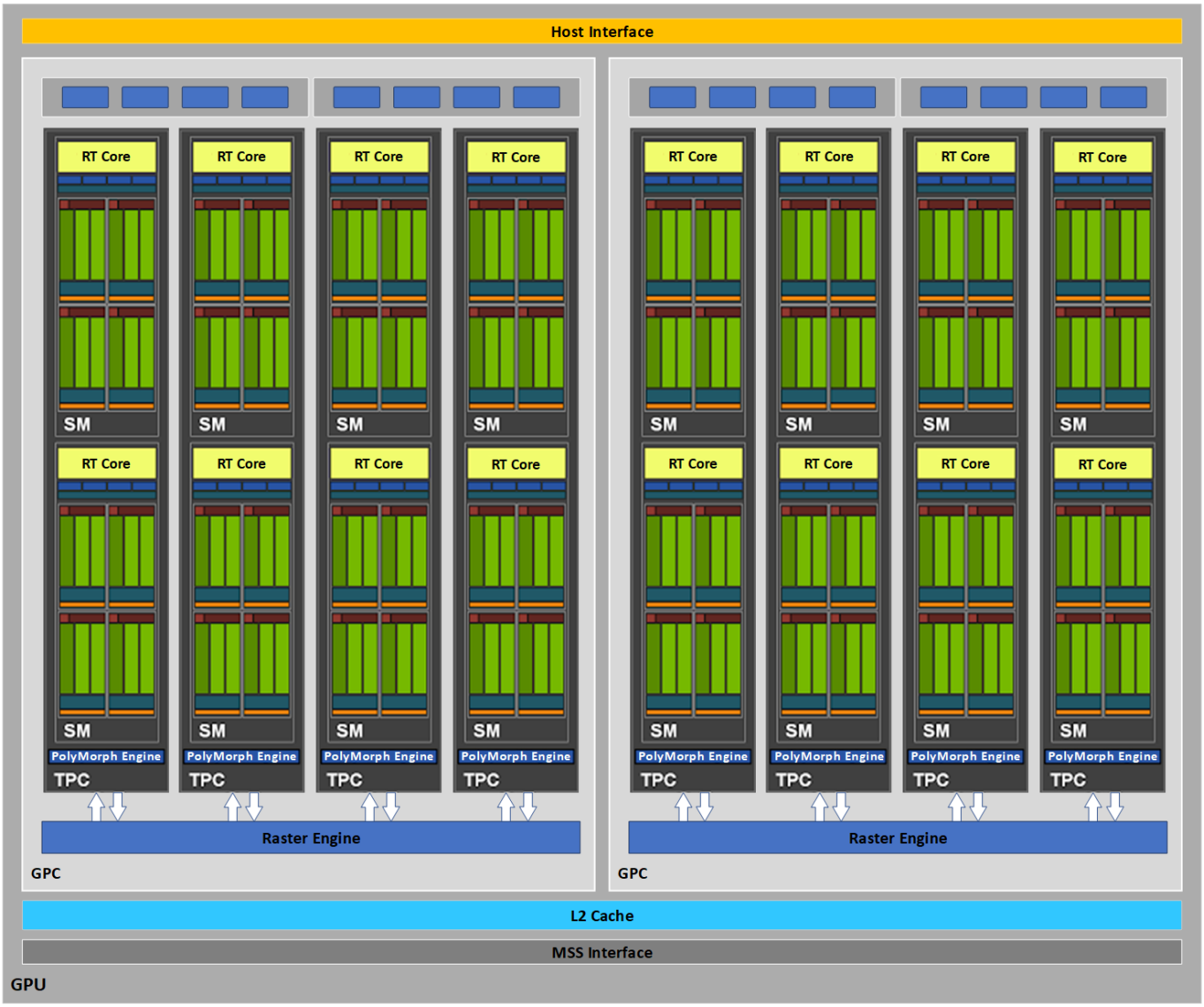
To achieve the above, the Ampere GPC features:

- 4x Texture Processing Clusters (TPCs), each consisting of:
  - 2x Streaming Multiprocessors (SMs), each with its own Ray Tracing (RT) core
  - 1x PolyMorph Engine (PE)
- 1x Raster Engine

To interact with the SoC for tasks to do and completed, the Ampere GPU features:

- Level-2 (L2) Cache
- MSS Interface
- Host Interface

Figure 6.1 Ampere GPU in Orin



The key functional blocks and features in the Orin's Ampere GPU are summarized in the following sub-sections.

6.1.1 Streaming Multiprocessor (SM)

The Ampere Streaming Multiprocessor (SM) has 128 CUDA cores. As shown in the following diagram, the SM is partitioned in four processing blocks, with each containing:

- 3rd-generation Tensor core
- 64-KiB Register Files (in 16,384 x 32 organization)

- Texture (TEX) unit
- L0 I-Cache
- Warp Scheduler
- Dispatch (32 threads/clk) unit

All four SM share:

- 192-KiB for L1 Data Cache / Shared Memory
- a 2nd-generation Ray Tracing (RT) core

Figure 6.2 Ampere Streaming Multiprocessor (SM)

6.1.1.1 Third-Generation Tensor Core

The third-generation Tensor Core supports various data types for improved performance, efficiency, and programming flexibility. These data types are:

- INT1
- INT4 (2's complement and magnitude)
- INT8 (2's complement and magnitude)
- INT16 (2's complement and magnitude)
- INT32 (2's complement and magnitude)
- FP16 (standard IEEE Half-precision FP format, with 1 Sign bit, 5 Exponent bits, and 10 Mantissa bits)
- FP32 (standard IEEE Single-precision FP format, with 1 Sign bit, 8 Exponent bits, and 23 Mantissa bits)
- FP64 (standard IEEE Double-precision FP format, with 1 Sign bit, 11 Exponent bits, and 52 Mantissa bits)
- BF16 (an alternative to FP16, with 1 Sign bit, 8 Exponent bits, and 7 Mantissa bits)
- TF32 (1 Sign bit, 8 Exponent bits, and 10 Mantissa bits)

Its new Sparsity feature can take advantage of fine-grained structured sparsity in deep learning networks to double the throughput of its operations compared to its predecessor.

In addition, the new Tensor Float 32 (TF32) precision provides up to 5x throughput compared to the previous generation to accelerate AI and data science model training without any code changes. It also significantly accelerates the following operations:

- AI denoising
- NVIDIA DLSS for AI super resolution (now with support for up to 8K)
- NVIDIA Broadcast apps for AI-enhanced video and voice communications
- NVIDIA Canvas app for AI-powered painting

6.1.1.2 2x FP32 Processing

Most graphics workloads are composed of 32-bit Floating point (FP32) operations. The Streaming Multiprocessor (SM) in the Ampere GPU Architecture is designed to support double-speed processing of FP32 operations by adopting the FP32 processing capability on both of the two datapaths within an SM partition, thus doubling the peak processing rate for FP32 operations. In particular, one datapath in the partition consists of the equivalence of 16 FP32 CUDA Cores capable of executing 16 FP32 operations per clock, while the other consists of 16 INT32 Cores and the equivalence of 16 FP32 CUDA Cores capable of executing either 16 INT32 operations or 16 FP32 operations per clock. Consequently, each SM partition can execute either 32 FP32 operations per clock, or 16 FP32 and 16 INT32 operations per clock.

Typical graphics workloads have a mixture of FP32 arithmetic instructions such as:

- Full single-precision Floating-point Multiplication and Addition (FFMA)

- Floating-point Addition (FADD)
- Floating-point Multiplication (FMUL)
- Floating-point Compare or Min/Max for processing results
- Many simpler Integer instructions such as Additions for addressing and fetching data, etc.

Having the floating-point capability in both datapaths significantly improves the workloads processing.

The Ampere SM continues to support double-speed FP16 Half single-precision Floating-point Multiplication and Addition (HFMA) operations. Standard FP16 operations are handled by the Tensor Cores.

The 2x FP32 processing provides significant performance improvements for graphics workflows such as 3D model development and workloads compute acceleration for complex 3D simulation critical in Computer-Aided Design (CAD) and Computer-Aided Engineering (CAE) applications.

Also worth noting is that the SM features 2 FP64 units to ensure programs with FP64 codes (including FP64 Tensor Core code) operate correctly.

6.1.1.3 Second-Generation Ray Tracing (RT) Core

The second-generation Ray Tracing (RT) Core includes a number of enhancements combined with improvements of the Cache subsystems to effectively deliver up to 2x performance improvement over its predecessor.

The Ampere SM allows RT Core and graphics, or RT Core and compute workloads to run concurrently to significantly accelerate many Ray Tracing operations.

In addition to the Ray-Traced game rendering benefits, the second-generation RT Core also delivers enormous improvements in workloads acceleration for:

- Photorealistic rendering of movie content
- Architectural design evaluations
- Virtual prototyping of product designs

Other improvements include:

- Rendering of Ray-Traced motion blur for faster results and greater visual accuracy.
- Rendering of complex models with physically accurate shadows, reflections, and refractions to empower users with instant insight.
- Powering truly interactive design workflows to provide immediate feedback for unprecedented levels of productivity, when working in concert with applications leveraging APIs such as NVIDIA OptiX, Microsoft DXR, and Vulkan Ray Tracing.

6.1.1.4 Larger and Faster Unified Shared Memory and L1 Data Cache

The Ampere SM used in the Orin SoC features a unified 192-KiB memory architecture for L1 Cache and Shared Memory. This unified structure is configurable to optimally allocate the most suitable L1 Cache and Shared Memory size for the workload at hand.

For compute mode, the L1 Cache/Shared Memory configurations are:

- 192-KiB L1 Cache + 0-KiB Shared Memory
- 184-KiB L1 Cache + 8-KiB Shared Memory
- 176-KiB L1 Cache + 16-KiB Shared Memory
- 160-KiB L1 Cache + 32-KiB Shared Memory
- 128-KiB L1 Cache + 64-KiB Shared Memory
- 92-KiB L1 Cache + 100-KiB Shared Memory
- 60-KiB L1 Cache + 132-KiB Shared Memory
- 28-KiB L1 cache + 164-KiB Shared Memory

For graphics and asynchronous compute mode, the 128-KiB memory structure is configured for:

- 128-KiB L1 Cache (data/texture)
- 48-KiB Shared Memory
- 16-KiB Reserved Usage (for various graphics pipeline operations)

The memory bandwidth of the unified memory structure is 128 bytes/clock.

6.1.1.5 Texture (TEX) unit

The TEX unit performs texture fetching and filtering. Beyond plain texture memory access, the TEX unit is also responsible for the following operations necessary to convert a texture Read request to a result:

- Addressing
- Level of Detail (LOD)
- Wrap
- Filter
- Format conversion

6.1.2 PolyMorph Engine

The PolyMorph Engine performs several primitives, typically triangle-based functions that occur at different stages of the graphics pipe. These functions include:

- Fetching per Vertex Attributes from Memory
- Topology Generation for Tessellation
- Viewport Clipping and Culling
- Attribute Plane Interpolation

6.1.3 Raster Engine

The Raster Engine deals with the realm of pixels in the graphics pipe. It takes the primitive vertices (typically triangles) from the PolyMorph Engine then determines which pixels on the screen are covered. In addition, the Raster Engine also performs the following:

- Depth testing to compute visibility of overlapping primitives (ZROP)
- Writing of final pixel colors (CROP)

The CROP unit is a fixed function unit that performs color blending of pixels/samples. The ZROP unit is a fixed function unit that performs depth testing of pixels/samples.

6.1.4 L2 Cache

All GPU units communicate to memory through the Level-2 (L2) Cache, also known as the L2. The L2 Cache sits between on-chip memory clients and the framebuffer. It is connected to the XBAR (to communicate with GPC units), the CROP, and ZROP units. In addition to providing the caching functionality, the L2 also includes hardware to perform compression and global atomics.

6.1.5 Host Interface

The Host Interface provides a path to assign work to do for the GPU.

6.1.6 MSS Interface

The MSS interface provides a full bandwidth path for the GPU to read from and write to system memory.

7. Multimedia Complex

7.1 Host Controller

7.1.1 Overview

The Host Controller provides a sophisticated programming and control interface to various SoC controllers, referred to as host clients. Commands are either gathered from a push-buffer in memory or provided directly by the CPU, and then supplied to the clients behind the Host Controller via Host channels. The channels also provide a means of synchronization between software and any individual block, or among the blocks themselves via hardware Sync Point Signals (Syncpts).

7.1.1.1 List of References

This chapter may require knowledge of the following document available from Arm®.

- *Arm Security technology, Building a Secure System using TrustZone® Technology* http://infocenter.arm.com/help/topic/com.arm.doc.prd29-genc-009492c/PRD29-GENC-009492C_trustzone_security_whitepaper.pdf
- *Arm AXI Bus Standard*

7.1.1.2 Glossary

This glossary shows Host Controller specific terms. Refer also to the glossary in the Introduction chapter of this TRM for further reference.

Term	Definition
CDMA	Command DMA. It is responsible for reading commands from memory and supplying them to all channels. It starts from the address in the channel's DMASTART register and continues until it reaches the address in the channel's DMAPUT register.
Channel	A piece of hardware that provides a programming interface to one or more classes. A channel contains a sequence of commands embodied in a command FIFO. This sequence of commands can also be thought of as a thread of execution and of a single context. There exists only one sequence of commands or context per channel. That is to say, there is no hardware-managed context switching within a single channel.

Term	Definition
Channel Commands	Entries in the command FIFO. Commands take a common form interpretable by Host and are used for three main functions: controlling class ownership and class virtualization of the channel; command expansion via gather commands; and issuing class methods.
Channel Switch	A transfer of ownership of a client from one channel to another; a type of context switch. Sometimes a source of confusion, this is not a transfer of ownership of a channel, but of a client. This is the preferred narrowed nomenclature to context switch in order to avoid confusion.
Class	An abstraction of a client, device, or resource. It is a collection of methods. A class can be owned by multiple channels, but only one channel may actively be using any class, which is known as a channel's working class. The transfer of working class from one channel to another is known as a context switch, which is managed by Host.
Class ID	A unique tag corresponding to each class.
Class Method	A method that belongs to an unspecified class.
Class Switch	A change in the current in the active class of a client; a type of context switch. In the event that a class switch also involves a channel switch, channel switch is the preferred declaration.
Central RM MMIO Range	Referring to the MMIO aperture use by central RM or RM server Software layer.
Client, Device, Resource	A piece of hardware that resides behind Host, connected via the HRD and HRW buses. Client is the preferred terminology for this document. Generally mapped one-to-one with a class, although this is not a strict requirement.
Command FIFO	Holds channel commands supplied by the CDMA or PIO accesses. It is stalled by synchronization methods, backpressure from its destination client, or ownership of the destination client by a different channel
Context Switch	A Host event where it facilitates a client's state transition. A context switch is either a channel switch or a class switch.
CPU	Referring to the Carmel processors in CCPLEX.
Direct Register Access	Access to a client initiated by the CPU. When a channel and the CPU initiate a data transaction to the same client at the same time, the CPU gets priority. Direct register accesses are orthogonal to channels in regards to client ownership.
DMAGET Register	Holding the current address of the CDMA. One per channel.
DMAPUT Register	Holding the end address of a command stream in memory. One per channel.
DMASTART Register	Holding the start address of a command stream in memory. One per channel.
DOS	Denial of Service. Referring to Denial of Service attack; considering a case where a software thread running on some Guest OS locks an engine (camera, video encoder, video decoder, etc.) for very long time, as result, other software thread running on another OS gets no resources (starved).
Gather	A channel command to gather contiguous chunks of memory and inserts it into the command stream.

Term	Definition
GB	Gather Buffer. A block in memory from which the Gather command gathers contiguous chunks of memory from.
Host Master	General term to indicate an entity that can control Host. The current list is the CPU, AVP, and TSEC.
Host Method	A method that belongs to the Host class.
HRD	Host Controller Read Bus
Increment	A channel command that specifies an offset and a count. Increment works like non-increment, except the offset is incremented by one on each on each write.
Inter-OS StreamID	Inter-OS StreamIDs refer to the different StreamIDs assigned to different Guest OS's. The Inter-OS StreamIDs allow hardware to implement security among Guest OS's based on the different Inter-OS StreamIDs assigned, thus achieving Inter-OS StreamID Protection.
Intra-OS StreamID	Intra-OS StreamIDs refer to the StreamIDs allocated to the different kernels (for applications, i.e., SMMU contexts.) of a specific Guest OS. The Intra-OS StreamIDs facilitate security protection of these applications (SMMU contexts) on the same Guest OS, thus achieving Intra-OS StreamID Protection.
KMD	Kernel Mode Driver, also referred to as Resource Manager (RM). In each Guest OS, there is a central Kernel responsible for the resource allocation and managing the different applications running in that OS.
Mask	A channel command that specifies an offset and a mask. A mask command works much like increment and non-increment, except the offsets are calculated by looking at the bits set in the mask. The bit position indicates the relative offset from the specified offset in the command. Mask expects the number of words to follow to be equal to the number of bits set in the mask.
Method	An operation on a class. The most common example is a single register write.
Nonincrement	A channel command that specifies an offset and a count. The offset specifies a method in the current class and the count indicates the number of subsequent words to be written at that offset.
PB	Push Buffer. A set of commands residing contiguously in memory. It is a communication method between the processor and Host. Commands are placed at the end of the buffer and a pointer is updated in Host.
RAR	Read After Read. Referring to the ordering between Read after Read transactions, where the next Read response comes only after the previous Read response.
RM	Resource Manager. Referring to the software layer responsible for managing hardware resources.
SetClass	A channel command that takes a class ID as an argument. SetClass is the sole means to indicate the current virtualization of the channel. It also the sole means that a channel can acquire ownership of a class. Subsequent channel commands are directed towards this class.
SO	Strongly Ordered. Referring to non-posted Writes where the master waits for ACK from the client before sending the next Write transactions.

Term	Definition
StreamID	StreamID refers to specific SMMU context in system memory used to isolate system memory among different Guest OS's for security reasons.
Sync Point (Syncpt)	A counter used for synchronization. Every time a specified event occurs, the counter is incremented. A channel can wait until the Syncpt attains a specified value, or an Interrupt can be generated when a specified value is reached.
Teardown	Either a module or channel teardown. Essentially, it means all links and references within Host between the specified module or the specified channel are removed and reset.
Tick Count	Running count of the Host clock after it has been enabled.
WAIT	A host method that takes a single vector argument. It is used in conjunction with Syncpt. A WAIT stalls the command FIFO until the supplied vector intersects the Syncpt register.
WAW	Write After Write. Referring to the ordering between Write after Write transactions, where the next Write transaction completes only after the previous Write transaction.

7.1.1.3 Relevant Chapters in the TRM

- Activity Monitor (ACTMON)
- Address Map
- Clock Controller and Reset (CAR)
- Control Backbone (CBB)
- Data Backbone (DBB)
- Display Interface (DisplayIF)
- Memory Controller (MC)

7.1.1.4 Features

- 63 Channels
- 704 Sync Points
- 32-bit Syncpt Comparison
- 32-bit Timeout Register
- Unit Activity Monitor (ACTMON): NVENC, VIC, NVDEC, and NVJPG
- Support for VI, ISP, SOR0, SOR1, SOR2, SOR3, DPAUX0, PVA, and DLA clients
- TrustZone® secure bit in Push Buffer (PB) path
- Master Interface protocol: AXI 3.0
- Client Interface protocol: HWR/HRD
- 40-bit Host1x2MC address
- Virtualization support: eight Guest OSs and one Central RM

- Syncpt protection per channel
- RAM-based Syncpts
- Asymmetric client MMIO address range

7.1.2 Functional Description

The Host Controller implements Channels and Classes to achieve sophisticated programming and control interface to its clients.

7.1.2.1 Channels

A channel is a thread of execution within the Host Controller. There are 63 Host Controller channels. Each channel has a set of registers and a command FIFO. Each channel can be associated with one or more Host Controller clients. The channel is the primary means of delivering commands to clients, and is described in the Programming Guidelines section.

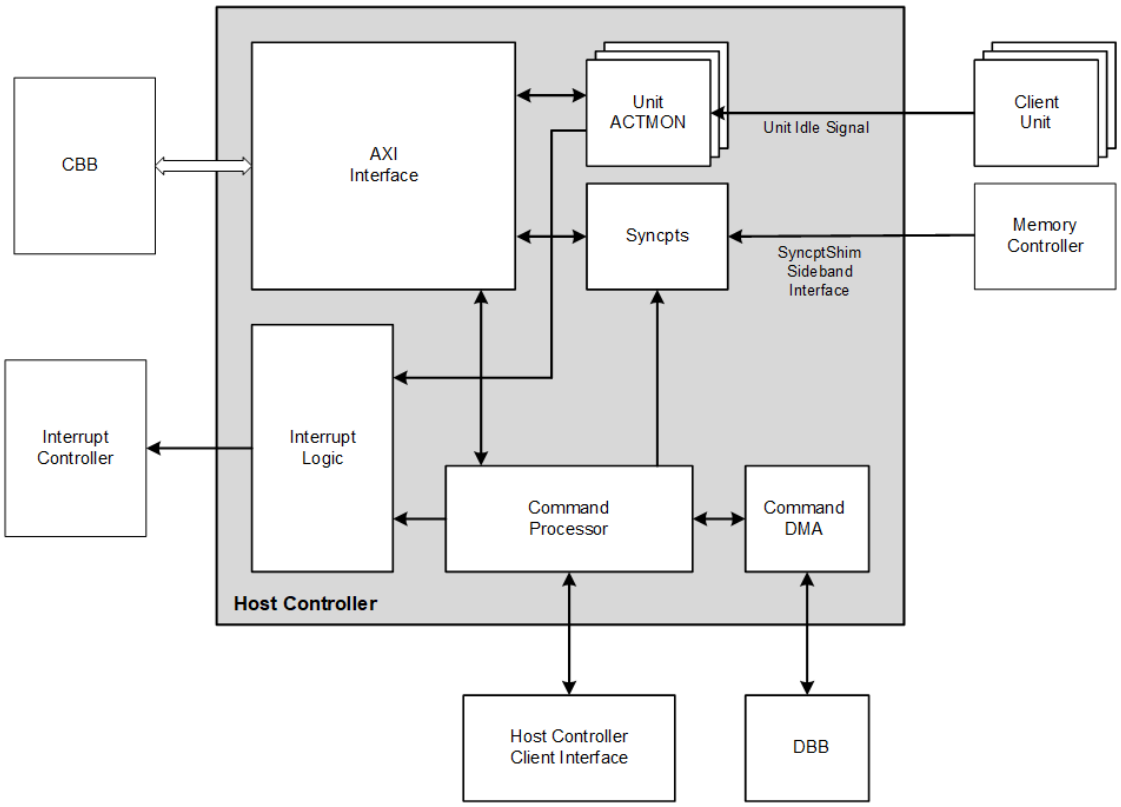
7.1.2.1.1 Multiple Channels

Similar to a multi-threaded CPU, a channel helps define a context that can be used to allow multiple users of the Host Controller, and plays a role in context switching. A context switch is sent to the client when the channel that programs the client is changed. There is a state change within the Host Controller. However, when channels have non-overlapping usages (for example a VIC channel and NVENC encode channel), there is no need to send context switches.

7.1.2.2 Classes

A Class is an abstraction of a client, device, or resource. It is a collection of methods. A class can be owned by multiple channels; but only one channel may actively be using any class, which is known as a channel's working class. The transfer of working class from one channel to another is known as a context switch and is managed by Host.

Figure 7.1 Host Controller Top-Level Block Diagram



The Host Controller connects to the Control Backbone (CBB) via its AXI Interface block.

The incoming MMIO commands from system NoC are either processed inside the Host Controller (Host Controller specific) or routed to its clients.

The Host Controller has a point-to-point interface with its clients used for sending requests to and accepting responses or Syncpt requests from its clients.

The Memory Controller interface to fetch Push-Buffer commands from memory through the Command DMA unit. The incoming Push-Buffer (PB) commands are processed inside Command Processor and afterwards either consumed inside the Host Controller or it generates transactions to the client interface. There is an internal Syncpt unit to synchronize between the Host Controller clients and also between software.

The Unit ACTMON block monitors activities of the Host Controller clients. The ACTMON statistics can be used by software for power management.

7.1.2.3 AXI Interface Block

This block is responsible for AXI3.0 to internal HWR/HRD protocol conversions as well as maintaining ordering rules.

It also has a timeout block to handle those cases where an MMIO transaction to a client got stuck either due to clock gating or power gating in the clients. We want to rely on Host Controller based timeout because the other CBB based timeout require system reset.

7.1.2.3.1 CPU Interface

The key features of this interface are listed below.

- **Outstanding Reads**

  Host Controller supports 16 outstanding transactions.

- **Ordering**

  RAR

  The Host Controller maintains RAR ordering for read requests from each Master. The Host Controller has a shared FIFO to store the request TAG that is also sent to the client via HWR_host1x2<client>.channel field. This *channel* field is returned along with a read response from the client, and the Host Controller uses its value to reorder responses to send back to the same master.

  The read requests for different masters are not ordered and can be bypassed.

  WAW

  The Host Controller maintains WAW ordering for write requests within same master. However, the write requests for different masters are not ordered and can be bypassed. Therefore software has to do an explicit read after write to enforce ordering between different clients if required.

  Strongly Ordered (SO) and Normal Writes

  The Host Controller returns ACK for SO and normal writes as soon as it is able to send a request to the client interface.

- **Byte Support**

  The Host Controller and its clients do not support byte writes.

- **Burst Length and Type**

The Host Controller supports only INCR and FIXED burst types. The maximum burst length is 16 for burst type INCR and FIXED cases with 4-Byte aligned address.

- **Read Response Types**

  The Host Controller only supports OKEY and SLVERR responses. The BLF logic in CBB does not send any transaction outside of client range, so there won't be any decode error cases. The SLVERR is for Host Controller MMIO access time-out.

7.1.2.3.2 Timeout Mechanism

The Host Controller has a timeout mechanism to handle illegal accesses to its clients by generating Interrupts for unserviced CPU read/write requests via the timeout register HOST1X_THOST_COMMON_AXI_TIMEOUT_CFG_0. The Host Controller generates a timeout Interrupt and the violating address is stored in the following registers:

- HOST1X_THOST_COMMON_AXIREAD_TIMEOUT_ADDR_0
- HOST1X_THOST_COMMON_AXIWRITE_TIMEOUT_ADDR_0

This timeout register is 32-bit wide and provides timeout value of ~19 seconds.

Only the last timeout address is saved when multiple timeouts happen before software clears the timeout Interrupts.

7.1.2.4 Syncpt (Sync Points) Block

The Host Controller employs a hardware mechanism, referred to as Sync Points, using hardware counters and Interrupts to achieve the following:

- synchronizing between software and Host Controller clients as well as in between Host Controller clients.
- stalling flows of commands to Host Controller clients.

A Sync point is a mechanism to synchronize between software and Host Controller clients and also in between the Host Controller clients. These are implemented as 32-bit counters which are incremented by 1 whenever some pre-destinated condition (or event) occurs. When a counter reaches its maximum value, it wraps back to zero on the next increment.

Synchronization using sync points can be done in the following ways:

- A CPU can be Interrupted when a sync point reaches a pre-specified value.
- A Host Controller channel can have wait commands for a channel to wait for a pre-specified sync point value.

Sync points do not normally reset, but can wrap; the comparison takes into account the possibility of wrapping.

Note: Sync point wrapping works only if (Syncpt value - Syncpt Threshold) <= 2 (syncpt_width-1)

For 16-bit Syncpt comparisons, the difference should be less than or equal to 32768. Software must take care of wrapping issues. For 32-bit comparisons, the difference should be less than or equal to 2,147,483,648. Because of the large value, software will not see any wrapping issues.

All Host Controller clients (e.g., VI) implement the following increment Syncpt method:

Incr_Syncpt<Condition><indx>

The Host Controller client would receive the Incr_Syncpt method and store the index for each condition. Whenever the "condition" event occurs, the client would return the index back to the Host Controller.

There are two basic ways for sync point increments:

1. When a CPU writes an index to the Host Controller's syncpt_cpu_incr register.
2. When a Host Controller client has received an incr_syncpt method and the condition specified by this method has become true.

The registers in Syncpt logic, including Syncpt registers and Threshold registers, are implemented with RAM to reduce the area.

The Syncpt registers can be read/written by many clients.

The write clients are:

- Host Controller clients
- Command processing logic
- Register write logic (direct and indirect paths)
- GPU interface

The read clients are:

- Command processing logic
- Syncpt Interrupt generation logic
- Register read Interface (MMIO path)
- SyncptShim interface

7.1.2.4.1 Virtualization

The Host Controller supports sharing of hardware resources between multiple Guest OSs in a time-sliced manner without impacting latency for high-priority thread in a Guest OS.

Software Partition for Virtualization

There can be eight Guest OS, one central RM driver, and one hypervisor in a virtualized system. Each guest OS has a dedicated RM/KMD partition driver to directly send commands to hardware without the need of central KMD/RM driver.

The hypervisor is responsible for static configurations like channel/Syncpt allocations and SMMU context IDs (StreamID) assignments per Guest OS. The Central RM handles runtime error, programming error, and timeout related error conditions.

7.1.2.5 Unit Activity Monitor (ACTMON)

The Host Controller implements two-unit ACTMON logic per unit for NVENC, NVDEC, VIC, NVJPG, NVJPG1, and OFA units. These Host Controller clients send their statistic information through a level signal (_monitor[1:0]). From the Host Controller to the unit ACTMON block there is register interface for MMIO access and interrupt interface to route unit ACTMON interrupts to CPU/BPMP.

7.1.2.6 Command Processor

The Command Processor processes the incoming responses for DMA requests to create an offset-data pair, which is either consumed by the Host Controller's internal registers or passed to the Host Controller's clients. It also includes logic to insert waits per channel based on syncpt values.

The Command Processor comprises Command Pre-processing, Channel RAM, and Command Post-processing as described below.

7.1.2.6.1 Command Pre-processing

The per-channel Command Pre-processing logic performs the following functions:

- Separating Class commands from DMA commands (Gather/Restart commands). The Gather and Restart command changes the address pointer of DMA for the particular channel and allows it to fetch from the address indicated through the Gather command.
- Conducting basic virtualization checks for the MLOCK Acquire and Release boundary as well as "Acquire MLOCK" and "SetClass" pairing.

7.1.2.6.2 Channel RAM

The per-channel FIFOs are implemented in a shared Channel RAM to allow software (the central RM) to resize these per-channel FIFOs based on the use cases.

7.1.2.6.3 Command Post-processing

The single Command Post-processing logic for all channels performs the following functions:

- Using round-robin arbitration to process the channel commands in the channel FIFOs of the active channels (i.e., channels with valid entries in the channel RAM and not stalled) to decide if they are destined for the Host Controller or its clients.
- Stalling a channel based on Syncpt wait. The stalled channel is placed in a wait state until Syncpt condition for that channel is true.

7.1.2.7 Command DMA Block

The Host Controller improves command programming efficiency by transferring commands in memory to the appropriate device, using command DMA. The CPU first assembles the command data in memory, and then programs the Host Controller engine to DMA it.

The Command DMA block has a DMA engine that is shared among all channels of the Host Controller. It uses a round-robin style arbitration for sending memory requests for all active channels. The arbitration boundary is fixed at 64 Bytes per channel to ensure fair bandwidth to all of the active channels.

For each channel, there is a dedicated set of DMA registers for configuring base and limit as well as dedicated control registers. Each channel also has a dedicated StreamID used along with memory requests. For TrustZone channels, DMA logic sends the corresponding TrustZone StreamID along with TrustZone attributes as part of the memory requests.

The incoming responses are stored in the Gather Buffer FIFO and Push Buffer FIFOs shared among all channels.

The command DMA block also has logic to avoid Head Of Line (HOL) blocking for those cases where a channel is stalled for a longer interval. Basically, the DMA block flushes out the command of the blocking channel from the gather or channel FIFO and send it again when the channel is ready. There is a channel timeout register to control the flushing interval per channel.

7.1.2.8 Interrupt Logic

The Interrupt logic collects both functional and error Interrupts then send them to the Interrupt Controller which can be either LIC or HSM or both.

7.1.2.8.1 Interrupts

The Host Controller (Host1x) has the following interrupt lines going to interrupt controller:

1. HOST1X2CPU0_GRAPHICS_INT(central RM/VMID0)
2. HOST1X2CPU0_SYNCPT_INT(central RM/VMID0)

3. HOST1X2VM1_[0..7]_SYNCPT_INT
4. HOST1X2VM2_[0..7]_SYNCPT_INT
5. HOST1X2VM3_[0..7]_SYNCPT_INT
6. HOST1X2VM4_[0..7]_SYNCPT_INT
7. HOST1X2VM5_[0..7]_SYNCPT_INT
8. HOST1X2VM6_[0..7]_SYNCPT_INT
9. HOST1X2VM7_[0..7]_SYNCPT_INT
10. HOST1X2VM8_[0..7]_SYNCPT_INT
11. HOST1X2CPU1_GRAPHICS_INT(Deprecated interrupt for BPMP)
12. HOST1X2CPU1_SYNCPT_INT(Deprecated interrupt for BPMP)
13. HOST1X2CPU2_SYNCPT_INT(camera proc0)
14. HOST1X2CPU3_SYNCPT_INT(camera proc1)

Note: Interrupt numbers 1-11 and 13-16 go to LIC, and interrupt number 12 is routed to BPMP.

7.1.2.8.2 Host Controller General Interrupts

Refer to the THOST registers for more information on these Interrupts.

7.1.2.8.3 Host Controller Syncpt Interrupts

The Host Controller can generate a Syncpt Interrupt upon reaching a threshold value by Syncpt registers. It can also generate an Interrupt to either CPU (referred to as CPU0 below) or AVP (referred to as CPU1 below).

An Interrupt is routed to cpuN when:

((SYNCPT<indx> >= SYNCPT_INT_THRESH<indx>) && (SYNCPT_THRESH_INT_MASK<indx> == cpuN) == 1)

The comparison takes wrapping into account. See the Syncpt (Sync Points) Block section for more information on wrapping.

The status is sticky and is PENDING until cleared. Write 1's to clear.

The Host Controller has 12 status registers corresponding to the following Interrupt lines:

- MPCORE(CPU0)
- BPMP(CPU1)
- MPCORE_OS[7:0]
- CAMERAPROC[1:0]

The status registers corresponding to each Guest OS resides into their corresponding address space. The Host Controller provides an additional Syncpt mask register that acts as a master control for Syncpt Interrupts.

7.1.2.9 Filtering out Kernel Commands from the Gather Buffer

This function maintains user mode driver versus kernel driver protection. The Host Controller kernel driver sets up mode bits that need to be protected from the user mode driver.

The Host Controller checks the incoming commands from the Gather Buffer (GB). When it contains the SetClass kernel mode command, the Host Controller drops these commands and raises an Invalid Gbuffer cmd Interrupt to software. Software reads the Interrupt status register to detect the channel that received the invalid gbuffer command. That particular channel needs to be restarted through the channel teardown mechanism.

Below is the list of privilege kernel commands.

- SetClass
- SetStreamID
- Extend
- Restart
- Restart_W

The following Interrupt status registers per channel manage an Invalid Gbuffer cmd Interrupt:

- HOST1X_THOST_SCHNL_CH< 62..0>_ILLEGAL_ACCESS_INTR_0
- HOST1X_THOST_SCHNL_CH < 62..0> _ILLEGAL_ACCESS_INTRMASK_0

There may be some cases where a channel can still see setClass, for example, context switching. For those channels, this feature must be disabled. This feature is controlled through HOST1X_THOST_COMMON_CH_KERNEL_FILTER_GBUFFER_0< 1..0> , which is under the central RM range.

7.1.2.10 SMMU Context Protection (StreamID Protection)

7.1.2.10.1 StreamID Switching of Host Controller Clients

In Host Controller clients, the software model user-mode driver (UMD) composes channel commands into a Gather Buffer which is then passed to the supervisor-mode RM software. The RM posts the gather buffer to a channel.

A single channel may hold requests from multiple processes. To isolate memory context of each process, StreamID switching per process is needed.

The StreamID switching commands are part of command stream to minimize latency in the MMIO path and simplify the software handshake mechanism.

7.1.2.10.2 Client StreamID Register

Each client creates a 7-bit StreamID register to store the current SMMU context.

Table 7.1 Client StreamID Register

6:0	STREAMID	i = 0x00	Indicates SMMU context; software can set an invalid StreamID at the end of its workload.

The client can create multiple StreamID registers depending on software requirements. The client also has to provide StreamID register ranges to the Host Controller so as to detect writes to this register through the MMIO path.

For the display engine, it is 12 offset registers corresponding to maximum requirement by display client corresponding to its four windows.

7.1.2.10.3 StreamID Update Mechanism

Channel-Based Clients

For channel based clients, the Host Controller supports a privilege command, SetStreamID to update the client's SMMU context mapping. This privilege command can be inserted only by RM partition not by user-mode driver; hardware filters out and indicates an error if it comes from any gather-buffer.

The Host Controller checks the incoming 8-bit StreamID value with StreamID table to check whether it is assigned to the Guest OS corresponding to that particular channel. In case the incoming StreamID does not belong to a particular Guest OS then the Host Controller drops this command and raise an illegal StreamID error to software (central RM). The channel is stalled until cleared by software.

This StreamID is written to the offset pointed out by offset field of SetStreamID command. The Host Controller also asserts the _secgroup[0] bit on HWR_host1x2client interface to indicate a privilege write.

Note: it is software's responsibility to make sure that the client is completely idle before initiating StreamID switching of client. Software uses the ENGINE_IDLE Syncpt in client to know whether the client is completely idle or not. Refer to the Standard Set of Incr_Syncpt Conditions section for more details.

MMIO-Based Clients

For MMIO programming model clients, the MMIO range of the client is assigned to one Guest OS only. In the absence of a dedicated 64 KiB-page MMIO range per client for the Guest OS, the Host Controller creates a mapping table under the central RM range to store the current Guest OS mapped to the client's MMIO range.

7.1.2.11 Direct Register Access

The Host Controller clients can have asymmetric address space which is assigned as part of global address map document.

7.1.2.11.1 Host Clients

The following table lists the Host Controller clients

Table 7.2 Host Controller Clients

Sr Number	Client	Address Width Bits)
1	SE1	16
2	SE2	16
3	SE4	16
4	SEU1SE1	16
5	SEU1SE2	16
6	SEU1SE4	16
7	VI_THI	16
8	VI_THI2	16
9	NVCSI	17
10	ISP	20
11	ISP_THI	16
12	PVA0_CLUSTER	22
13	DPAUX	16
14	VI	20
15	VI2	20
16	VIC	16
17	NVENC	16
18	NVDEC	16

Sr Number	Client	Address Width Bits)
19	NVJPG	16
20	NVJPG1	16
21	TSEC	16
22	NVDLA0	16
23	NVDLA1	16
24	OFA	16

7.1.2.11.2 Class IDs

The following table lists the class IDs.

Table 7.3 Host Controller Class IDs

Client	Class ID
NV_HOST1X_CLASS_ID	0x01
NV_VIDEO_ENCODE_NVENC_CLASS_ID	0x21
NV_VIDEO_STREAMING_VI_CLASS_ID	0x30
NV_VIDEO_STREAMING_VI_FALCON_CLASS_ID	0x31
NV_VIDEO_STREAMING_VI2_CLASS_ID	0xF6
NV_VIDEO_STREAMING_VI_FALCON2_CLASS_ID	0xF7
NV_VIDEO_STREAMING_ISP_CLASS_ID	0x32
NV_VIDEO_STREAMING_ISP_FALCON_CLASS_ID	0x33
NV_GRAPHICS_VIC_CLASS_ID	0x5D
NV_TSEC_CLASS_ID	0xE0
NV_NVJPG_CLASS_ID	0xC0
NV_NVJPG1_CLASS_ID	0x07
NV_NVDEC_CLASS_ID	0xF0
NV_NVCSI_CLASS_ID	0x38
NV_SE1_CLASS_ID	0x3A
NV_SE2_CLASS_ID	0x3B
NV_SE4_CLASS_ID	0x3D
NV_SEU1SE1_CLASS_ID	0xF9

Client	Class ID
NV_SEU1SE2_CLASS_ID	0xFA
NV_SEU1SE4_CLASS_ID	0xFB
NV_DPAUX_CLASS_ID	0x7D
NV_PVA0_CLASS_ID	0XF1
NV_DLA0_CLASS_ID	0XF3
NV_DLA1_CLASS_ID	0XF5
NV_OFA_CLASS_ID	0XF8

7.1.2.11.3 Host Controller Address Space

The Host Controller has a total address map size of 1152 KiB, which is divided into 64 KiB page size of sub ranges as follows. The VMID map per OS range is also shown here for Host Controller MMIO ranges.

Table 7.4 Host Controller MMIO Address Range

Range	Size	Map	VMID	Comments
0x13E0_0000-0x13E0_FFFF	64KB	Central RM	0x0	Contains runtime registers which will include interrupt related registers and channel-vm mapping registers
0x13E1_0000-0x13E1_FFFF	64KB	Hypervisor	0xf	Contains static configuration or resource allocation registers
0x13E2_0000-0x13E2_FFFF	64KB	BPMP/R5	NA	Contains registers which are to be accessed by BPMP/R5 only
0x13E3_0000-0x13E3_FFFF	64KB	HSM	NA	Contains registers which are accessed from HSM only
0x13E4_0000-0x13E4_FFFF	64KB	Guest OS 0	0x1	All Channel and syncpt registers.
0x13E5_0000-0x13E5_FFFF	64KB	Guest OS 1	0x2	
0x13E6_0000-0x13E6_FFFF	64KB	Guest OS 2	0x3	
0x13E7_0000-0x13E7_FFFF	64KB	Guest OS 3	0x4	
0x13E8_0000-0x13E8_FFFF	64KB	Guest OS 4	0x5	
0x13E9_0000-0x13E9_FFFF	64KB	Guest OS 5	0x6	

Range	Size	Map	VMID	Comments
0x13EA_0000-0x13EA_FFFF	64KB	Guest OS 6	0x7	
0x13EB_0000-0x13EB_FFFF	64KB	Guest OS 7	0x8	
0x13EF_0000-0x13EF_FFFF	64KB	Unit Actmon1	NA	
0x13F0_0000-0x13F0_FFFF	64KB	Unit Actmon2	NA	
0x13F1_0000-0x13F1_FFFF	64KB	Unit Actmon3	NA	
0x13F2_0000-0x13F2_FFFF	64KB	Unit Actmon4	NA	
0x13F3_0000-0x13F3_FFFF	64KB	Unit Actmon5	NA	
0x13F4_0000-0x13F4_FFFF	64KB	Unit Actmon6	NA	

7.1.2.12 Error Detection

Following are the error conditions for which the Host Controller raises an Interrupt to the central RM. In all of these error conditions, the particular channel is blocked and can be enabled only after channel teardown.

7.1.2.12.1 Timeout

The Host Controller hardware notifies the central RM through an error Interrupt when any channel does not get grant for some N numbers of cycles. This timeout value N is configurable per channel and can be set by the Guest OS.

7.1.2.12.2 Illegal MLOCK

The Host Controller hardware allocates MLOCK register per class ID which should be locked for setting up class for a channel. When any channel doesn't acquire MLOCK or the acquired MLOCK class doesn't match with the proceeding class through SetClass command, then an error condition results. Hardware sends an Interrupt to software (the central RM) to flag this error.

7.1.2.12.3 Illegal Gather Commands

The Host Controller hardware checks for privilege commands from Gather Buffer. When it contains any such command, it filters out that command and notifies the central RM through the CMDPP_ILLEGAL_OPCODE_INT Interrupt. This filtering mechanism is configurable through the

config register of each channel. This config register is protected from the Guest OS such that only the central RM can change filtering mechanism.

7.1.2.12.4 Illegal StreamID

When any Guest OS tries to use SMMU context from another Guest OS, the Host Controller hardware detects this condition and raises an error Interrupt (the central RM). The Host Controller compares the incoming StreamID set commands either through channel or MMIO path with the predefined list of StreamID table assigned to the specific Guest OS. This allows StreamID switch command to process only when the incoming StreamID value matches with its list of StreamIDs. Otherwise, it generates an illegal SetSmmuID Interrupt to the central RM.

7.1.2.12.5 Illegal Access

The Host Controller generates an Illegal Access Interrupt to the central RM when any Guest OS tries to access the shared resources not allowed for access either through Push-Buffer (channel) or MMIO path.

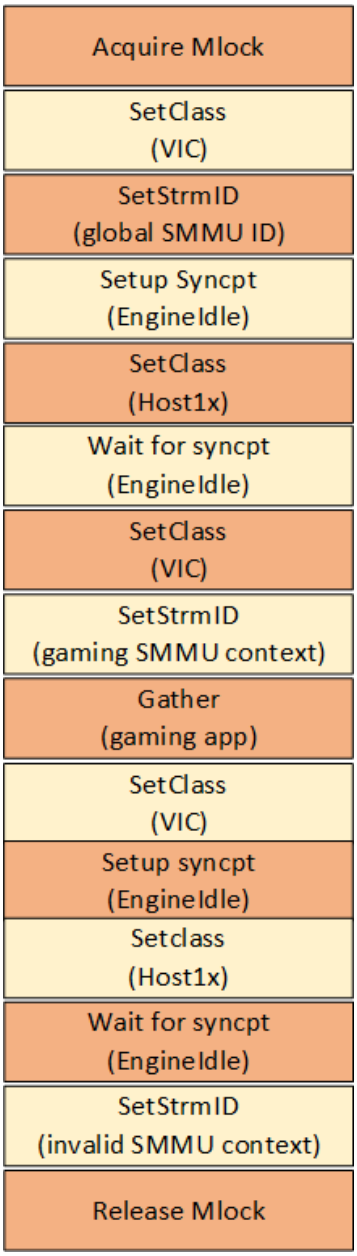
7.1.3 Programming Guidelines

7.1.3.1 Channel Programming Model

The Host Controller programming models reside on top of the concept of channels. A channel provides the means for software to supply an ordered sequence of commands to one or more classes. There are no restrictions on how many classes a channel may own or how often it can switch between classes, but it can only operate on one class at a time (known as the working class) and one command at a time. A stalled channel does not allow any commands to proceed from any class – there is a strict ordering of commands. A channel starts processing commands when commands are present, either supplied by the CPU (PIO mode) or gathered from memory. A channel can be stopped explicitly by a CPU write to channel state.

Normally, software does static allocation of a channel to one engine only as shown below.

Figure 7.2 Single Channel Example



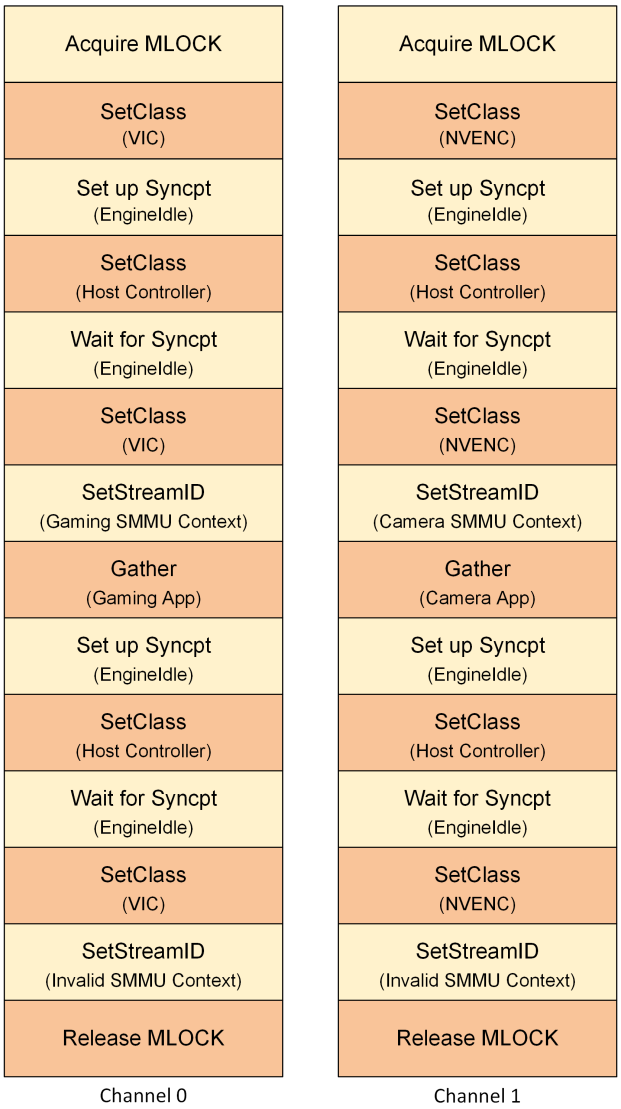
7.1.3.1.1 Concurrency

Channels operate in parallel and are unhindered by one another except in two specific cases:

- Synchronization points
- Class contention

The following figure depicts such concurrency.

Figure 7.3 Channel Concurrency Example

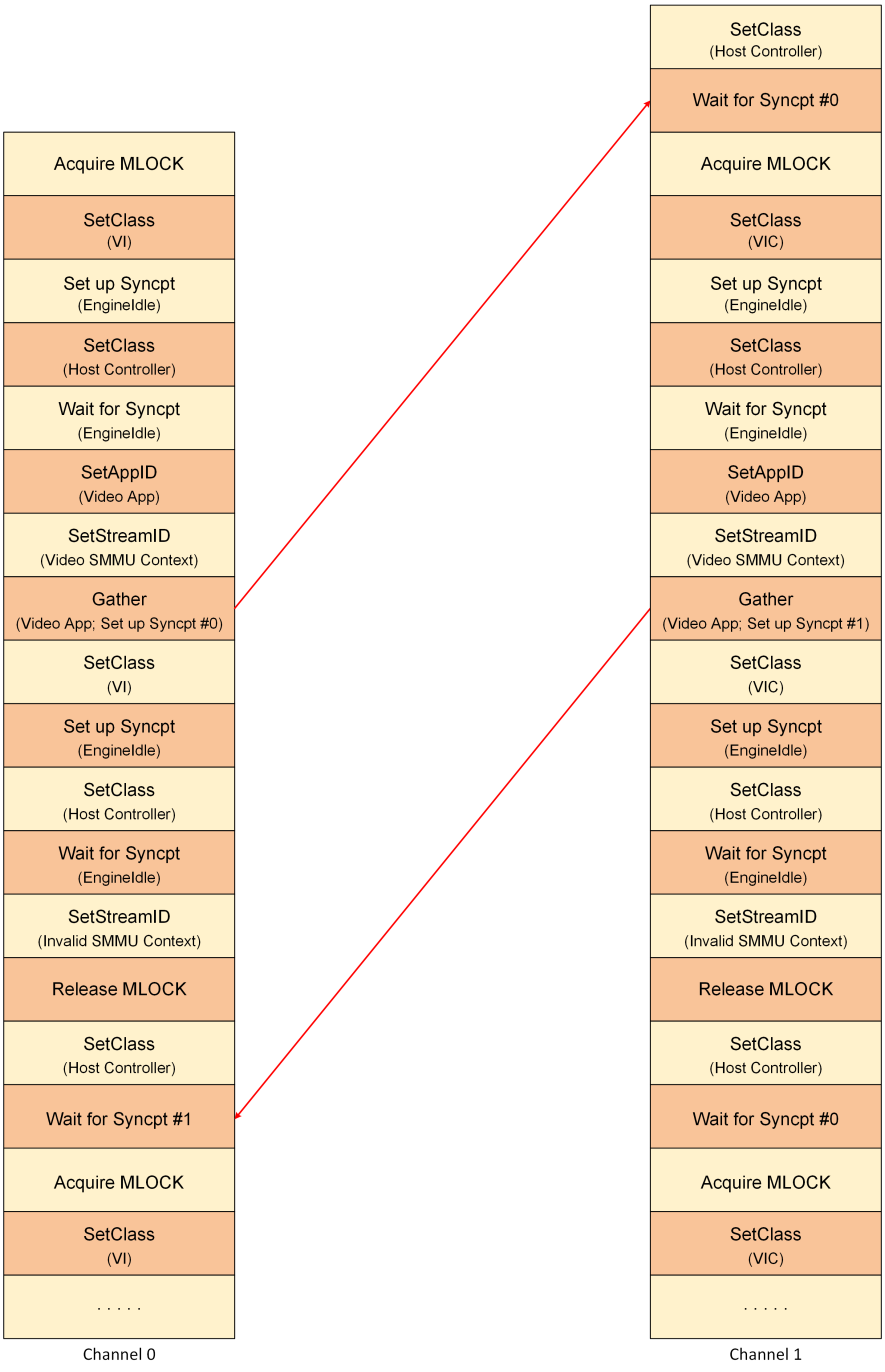


In the example above, no channel is blocked by another at any time so both can work concurrently. Care must be taken when assigning clients to channels. Channel concurrency as well as channel throughput is crucial for performance. For example, a client that is event-driven with a low latency requirement should exist in its own channel. Conversely, two clients that must be steadily supplied with commands could possibly coexist in the same channel given that their synchronization points are not conflicting (if they have any real synchronization points at all). Clients that operate sequentially on the same piece of data can easily reside in the same channel.

7.1.3.1.2 Synchronization

There are 704 total synchronization points (Syncpts). Similar to semaphores, Syncpt methods are issued to clients, mentioning Syncpt counter index and returning condition. Any channel can be made to wait on a Syncpt value of a particular Syncpt index through Host Controller "wait" method. Once the Syncpt method is received by clients, based on Syncpt condition, they return a Syncpt index back to the Host Controller. On receiving a Syncpt index from the client, The Host Controller increments that particular Syncpt counter. The following simple example shows how Syncpts and waits allow for synchronization. The arrows indicate channel dependencies (which channel is waiting for a Syncpt increment from another channel).

Figure 7.4 Channel Synchronization Example



7.1.3.1.3 Channel Registers

Channel registers are broken into two functional groups: one grouping is associated with the command FIFO and command delivery to clients; the other grouping is associated with register and memory access.

7.1.3.1.4 Channel Commands

Channel commands can be split into two categories:

- **Class commands**
  A class command is a mechanism to communicate a class write to a client. The channel must have an active class when processing class commands. The active class is set by a SETCL (SetClass) command.
- **DMA commands**
  DMA commands control what is being fetched. They are processed at the top of the command FIFO while class commands are processed at the bottom of the command FIFO. This is because DMA commands change the command stream and must be processed before entering the FIFO.

Offsets in class commands are relative to the active class' base as they are limited to only the methods in the active class.

- **SETCL** (SetClass, opcode 0x0)
  A channel command that takes a class ID as an argument. SetClass is the sole means to indicate the current ownership of the channel. It also means that a channel can acquire ownership of a class. Subsequent channel commands are directed towards this class.

  SetClass does not mean implicit acquisition of a module's ownership. It is possible to send commands simultaneously from more than one channel to the same module. When exclusive ownership of a module is required for software correctness, then ACQUIRE_MLOCK and RELEASE_MLOCK opcodes should be used to acquire and release the MLOCKn semaphore bits. An ACQUIRE_MLOCK that fails causes that channel's commands to stall until it wins subsequent MLOCK arbitration (arbitrations happen with each RELEASE_MLOCK). In summary:
- SetClass instructs the channel hardware which module and context to use for the following commands.
- If multiple channels write simultaneously to the same or different class IDs in the same hardware module (and no MLOCKs are used), then the actual commands are interleaved with a granularity of one command per channel.

  The SetClass behavior also supports virtualization use cases. See SETCL (SetClass, opcode 0x0, in the Virtualization section.)

- **HCFINCR** (Increment, opcode 0x1)
  Takes an offset and count as arguments. There are <count[15:0]> data following this

command to be written starting at the specified offset. The offset is incremented after each write.

- **HCFNONINCR** (Non-Increment, opcode 0x2)
  Takes an offset and count as arguments. There are <count[15:0]> data following this class command to be written to the specified offset. Non-increment indicates that the offset is not to be incremented per data write.
- **MASK** (Mask, opcode 0x3)
  Takes an offset and count as arguments. The number of data to write after the command is equal to the number of set bits in the count. Each data is written to the specified offset plus the next active bit location. For example, a count equal to 0x5 would have two data that are written to (offset+0) and (offset+2).
- **IMM** (Immediate, opcode 0x4)
  Takes an offset and 16-bit data as arguments. The 16-bit data is written to the offset.
- **EXTEND** (opcode 0xe)
  The EXTEND opcode includes ACQUIRE_MLOCK and RELEASE_MLOCK opcodes. See the Virtualization section for more details.
- ACQUIRE_MLOCK uses MLOCK registers, which are allocated by software. MLOCK (Module LOCK) bits are visible to all channels. ACQUIRE_MLOCK sets MLOCK<indx> to 1. However, when MLOCK<indx> is already 1, the channel stalls until the MLOCK arbiter grants the lock to that channel. A channel's priority can be set using the HOST1X_THOST_COMMON_CH_HIPRI_0 register. The ACQUIRE_MLOCK command is a privilege command.
- RELEASE_MLOCK clears the MLOCK<indx> bit when it is locked by the same channels (otherwise it is ignored). Thus, when one or more channels are waiting, the MLOCK arbiter picks a channel and allows that channel's ACQUIRE_MLOCK to complete. For virtualization use cases, the RELEASE_MLOCK command is a privilege command.
- **SETPYLD** (opcode 0x9, <Data[15:0]>)
  This command sets the internal 16-bit count register per channel to set the payload value to be used by the proceeding SETSTRMID/HCFINCR_W, HCFNONINCR_W commands.
- **HCFINCR_W** (opcode 0xa)
  This command is similar to HCFINCR command except that it uses a 22-bit offset value. The count value comes from an internal payload register which was set through a previous SETPYLD command.
- **HCFNONINCR_W** (opcode 0xb)
  This is similar to the HCFNONINCR command except that it uses a 22-bit offset value. The count value comes from an internal payload register which was set through a previous SETPYLD command.

7.1.3.1.5 Channel Command FIFO

The command FIFO is loaded either the CDMA or via PIO access. PIO is mainly used for debug purposes, but it allows a Host Controller master to fill the command FIFO with channel commands

by writing into the per channel CMDFIFO_PIOWR_0 register. PIO Command FIFO accesses should not be issued while a Gather process is busy.

A command FIFO has a set of registers that point to a location in memory where a sequence of channel commands resides. This sequence of commands is generally referred to as a Push-buffer.

A Host Controller master can write either to DMAPUT or DMASTART register to trigger command fetching by the CDMA. DMASTART indicates where to start fetching, and DMAPUT indicates where to stop. DMAGET is a read-only reference to the present location of the command FIFO; it is incremented when the command is popped from the command FIFO. DMAPUT and DMAGET are relative addresses to DMASTART.

DMAEND provides an upper boundary to prevent the CDMA from fetching illegal addresses; fetching ceases when DMAGET equals DMAEND.

The command FIFO can be halted by writing the DMASTOP field in the DMACTRL register. DMACTRL also provides a mechanism to reset the DMAGET pointer (DMAGETRST field) to either 0 or to the value of DMAPUT (DMAINITGET field).

7.1.3.1.6 Channel Control

Channel status and control resides in the synchronous register space of the Host Controller address map. These registers are aliased in each channel's space. The details of these registers are as below:

- **CH<0..8>_STATUS**
  Status includes client ownership and whether the channel is blocked or not.
- **CH_TEARDOWN**
  Using this register each channel can be reset, which is referred to as a teardown. This means all channel states are cleared and all client and class ownerships are relinquished.
- **MOD_TEARDOWN**
  Similar to channel teardown, there exists a mechanism to reset modules. Setting this register in the Host Controller can clear all states associated with a given client/module.
- **<client>_STATUS**
  This register indicates the client's currently active class.

7.1.3.2 Priority between Channels

The current MLOCK arbiter uses two round-robin rings, high and low priority. A high priority acquire request is always granted before a low priority request.

7.1.3.3 ACQUIRE_MLOCK

The ACQUIRE_ML OCK opcode replaces the implicit locking of modules previously done by HCFSSETCL. This is done using MLOCK registers, which is allocated by software. MLOCK (Module LOCK) bits are visible to all channels. ACQUIRE_MLOCK sets MLOCK [indx] to 1, but if MLOCK[indx] is already 1, then the channel stalls until the mlock arbiter grants the lock to that channel. The MLOCK arbiter uses two round-robin rings with high and low priority. A high-priority acquire request is always granted before a low-priority request. A channel's priority can be set using the HOST1X_SYNC_CH_PRIORITY_0 field.

7.1.3.4 Syncpt Protection

The Host Controller hardware provides inter OS Syncpt protection to support virtualization use cases and for single OS use cases. it adds inter-channel and inter-application Syncpt protection.

7.1.3.4.1 Inter OS Syncpt Protection

The Host Controller creates a Syncpt-VMID table under the central RM (VMID0) range through which the central RM allocates Syncpts to various Guest OSs.

Table 7.5 Syncpt-VMID Mapping Table

Syncpt Number	VMID[3:0]	VMID Reset Value
0	<vmID1>	<Undefined>
...	...	<Undefined>
703	<vmIDN-1>	<Undefined>

For each HWR transaction to clients the Host Controller sends VMID[3:0] field associated with that Guest OS as part of the 10-bit channel ID(VMID[3:0],ChID[5:0]) field. Clients are supposed to return this 10-bit channel ID field along with the Syncpt response packet.

Based on the VMID of the incoming Syncpt response packet (indicated through VNID[9:6] of channel ID[9:0] field) and stored VMID for corresponding Syncpt index, the Host Controller allows or denies Syncpt increment requests.

When it is an illegal request, the Host Controller also raises an illegalAccess error to the central RM.

For MMIO-based programming models, only Inter-OS checks are performed. Channel and appID checks are ignored in that case assuming the partition RM holding that MMIO range has already completed the check.

7.1.3.4.2 Intra OS Syncpt Protection

Within a single OS, the Host Controller adds Syncpt protection between channels associated with that channel and different applications sharing the same channel in that OS.

The Host Controller creates a Syncpt-ChAppID mapping table in each Guest OS range so that each OS can assign its own pool of channels to various channels and various applications within that OS.

Table 7.6 Syncpt-ChAppID Mapping Table [All MMIO ranges]

Syncpt Number	Application ID[7:0]	ChID[5:0]
0	<appID0>	<chID0>
...	...	...
703	<appIDn-1>	<chIDn-1>

AppID is 8-bit wide to support the maximum 256 applications per channel.

For the MMIO-based programming model, the channel ID is 0x3F to indicate MMIO channel.

For each hardware transaction to clients, the Host Controller sends the vector in ChID[5:0] associated with the channel number or MMIO range of the Guest OS together as the channel ID[9:0] field (= {VMID[3:0],ChID[5:0]}).

On the response packets, the clients return this channel ID[9:0] field and based on the ChID of the incoming Syncpt response packet and stored ChID for the corresponding Syncpt index the Host Controller allows or denies Syncpt increment request.

When an illegal request occurs, the Host Controller also raise an illegalAccess error to the central RM. An additional status register indicates this Inter-channel illegal access.

Note that hardware does not support APPID protection for THOST class. Either software need to prevent THOST class from GB or only one application should be used per channel in a given time so that APPID protection is not required.

7.1.3.4.3 Client Changes to Support Channel ID

The Syncpt interfacing logic of clients stores the incoming channel ID [9:0] field along with Syncpt request packets per Syncpt.

For continuous Syncpt registers, the clients store the channel field which is a part of the MMIO write command.

Once the Syncpt setup condition is true, the clients return the corresponding stored channel value along with the Syncpt response packet over the HRD interface.

7.1.3.5 DMA Commands

- **RESTART** (Restart or Jump, opcode 0x5)
  This command specifies an offset relative to DMASTART. The next command to be fetched is from (DMASTART + offset). The jump address granularity is 16 bytes because the lower 4 bits are always zeroes.
- **RESTART_W** (Restart or Jump, opcode 0xd)
  This command is similar to the legacy RESTART command with the following differences:
  - It consists of two words, (Lower 28 bits, Upper 10 bits)
  - The jump address granularity is 4 bytes because the lower 2 bits are always zeroes.
- **GATHER** (opcode 0x6)
  This command comprises of two words and has arguments: <offset, insert, type, count> and <address>. When GATHER is processed, <count> data is fetched from the address and inserted into the command stream. When the <insert> argument is enabled, it inserts either an HCFINCR or HCFNONINCR opcode preceding the fetched data, which is specified by <type>. DMA commands do not need an active class, but often are processed when an active class exists. In the case of the GATHER command, when <insert> is enabled, there must be an active class.

Gather_0		
31:28	OPCODE	GATHER = 0x6
27:16	OFFSET	Starting offset (if inserting opcode into the stream)
15	INSERT	Insert an opcode (type set by TYPE, initial offset set by OFFSET), immediately before the gathered data stream
14	Type	Type of opcode to be inserted (incrementing or non-incrementing). Only valid if INSERT is enabled. 0: HCFNONINCR 1: HCFINCR
13:0	Count	Count of 32-bit words to be gathered from memory
Gather_1		
31:2	Address	Base address of memory

  While sending the address to memory, the Host Controller sets the upper 8-bit address to 0 to support 40-bit addressing.
- **GATHER_W (opcode 0xc)**
  This gather command provides 40-bit address support. It is similar to the legacy GATHER command but with wide address support and without the insert feature. The gather stream can have self-contained commands only. This command comprises three words with the following arguments: Count, Lower 32-bit address, Upper 32-bit address.

Gather_W_0		
31:28	OPCODE	GATHER_H = 0xc

15:0	Count	Count of 32-bit words to be gathered from memory
Gather_W_1		
31:2	Addr_Low	Lower 32-bit base address
Gather_W_2		
7:0	Addr_High	Upper 8-bit base address

When GATHER_W is processed, <Count> data is fetched from MC base address (Addr_High, Addr_Low) and inserted into the command stream.

7.1.3.5.1 Command DMA (CDMA) Registers

- **HOST1X_THOST_CHANNEL_CH*n*_DMASTART_0 (*n* is the channel number.)**
  This register holds the start address of memory location for fetching DMA for the channel as indicated.
- **HOST1X_THOST_CHANNEL_CH*n*_DMASTART_HI_0 (*n* is the channel number.)**
  This register holds the upper 8 bits of the DMASTART memory address to support 40-bit MC addressing for the channel as indicated.
- **HOST1X_THOST_CHANNEL_CH*n*_DMAPUT_0 (*n* is the channel number.)**
  This register triggers a DMA fetch from memory for the channel as specified, when the PUT register does not equal the GET register. This address is relative to the DMASTART base address. It does not support byte writes. All 4-byte data needs to be programmed.
- **HOST1X_THOST_CHANNEL_CH*n*_DMAPUT_HI_0 (*n* is the channel number.)**
  This register holds the upper 8 bits of the DMAPUT memory address for the channel as indicated.
- **HOST1X_THOST_CHANNEL_CH*n*_DMAGET_0 (*n* is the channel number.)**
  This register tracks the MC offset, which the DMA engine has read. It gets incremented as entries are loaded from the channels command buffer into the FIFO. This address is relative to the DMASTART base address for the channel as specified..
- **HOST1X_THOST_CHANNEL_CH*n*_DMAGET_HI_0 (*n* is the channel number.)**
  This register holds the upper 8 bits of the DMAGET memory address.
- **HOST1X_THOST_CHANNEL_CH*n*_DMAEND_0 (*n* is the channel number.)**
  This is the boundary of illegal addresses (at the end of either the push buffer or physical memory) and is an offset relative to the DMASTART address for the channel as specified. This is designed to prevent DMA from prefetching illegal addresses. When DMA reaches this address before seeing a RESTART, it stops. This would be a software error condition.
- **HOST1X_THOST_CHANNEL_CH*n*_DMAEND_HI_0 (*n* is the channel number.)**
  This register holds the upper 8 bits of the DMAEND memory address for the channel as specified.
- **HOST1X_THOST_CHANNEL_CH*n*_DMACTRL_0 (*n* is the channel number.)**
  The various fields of DMA control register for the channel as specified are described below:

- **DMAGETRST**
  Reset GET pointer to 0. Useful for cleaning up a crashed channel. DMAGET value is not updated instantly. It takes four cycles between programming of reset and valid DMAGET.
- **DMAGETINIT**
  Reset GET pointer to the value of DMAPUT when the DMAGETRST field is asserted.
- **DMASTOP**
  Stop DMA from fetching on the channel.

A Command DMA channel needs to be enabled for PIO-gather to work.

7.1.3.5.2 Access Registers

- **HOST1X_THOST_CHANNEL_CH*n*_FIFOSTAT_0 (*n* is the channel number.)**
  CFNUMEMPTY is the number of free slots available in the per-channel command FIFO (needed for PIO or polling for completion of a wait).

7.1.3.6 Progress Status

Channel progress is monitored through two means – GET and Syncpts. GET is a channel state that indicates the address of the last command that has been fetched from memory and sent to the channel. GET is the same as DMAGET. Syncpts are counter registers that are incremented when specified events occur (e.g., after the completion of each operation done by a module). The 32-bit Syncpt values typically are monotonically increasing and can be used for in-channel waits and to indicate channel completion status. The GET and Syncpt registers can also be read directly (out-of-band) by the CPU.

7.1.3.7 Programming Model for MMIO-Based Clients (Single MMIO Range)

1. The client MMIO range is statically assigned to only a single Guest OS other than the central RM(VMID0).
2. That particular Guest OS maintains all SMMU contexts in clients. Basically, all StreamID registers of clients.
3. The Host Controller hardware needs to protect the StreamID update from this Guest OS (through MMIO path) and not conflict with other Guest OSs (Inter-OS StreamID protection)
4. Similarly for Syncpts, the Host Controller hardware only does Inter-OS protection.
5. The Intra-OS protection for StreamID and Syncpt is the responsibility of the Guest OS only, which owns that particular MMIO range.

7.1.3.8 Programming Model for MMIO-Based Clients (Multiple MMIO Range)

1. The client has dedicated MMIO ranges for each Guest OS, and each MMIO range is statically assigned to only a single Guest OS other than the central RM (VMID0).
2. That particular Guest OS maintains the SMMU contexts within its range in the client.
3. Host Controller hardware needs to ensure that the StreamID update from this Guest OS (through the MMIO path) does not conflict with other Guest OS (Inter-OS StreamID protection).
4. Similarly for Syncpt, the Host Controller hardware does Inter-OS protection only.
5. The Intra-OS protection for StreamID and Syncpt is the responsibility of the Guest OS that owns that particular MMIO range.

7.1.3.8.1 Basics of Syncpt-Based Programming Model

The basic programming model that software follows is:

Each module should be programmed to do a unit of work (an operation) by the Host Controller using CDMA and Push Buffers (PBs).

Examples of an operation include:

- BLT (VIC)
- Draw a set of triangles (GPU)
- Encode a single frame (NVENC)

When nothing else is programmed, module goes idle until the Host Controller sends commands to start another operation (no continuous mode). To do its operation, a module reads data from memory and writes the results to memory. Modules interact with each other using memory buffers: one module is the producer of data, and another is the consumer of that data.

7.1.3.8.2 Basic Synchronization

There are two basic needs for synchronization: management of memory buffers and timing of control register writes. Memory buffers used to pass data from one module to the next use a producer/consumer model with circular buffers. To prevent buffer underflow and overflow, synchronization needs to be done in both directions:

- The consumer cannot read until the producer is done writing (and the writes are committed to memory).
- The producer cannot reuse an output buffer (i.e., write to buffer) until the consumer is done reading the buffer.

Thus, the synchronization events required for memory buffers are:

- The module has completed all reads from the buffer.
- The module has completed all writes to the buffer (and they are committed by the memory controller).

To understand the requirements for timing the writes to control registers, a typical sequence is provided below:

1. reg wr for operation A
2. reg wr for operation A
3. reg wr (trigger) for operation A
4. reg wr for operation B
5. reg wr for operation B
6. reg wr (trigger) for operation B

When no WAIT method is placed between the trigger for A and the first register write for B, then in the worst case, corruption of operation A may occur because the new value of the control register is used before operation A is done. For modules that protect against this corruption, there is still the undesirable behavior of the module delaying the register write and subsequently causing back pressure on the Host Controller's write bus. For direct register reads by ISRs occurring asynchronously with respect to the channel command writes, it must be ensured that the Host Controller bus does not stall for significant periods of time. To synchronize writes to a control register, a safe time to start writing the register for the next operation is when:

- No corruption occurs for previous operations.
- No stalls of the HWRBUS bus.

7.1.3.8.3 Standard Set of Incr_Syncpt Conditions

The following values of the "incr_syncpt cond" field are predefined:

- **0** (Immediate): Return indx to Host Controller immediately (used by software push-buffer allocation and helpful for debug).
- **1** (Op_Done): Return indx to Host Controller when all previously triggered operations have completed and their writes to memory are committed.
- **2** (Rd_Done): Return indx to Host Controller when all previously triggered operations have completed their reads from memory.
- **3** (Reg_Wr_Safe): Return indx to Host Controller when it is safe to program registers for the next operation. Safe means no corruption to previous operations, and no stalling of Host Controller write bus should occur.
- **4** (Engine_Idle): The engine returns Syncpt index to Host Controller only when all previously queued operations to an engine are completed including commitment of their memory writes.

There are special cases which require the use of additional condition values. Some modules have multiple read buffers, thus requiring (condition == 2) that indicates all reads to all buffers

complete. It is useful for software to know when reads are done to a specific input buffer. For some modules, e.g., VI, there are different safe times to update different sets of registers, so (condition == 3) needs several variations (one for each "safe time"). If a module can have two operations happening at one time (as in VI), then special considerations needs to be made for these cases: either two separate Incr_Syncpt methods or one Incr_Syncpt method with many special conditions. For some modules, one condition can replace another if the two conditions always happen within a short period of time of each other. Then the condition that always happens last can be used for both.

7.1.3.8.4 Continuous Mode — Display

For each display, software wants a free-running Syncpt increment on every display "Vsync". The most appropriate implementation of this is to have an additional register which has "Enable" and "Indx" fields to control the increment of Syncpt on every Vsync event. If enable is set, then on every Vsync display would return this "Indx" back to the Host Controller.

Host Controller clients implements the logic associated with these registers. When this continuous mode is enabled, the client should set a pending bit for when the condition occurs. When the pending bit is set, the client arbitrates for the hrd_<client>2host1x bus and when selected sends "Indx" tagged with "Syncpt type" on the hrd_ bus. After successfully sending the Indx on the HRD bus, the client clears the pending bit.

For the continuous syncpt case, the client has to implement logic to return the TrustZone flag and "channel" fields back to Host Controller along with syncpt response which were received earlier as part of the continuous syncpt update request.

See the **Host Clients** section for a list of the Host Controller clients.

7.1.3.8.5 Allowing Multiple Pending INCR_SYNCPTs

Clients can have multiple pending Syncpts which are queued into the client's Syncpt FIFOs, which are dedicated ones for each Syncpt condition. The behavior of these Syncpt FIFOs is controlled by an "INCR_SYNCPT_CNTRL" register in the client's register space:

7.1.3.8.6 Syncpt Protection

The Host Controller supports the following types of protection mechanisms for Syncpts:

- Syncpt Protection Between Multiple Guest OS

  To support virtualization use cases, the Host Controller protects Syncpts between Guest OS. Host Controller hardware creates a Syncpt-VMID Mapping table to store Syncpt allocation which is statically done by the central RM (VMID0) between Guest OSs. Based on this table, Host Controller allows or denies Syncpt increment initiated from an OS.
- Within a Single Guest OS Syncpt Protection Between Multiple Channels

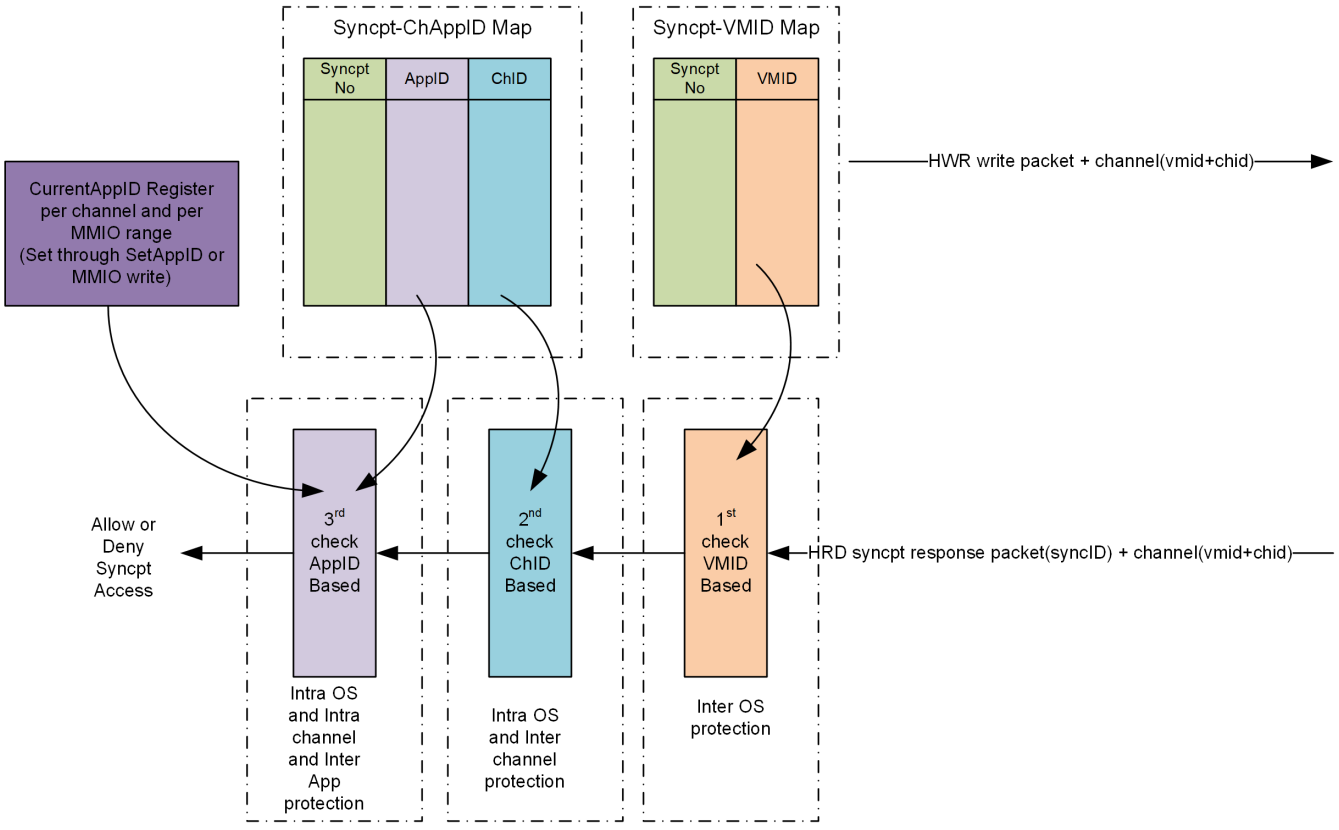
Each Guest OS allocates Syncpts from its assigned pool of Syncpts to various channels through a Syncpt-VMID Mapping table. Based on this table, the Host Controller protects increments of Syncpts from another channel within that Guest OS.

- Within a Single Channel of a Single Guest OS Syncpt Protection Between Multiple Applications

  The Host Controller protects increments of Syncpt by various applications which are sharing the same channel in a Guest OS. This is done through a Syncpt-VMID Mapping table, where each Guest OS assigns Syncpts to various applications sharing same channel. There is also a hardware register per channel and per MMIO range to store the current AppID using that channel or MMIO range.

  Based on this hardware register and Syncpt Mapping Table, the Host Controller does Syncpt protection between various apps sharing the same channel within a Guest OS as shown in the following diagram.

Figure 7.5 Syncpt Protection



7.1.3.8.7 Virtualization Methodology

The Host Controller supports virtualization with dedicated hardware resources (channels and Syncpts) with arbitration through partitioned RM (inside Guest OS) based privilege commands. The basic assumption here is that each partition RM can be trusted to send kernel specific commands for channel switching. Also, each submit from Guest OS contains commands for only one frame so that switching latency cannot be more than one frame.

The Host Controller implements dedicated 63 channels and 1024 Syncpts as required by software to support eight Guest OSs with no change in behavior of channel functionality. Channels and Syncpts are allocated to each Guest OS by Hypervisor and hardware protects it from interference from other Guest OSs.

Mapping of channels and Syncpts to a Guest OS can be done in an asymmetric manner so that based on configuration, the Hypervisor can change allocation of Syncpt and channels.

Host Controller supports an 7-bit StreamID register for storing SMMU context. This StreamID configuration in the client is done by the partition RM inside each Guest OS. This partition RM can select StreamID from a list assigned to it by the hypervisor. The hypervisor also configures the list of StreamID per Guest OS inside the Host Controller, and the Host Controller checks the incoming StreamID config value from each Guest from this list and that is allowed to go to client only if it matches this list.

A Guest OS uses the SETSTRMID privilege command to set the SMMU context for an engine class from its allocated StreamID list. Within a single Guest OS it is the responsibility of partitioned RM (inside Guest OS) to make sure that StreamID values do not overlap between multiple threads.

The push-buffer fetch from each channel uses distinct SMMU context, configured through the 7-bit StreamID configuration register per channel, which is programmed by the partition RM of the Guest OS while hardware ensures that it does not interfere with other StreamIDs.

The arbitration of an engine between groups of channels is done through partitioned RM based privilege commands inserted at each submit boundary. These are ML OCK_ACQUIRE/RELEASE commands that are privilege and compulsory used to pin an engine to channel unless it is released through MLOCK_RELEASE command from the same channel. So, a channel not inserting ACQUIRE_MLOCK prior to SetClass is not allowed to use that class.

Because channel switching happens at the submit boundary only, no engine context switching is required. Partitioned RM inserts proper SMMU context–switching commands at each submit boundary to ensure proper context.

To prevent corruption from UMD, the Host Controller hardware filters out privilege commands (SetClass, Acquire(Release) MLOCK, and SetStrmID) from the Gather Buffer.

7.1.3.9 Host Controller Class

The Host Controller has its own class that can be executed from the command FIFO. It is comprised of methods to access client registers as well as methods to control channel flow.

7.1.3.9.1 INCR_SYNCPT

All Host Controller modules, including the Host Controller itself, implement the Incr_Syncpt class.

Incr_Syncpt.Cond ([17:10])
Incr_Syncpt.Indx ([9:0])

For the Host Controller, this method immediately increments Syncpt<Indx> irrespective of the condition. The Indx field supports 704 Syncpts.

7.1.3.9.2 WAIT_SYNCPT (Wait on Syncpt)

The command dispatches stalls until the Syncpt counter pointed to by the Indx field reaches the threshold value specified in the Thresh field:

Wait_Syncpt.Indx ([31:24])
Wair_Syncpt.Thresh ([23:0])

The channel waits until the following is true:

(Syncpt<Indx>[15:0] >= Thresh[15:0]), where the ">=" takes wrapping into account.

More specifically, the channel waits until:

(((Syncpt<Indx> – thresh) & (1 << 15)) != 0)

Note: To ensure backward compatibility, only a maximum of 256 Syncpts can be accessed.

7.1.3.9.3 WAIT_SYNCPT_INCR<indx> (Waits until Syncpt increments)

The channel stalls until Syncpt<Indx> is incremented. This wait method is not recommended.

7.1.3.9.4 LOAD_SYNCPT_PAYLOAD_32 ([31:0])

This method loads a 32-bit value into the corresponding channel's Channel_Syncpt_Payload register:

(Channel_Syncpt_Payload[31:0] = Payload[31:0])

7.1.3.9.5 WAIT_SYNCPT_32 <Indx(10)>

This method stalls the current channel until following condition is true:

((SYNCPT<indx>[31:0] - PAYLOAD[31:0]) & 0x80000000 == 0)

Here the Payload value is taken from Channel_Syncpt_Payload of the current channel. This is essentially a wrapping stall until (SYNCPT<indx>[31:0] >= PAYLOAD[31:0]). The index <indx> supports 704 Syncpts.

7.1.3.10 Class-based Programming

The Host Controller supports a programming interface that is based on writing to offsets within a "class" that implements a function, rather than to specific register offsets and formats. By using the class interface, register offsets/formats need not remain fixed so there are no chip specifics in the API. This allows both hardware flexibility and software driver compatibility between SoC generations.

7.1.3.11 Context Switching

Context management of modules included by the Host Controller is completely under software control. When context switching is needed (software knows this), it issues context switching routine to save the current context readying for the switch to the new context.

7.1.3.11.1 Example Software Sequence for StreamID Switch

Channel-based Programming Mode

Consider two different apps (Gaming and Banking), running in a sequential manner in a single OS, which are going to share the same channel and same engine (VIC). Here software should be using EngineIdle Syncpt to ensure engine context is clean before switching to new SMMU context. The Push Buffer (PB) sequences are shown as follows:

A) Gaming App(GuestOs0)

1. Acquire Mlock
2. SetClass(VIC)
3. SetStrmID(global SMMU ID)
4. Setup Syncpt (EngineIdle)
5. SetClass(host1x)
6. Wait for syncpt (EngineIdle) -> To ensure engine is completely idle before SMMU context switch
7. SetClass(VIC)

8. SetStrmID(gaming SMMU context)
9. Gather(gaming app)
10. Setup syncpt(EngineIdle)
11. Setclass(host1x)
12. wait for syncpt (EngineIdle)
13. SetClass(VIC)
14. SetStrmID(invalid smmu context)
15. Release Mlock

B) Banking App(GuestOs1)

1. Acquire Mlock
2. SetClass(vic)
3. SetStrmID(global smmu ID)
4. Setup Syncpt (EngineIdle)
5. SetClass (host1x)
6. Wait for syncpt (EngineIdle)
7. SetClass(vic)
8. SetStrmID(banking smmu context)
9. Gather(banking app)
10. Setup syncpt(EngineIdle)
11. Setclass (host1x)
12. Wait for syncpt (EngineIdle)
13. SetClass(vic)
14. SetSteamID(invalid smmu context)
15. Release Mlock

MMIO-based Programming Mode

For MMIO-based programming clients, specific sequences SMMU context update is used. StreamID changes are synchronized with window updates by software to use the display buffer flip mechanism for StreamID update synchronization without the EngineIdle Syncpt.

7.1.3.12 Specific Programming Guidelines

This secti on provides the programming guidelines for certain use cases:

- For Virtualization during Initialization time, the Hypervisor should program all shareable resources (channel/Syncpt/clients) through configuration registers and table.
- Enable Syncpt protection (SYNCPT_PROT_EN register) by Hypervisor.
- For some use cases, VMID1 (Guest OS0), which is default OS, should be used to access shareable resources.

- For MMIO based programming model, the Hypervisor software has to make sure programming of MMIO Range-VMID table before accessing that client.
- For those clients that use multiple channels per single OS, the Hypervisor should disable MLOCK checks.
- Central RM should enable filtering of kernel commands as mentioned in the **Filtering Out Kernel Commands from the Gather Buffer** section above.

7.1.3.12.1 Programming Sequence for Functional Syncpt Interrupts

Functional Syncpt interrupts are threshold interrupts that are triggered when a Syncpt reaches a pre-programmed value. The targets for these interrupts are CPU0/1, CPU2/3 (camera processor), TrustZone CPU (if the Syncpt is TrustZone secure), VM 1-8 (Each VM has an eight dedicated interrupt line).

This section elaborates on register update sequence during setting and clearing of this interrupt.

1. Threshold value is programmed in the SYNC_INTR_THRESH[i] register in the sync aperture. This value is compared to the Syncpt register.
2. For Syncpt interrupt per VM, Software also programs per Syncpt interrupt destination register HOST1X_THOST_COMMON_VM1_SYNCPT_INTR_DEST_VM_[0..1023] register to select interrupt line per VM to use for a particular Syncpt.
3. If the threshold has been reached, threshold mask registers for all the targets are checked [SYNCPT_THRESH_INTRMASK_CPU0/CPU1/CPU2/CPU3/VMxx]. In case the Syncpt is marked TrustZoneSYNCPT_THRESH_INTRMASK_CPU[i] is checked.
4. SYNCPT_THRESH_INTRSTATUS_xx[i] register is updated based on mask settings and threshold condition. (xx can mean any target CPU0/1/2/3, TrustZoneCPU, VMxx)
5. Based on the target, the appropriate field in the INTRSTATUS_0 register is updated and the interrupt is sent to the target.
6. Once the interrupt is logged, the appropriate bit in the SYNCPT_THRESH_INTRSTATUS_xx[i] should be cleared.
7. A Guest OS can also set the Syncpt interrupt on Syncpt which are associated with other Guest OS using HOST1X_THOST_SYNCPT_SYNCPT_INTR_THRESH_0[0..1023][0..7] register.

7.1.3.12.2 Programming Sequence for Channel Teardown

1. While the channel is in active state, the channel DMA is de-activated by writing to the DMASTOP bit in the per-channel DMACTRL register.
2. The per-channel CMDPROC_STOP register should be written to stop the command processor.
3. The channel teardown register is written 1 to initiate a teardown.
4. After teardown complete, the cmdfifo_stat register is read to make sure it comes to its reset value and stays there for next few cycles.

5. Software resets the client to ensure it is idle.
6. Before reprogramming the same channel to another Push Buffer, the MOD_TEARDOWN register is written to clean context of channel.
7. CMDPROC is written to reset value(0) so the channel can start again. The channel is programmed with a new Push Buffer accessing same client.
8. Data checks are done to ensure it behaves correctly with new programmed data.

7.1.3.12.3 Programming Sequence for Client Reset

1. Enable clamp for client using the HOST1X_THOST_SCR_PROT_COMMON_MOD_CLAMP_EN_SET_0/1 register (via BPMP)
2. Reset client
3. Disable clamp for client using the HOST1X_THOST_SCR_PROT_COMMON_MOD_CLAMP_EN_CLR_0/1 register (via BPMP)

7.1.3.12.4 Programming Sequence for Command FIFO Debugging Register

1. During debug, if Software wants to read the content of channels it can use the Command FIFO debugging register.
2. When CMDFIFO_PEEK_ENA is asserted, the command FIFO reads the data that is stored at address CMDFIFO_PEEK_ADDR. The read data is available in CMDFIFO_PEEK_DATA. This gives visibility into all locations of the RAM.
3. During this time channel related requests to read from the command FIFO RAM are blocked, effectively freezes the contents of whatever is stored in the command FIFO RAM on the read side.
4. The write side logic (PIO or CDMA) is still allowed to add data to the command FIFO (until, per channel, the FIFO is full.) If this is undesired, one could first issue a DMASTOP for all CDMA channels. The command FIFOs are implemented with chasing read/write pointers. Since CMDFIFO_PEEK_ADDR is directly connected to the RAM address input, the address value will not necessarily correspond to the top of a FIFO.
5. The values of read and write pointers into a command FIFO can be found in the CMDFIFO_PEEK_PTRS register. The CMDFIFO_PEEK_CHANNR field selects the channels for the pointers are shown. The begin and end addresses of a command FIFO can be found in the CMDFIFOx_BASE and CMDFIFOx_LIMIT registers.
6. This sequence should not be applied to normal running channel otherwise it will cause corruption in that channel.

7.1.3.12.5 Programming Sequence for Power Up/Down Sequence for Host Controller Clients

Power Down sequence of Host Controller clients:

1. Software asserts the power down request for a corresponding power domain in PMC NV_ADDRESS_MAP_PMC_BASE + PMC_IMPL_PART_<>_PWRDWN_REQ_CONTROL_0=1

2. Software polls the corresponding power down ack bit in PMC NV_ADDRESS_MAP_PMC_BASE + PMC_IMPL_PART_<>_PWRDWN_ACK_STATUS_0=1
3. Software initiates the power down sequence for Host Controller client.

Power Up of Host Controller clients:

1. Software de-asserts the power down request for a given power domain in PMC NV_ADDRESS_MAP_PMC_BASE + PMC_IMPL_PART_<>_PWRDWN_REQ_CONTROL_0=0
2. Software polls the corresponding power down ack bit in PMC NV_ADDRESS_MAP_PMC_BASE + PMC_IMPL_PART_<>_PWRDWN_ACK_STATUS_0=0
3. Software initiates the power up sequence for Host Controller client.
4. Host Controller uses power domain clamps as resets for power domains within a voltage rail.
5. Software de-asserts resets.

7.1.3.12.6 Hypervisor Programming Guideline for Client StreamID (Falcon-based clients)

For Falcon-based client, Hypervisor Software must program the <Engine>STRMID_0_OFFSET_BASE/LIMIT register in host controller to streamID1 offset only (used by Falcon to fetch ucode). This ensures the compromised RMServer does not overwrite streamID0 in client via MMIO path.

Hypervisor Software must program <Engine>_CHANNEL_STRMID_0_OFFSET_BASE/LIMIT register in host controller to streamID0 offset only (used by client engine other than falcon to fetch data). This ensures the compromised OS does not skip updating the streamID0 and use streamID0 set by previous OS.

7.1.3.12.7 Programming Guideline for SetClass(Thost) Command

Here are the programming guidelines for SetClass in case of Thost client:

SetClass(Thost) as a first command:

- In case of SetClass(Thost) as first command in a channel, Software must follow the AcquireMlock(Thost)-SetClass(Thost)-SetStreamID(Thost).... ReleaseMlock(Thost) sequence similar to other clients.
- Here AcquireMlock(Thost), SetStreamID(Thost) and ReleaseMlock(Thost) is treated as NOP by Hardware
- Hardware will be doing virtualition checks for SetStreamID(Thost) command so Software has to program THOST_CHANNNEL_STRMID_0_OFFSET_BASE/LIMIT registers to non-zero values.
- For the SetStreamID(Thost) command, Software has to choose the correct streamID value based on streamID-VM table.

SetClass(Thost) as middle command:

- For AcquireMlock(Engine)-SetClass(Engine)-SetStreamID(Engine).... SetClass(Thost)... ReleaseMlock(Engine) cases there is no restriction on SetClass(Thost) command.

7.2 Camera Subsystem

7.2.1 MIPI Camera Serial Interface (CSI)

Overview

This section describes the SoC CSI implementation, which is based on the MIPI Alliance Specification for CSI 2 (CSI-2), hereafter referred to as the CSI-2 specification. The MIPI Alliance Specification for CSI 2 (CSI-2) is available from the MIPI Alliance to its members at https://www.mipi.org/. This section assumes familiarity with the CSI-2 specification and the Video Input (VI) unit (review the VI section to understand the context in which NVCSI operates).

The NVCSI works with the VI unit to capture an image from a sensor, where NVCSI is a source of pixel data to VI. NVCSI works in streaming mode while VI captures the required frames using a single-shot mode of operation.

All sync point generation for software is handled at VI. The delay between NVCSI and VI is negligible in software terms. NVCSI does not have a direct memory port. Instead it sends the pixel data to memory through VI. The NVCSI stack includes:

- NVCSI CORE - independent host solution, decoupled from VI in terms of software programming and image capture. NVCSI CORE is a Host Controller client that is configured through the Host Controller
- NVCSI Control and Interface Logic (CIL) - focuses on pad brick control, packet boundary extraction, and lane management of the NVCSI stack
- Physical Layer - four dedicated CSI pad bricks

7.2.1.0.8 Standards Support

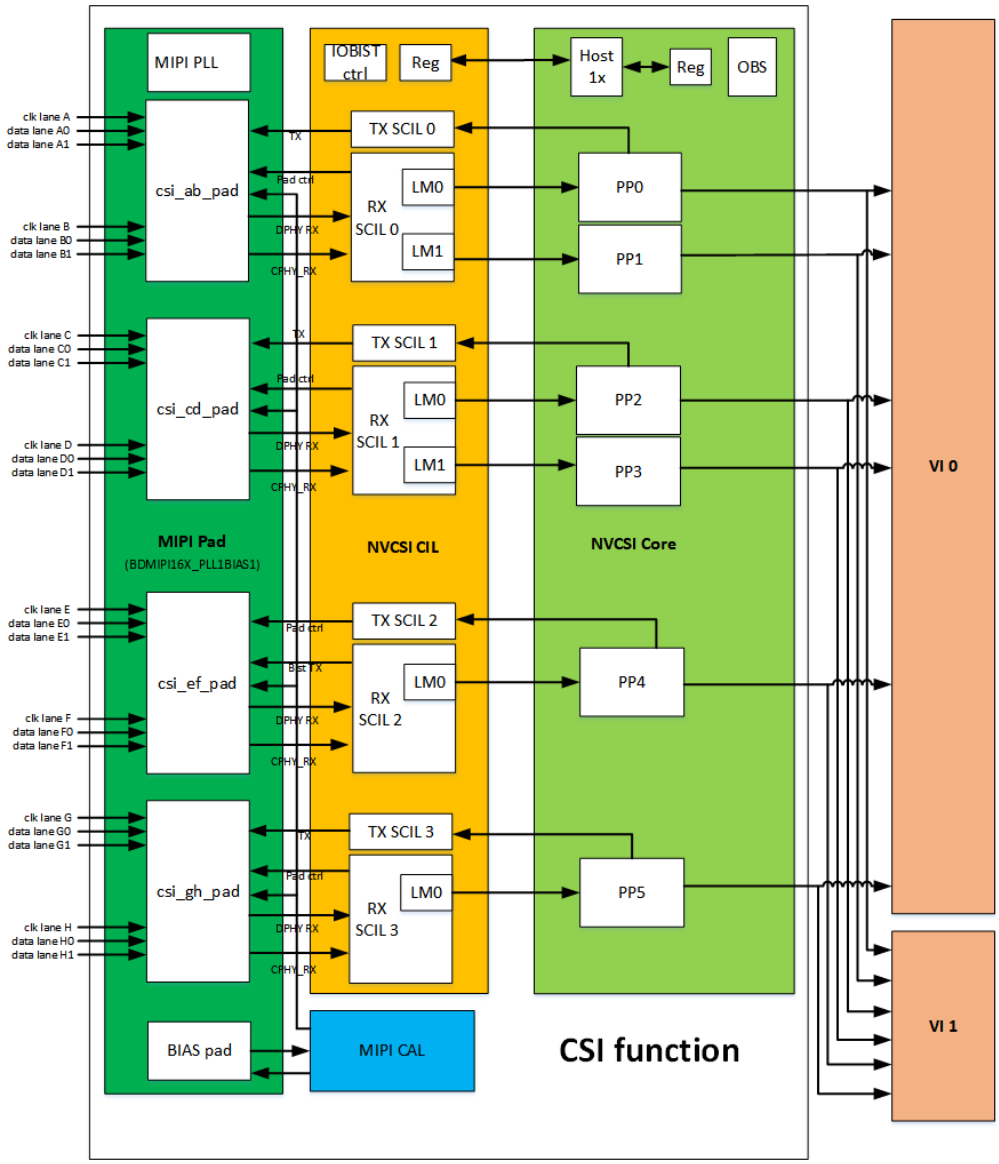
Standard
MIPI Alliance Specification for CSI 2 (CSI-2), version 3.0, MIPI Alliance, Inc.
MIPI Alliance Specification for C-PHY, version 2.0, MIPI Alliance, Inc.
MIPI Alliance Specification for D-PHY, version 2.1, MIPI Alliance, Inc.

Note: The CSI-2 Camera Controller Interface (CCI) is not included, and CSI-2 Predictor2 is not supported.

7.2.1.0.9 Features

- Fifth-generation NVIDIA camera solution (internal versions are NVCSI-2 3.0, VI 6.0, and ISP 6.0).
- Supports both the MIPI D-PHY v2.1 and the MIPI C-PHY v2.0 physical layer options.
  - MIPI D-PHY supports up to 2.5 Gbits/sec per pair
  - MIPI C-PHY supports up to 4.5 Gsym/s (G symbol/sec) per trio
- Based on MIPI CSI-2 v3.0 protocol stack
- Includes six pixel parsers (PP)
- Supports up to 16 virtual channels (VC) per active PP
- Data type interleave support
- Extended data types to include support for raw16/raw20
- Extended DPCM to include support for 12-10-12 coder
- Supports up to eight pixels-per-clock (PPC) throughput on the video input interface
- CSI Tx with TPG

Figure 7.6 External Connectivity Diagram



7.2.1.0.10 NVCSI 2.0 Capabilities

NVCSI provides a combination host that supports both MIPI C-PHY and enhanced MIPI D-PHY (with lane deskew support) physical layer options in four 4-lane, six 2-lane, or six 1-lane configurations; or combinations of these. Each lane can support up to 16 virtual channels (VC) and supports data type interleaving.

- Virtual Channel Interleaving: VCs are defined in the CSI-2 specification and are useful when supporting multiple camera sensors. With the VC capability, a one-pixel parser (PP) can de-interleave up to 16 image streams.

- Data Type Interleaving: In HDR line-by-line mode, the sensor can output long/short exposure lines using the same VC and a different programmable data type (DT).
- Frequency Target: The parallel pixel processing rate, measured in pixels-per-clock (PPC), is increased to give higher throughput and lower clock speeds. To support higher bandwidth without increasing the operating frequency, the host must process multiple pixels in one clock. NVCSI is capable of processing four PPCs when bits-per-pixel (BPP) is greater than 16, and eight PPC when BPP is less than or equal to 16.
- One PP can handle all traffic (embedded data and image data) from one-camera device, including 16 VCs with the streaming mode.

Supported Data Types

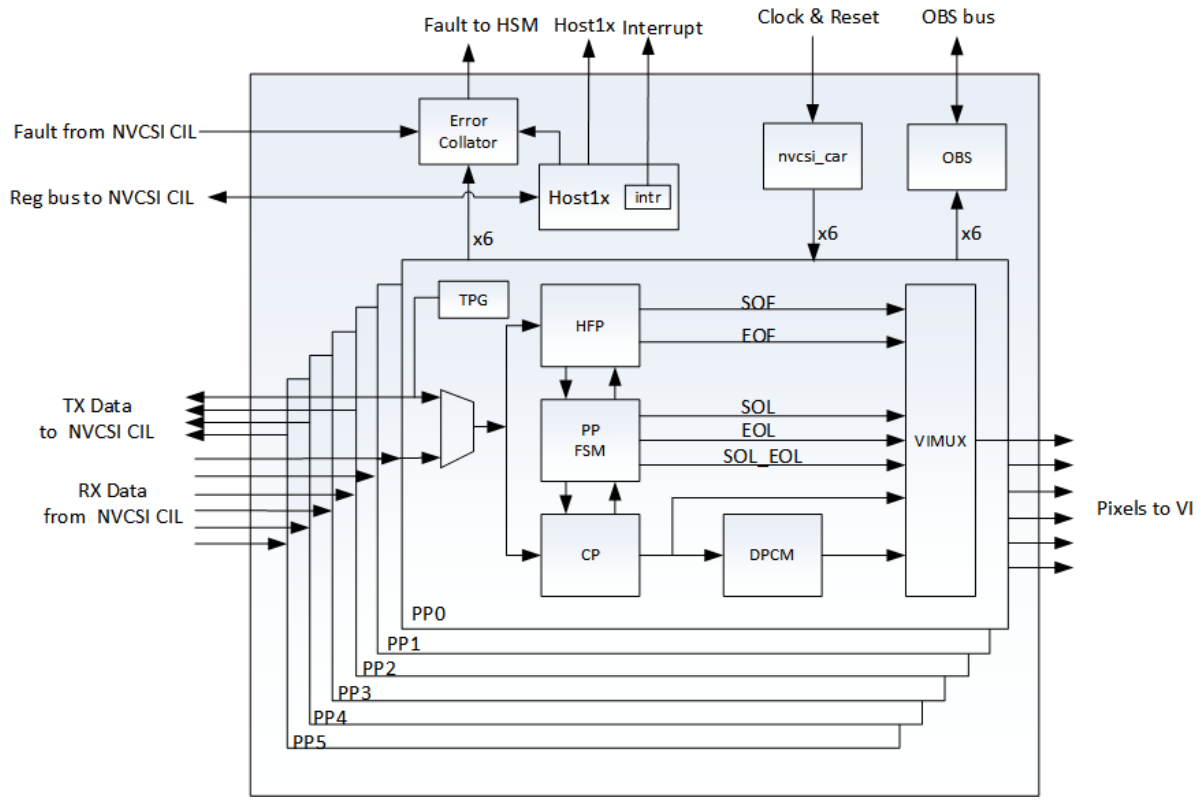
- RAW6/7/8/10/12/14/16/20
- RGB888/666/565/555/444
- YUV420-8bit (Legacy)
- YUV420-8bit
- YUV420-10bit
- YUV422-8bit/10bit

7.2.1.1 NVCSI CORE

7.2.1.1.1 Overview

NVCSI CORE handles the low-level protocol layer and the byte to pixel-unpacking format layer as defined in the CSI-2 specification. NVCSI has six Packet Parser (PP) function units, therefore it can process six individual CSI-2 streams at the same time. The inputs of the NVCSI CORE are six individual streams from NVCSI CIL.

Figure 7.7 NVCSI CORE Block Diagram



7.2.1.1.2 Packet Header Footer Parser

Each PP unit has a Packet Parser forwards the data from the NVCSI CIL to the Header Footer Parser (HFP) directly. The Header Footer Parser:

- Extracts the information in the packet header and performs ECC/CRC checks for the packet header.
- Checks if this packet is a long packet or short packet; if it is long packet, starts the long packet color Parser.
- Checks if this packet is a short packet. For SOF/EOF packets, set appropriate bits to communicate with VI. HFP drops SOL/EOL packets.
- Performs a Payload CRC calculation for the long packet and check that the calculated CRC with the CRC carries that in the packet footer.
- Performs a word count check for the long packet. Byte enable for the long packet Color Parser.

Error-Correcting Code

If there is a correctable error, the Error Correcting Code (ECC) logic corrects it and sends the corrected error to interrupt. This error is per the VC and reported to software and forwarded to VI.

If there is a noncorrectable error, ECC logic reports it to interrupt that this error is not per VC because the packet content is unknown. The packet is dropped and is not sent to VI. If this packet is the FS/FE short packet, VI finds a FS/FE miss error for one VC.

Packet Header CRC

With C-PHY, there are two packet headers sent by the image sensor. In the NVCSI, the redundant packet header is dropped so VI does not get the same short packet twice. If the first-packet header CRC check passes, the second-packet header is masked, but the CRC check is still there. If the second-packet header CRC check fails, it does not affect the payload decode, but it reports a single packet header error status. If the first-packet header CRC check fails, NVCSI continues to check the second-packet header. If the second-packet header CRC also fails, the whole packet is dropped and a packet error is reported. If the second-packet header CRC check passes, the payload decode continues and reports a single packet header error.

Table 7.7 Two-PH Check

Condition	PH1	PH2	Result
1	PASS	PASS	Send one packet to video input if it is a short packet. Continue to decode the payload if it is a long packet. No error is reported.
2	PASS	FAIL	Send one packet to video input if it is a short packet. Continue to decode the payload if it is a long packet. Report single packet header error.
3	FAIL	PASS	Send one packet to video input if it is a short packet. Continue to decode the payload if it is a long packet. Report single packet header error.
4	FAIL	FAIL	Drop the packet. Report both packet header errors.

If the payload size is more than the word count mentioned in PH, the excess bytes are dropped at NVCSI, and no separate error indication is generated. If payload size is less than the word count mentioned in PH, "line-short error" is communicated along with EOF. Commonly, this type of error is accompanied with an ErrCrc.

Color Parser Control (PP FSM).

Any LS/LE packets from the image sensor are dropped in the header parser. The PP FSM generates the LS/LE packets for each long packet (line).

The PP FSM handles the following functions:

- Second PH of C-PHY Masking
- Color Parser enable signal generation to control the Color Parser
- LS/LE generation
- Short line handle
- WDT for a packet

Color Parser

The Color Parser (CP) decodes the packet payload bytes to pixels. The Color Parser accumulates 4/8 pixels size of data based on DTYPE and buffers the extra bits (in column matrix) for the next cycle. It also generates 4/8 bit pixel enables.

DPCM Decompression

When image sensor DPCM compresses the image, it sends the compressed data with a user-defined data type and packs the data to byte as the data type it compressed to. Software overrides the data type which tells the color parser know how to unpack it from byte to pixel and then send to the DPCM decompression unit.

NVCSI supports the Predictor-1 DPCM compression defined in Annex E of CSI-2 specification. Predictor-2 is not supported. NVCSI supports the following DPCM ratios:

- 14–10–14
- 14–8–14
- 12–10–12
- 12–8–12
- 12–7–12
- 12–6–12
- 10–8–10
- 10–7–10
- 10–6–10

VIMUX

The VI output mux collects different information such as SOF, EOF, VC, DTYPE, Frame number, SOL, EOL, Embedded Data, Pixel Data, Error Flags. These are generated from various blocks in NVCSI and then combines and places them on the interface bus to VI.

Host1x Interface

NVCSI is a simplified Host1x client and it only supports the register read/write and the immediate sync point.

7.2.1.1.3 Image Data Types

Table 7.8 Supported Data Types

Image Data (DTYPE)	Bits per pixel (BPP)	Input Rate in BPC (bits per clock)	Output Rate in PPC (pixel per clock)
RAW6	6	64	8
RAW7	7	64	8

Image Data (DTYPE)	Bits per pixel (BPP)	Input Rate in BPC (bits per clock)	Output Rate in PPC (pixel per clock)
RAW8	8	64	8
RAW10	10	64	8
RAW12	12	64	8
RAW14	14	64	8
RAW16	16	64	8
RAW20	20	64	4
RAW (10-6-10)	6	64	8
RAW (10-7-10)	7	64	8
RAW (10-8-10)	8	64	8
RAW (12-6-12)	6	64	8
RAW (12-7-12)	7	64	8
RAW (12-8-12)	8	64	8
RAW (12-10-12)	10	64	8
RAW (14-10-14)	10	64	8
RAW (14-8-14)	8	64	8
RGB888	24	64	4
RGB666	18	64	4
RGB565	16	64	8
RGB555	16	64	8
RGB444	16	64	8
YUV420-8bit (Legacy)	12	64	8
YUV420-8bit	12	64	8
YUV420-10bit	20	64	4
YUV422-8bit	16	64	8
YUV422-10bit	20	64	4

Data Type Override

Data Type (DT) override means overriding the Data Type field, which is part of the incoming packet header inside the NVCSI for processing purposes only. NVCSI still sends the original DT (from incoming PH) to VI. Override is only applicable for the internal processing of pixels in NVCSI.

Supporting Compressed RAW Formats

The CSI-2 specification does not allocate standard DT for compressed RAW formats (X-Y-Z), it comes under USER defined Data Types. Therefore, different sensor vendors may come up with different DT values and the NVCSI needs to unpack the pixels based on RAWY (where Y = 6, 7, 8, and 10). Incoming DT needs to override any of the RAWY values so that NVCSI can treat them like regular RAW formats without adding extra logic. The extracted pixels are fed to the DPCM decompression unit, if decompression is enabled.

Supporting HDR Line by Line Mode

Sensors can give out Long/Short exposure lines using the same Virtual Channel (VC) and different programmable DTs. These programmable DTs are mapped to standard Data Types defined in the CSI-2 specification, so that NVCSI can understand the format while unpacking the packet for pixels.

7.2.1.1.4 Clock and Reset

NVCSI CORE receives a clock from PLLNVCSI. All parts of the PP share the same NVCSI clock source. CAR logic provides separate clock enable control to the NVCSI CORE. NVCSI engine-level clock gating is implemented inside NVCSI CORE and NVCSI CIL.

Clock Gating

The NVCSI CORE has two clock sources. The host controller clock does not have any SLCG. The nvcsi_clk is gated with PP enable for each PP in the NVCSI CORE. The register NVCSI_STREAM_[0-5]_SLCG_CTRL_0 controls the SLCG override.

7.2.1.1.5 Line/Frame CRC Check

CSI protocol natively supports the line CRC check for all types included in the embedded line packets and pixel line packets. NVCSI follows the CSI specification v2.0 to implement the standard packet level CRC check. NVCSI only processes the packet level (line level) information. It does not have any view of the frame. VI or software handles the Frame CRC. When line CRC error is detected, a fault interrupt line asserts to the hardware safety manager (HSM).

The line CRC error can be set to correctable fault or uncorrectable fault by safety software. The line CRC error means that there is one or more pixels incorrect in a line. Hardware reset is not required for this error if it is just a transient error in NVCSI and the next frame is correct from sensor, it can recover automatically. Software can set a threshold of this CRC error. If the CRC error continues to occur on multiple lines, this may indicate a problem in the receiver or in the transmission lines and a hardware reset of NVCSI may be required. This error status is also sent to VI, and the safety software can get error status from either the NVCSI fault or VI notifies.

If all the lines in one frame pass the line CRC check, it means that the frame matches the CRC. NVCSI reports the line CRC check result, but it does not know if the expected lines are received from the sensor. VI has the capability to check the frame integrity and NVCSI/VI cowork to check the frame CRC.

Performance

The system level NVCSI maximum bandwidth is listed in the following table.

Table 7.9 System Level Peak Bandwidth

PHY Mode	Target	VLV	LV	SV
D-PHY	BW/Lane	2.5 Gbps	2.5 Gbps	2.5 Gbps
	Total BW	40 Gbps	40 Gbps	40 Gbps
	Byte clock	312.5 MHz	312.5 MHz	312.5 MHz
	Core clock	156 MHz	156 MHz	156 MHz *1
C-PHY	BW/Trio	1.68 Gsps	3.0 Gsps	4.5 Gsps
	Total BW	61.4 Gbps	109.7 Gbps	164.6 Gbps
	Byte clock	240 MHz	428.6 MHz	642.9 MHz
	Core clock	240 MHz	428.6 MHz	642.9 MHz *2

*1 2.5 GHz x 4 / 64 ~= 156 MHz
*2 4.5 GHz x 16 / 7 x 4 / 64 ~= 642.9 MHz

The NVCSI/VI interface throughput is a maximum of eight PPC, although for the pixel format that is less than 16 Bpp.

For the input stream of pixel formats such as RAW6 and RAW7, the peak PPC value at the input side is 10.7 and 9.14. Therefore, for the RAW6/RAW7 format, NVCSI cannot support the full bandwidth. As the RAW6/RAW7 maxim accepts one packet data every two cycles, the max bandwidth at the core side is 64*Fcore/2.

Table 7.10 C-PHY RAW6/7 Maximum Bandwidth

Lanes	Rate/Trio	BW/Trio	Total BW	Lane Clock	NVCSI Clock
1	4.5 Gsps	10.28 Gbps	10.28 Gbps	642.9 MHz	321.25 MHz*1
2	4.5 Gsps	10.28 Gbps	20.57 Gbps	642.9 MHz	642.9 MHz*2
3	3 Gsps*3	6.86 Gbps	20.57 Gbps	428.6 MHz	642.9 MHz

Lanes	Rate/Trio	BW/Trio	Total BW	Lane Clock	NVCSI Clock
4	2.25 Gsps*4	5.14 Gbps	20.57 Gbps	321.5 MHz	642.9 MHz

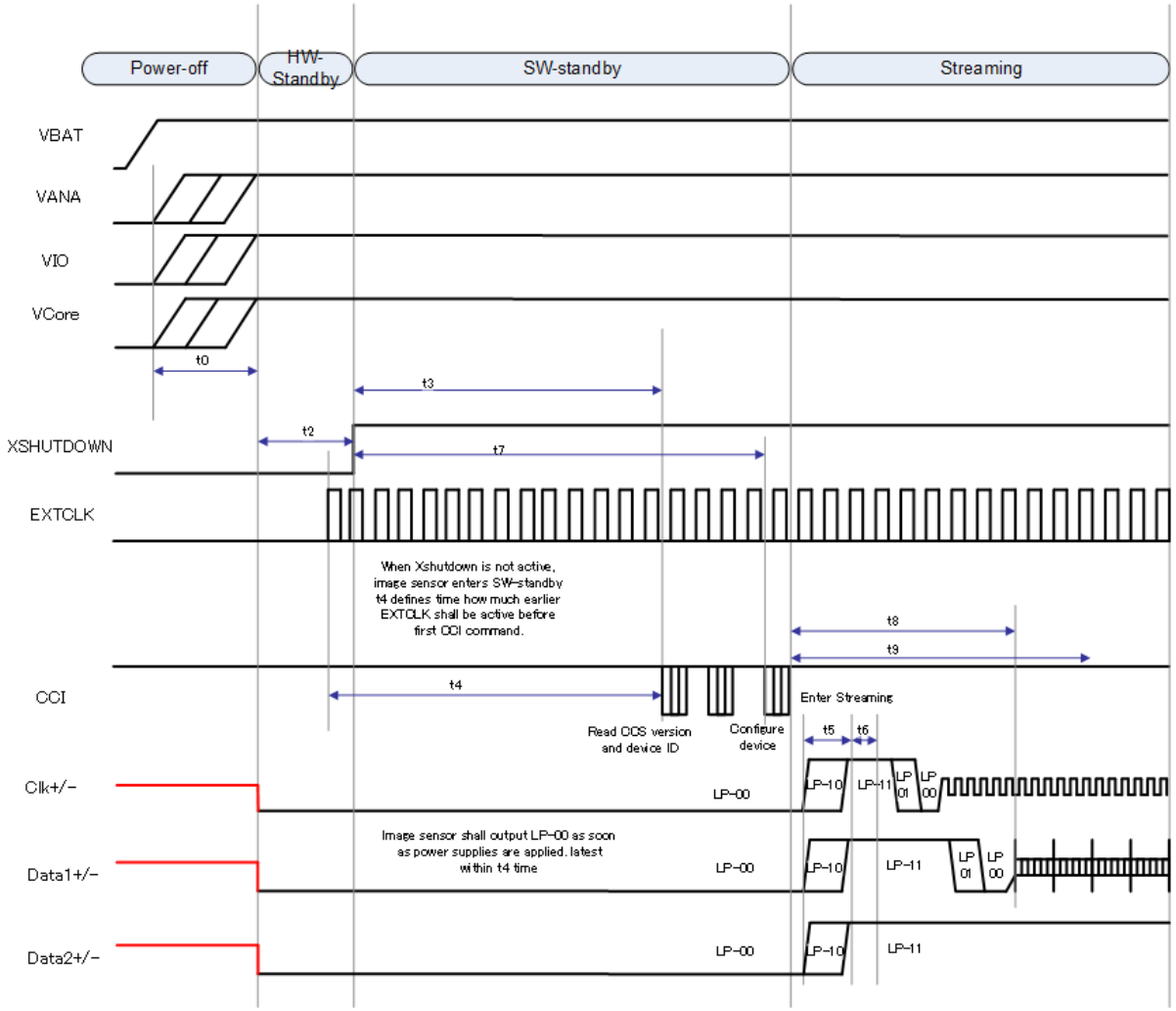
*1: 10.28G/32 ~= 321.25 M
*2: 20.57G/32 ~= 642.9 M
*3: 20.57G*7/16/3 ~= 3 G
*4: 20.57G*7/16/4 ~= 2.25 G

7.2.1.1.6 Software Programming Sequence

Programming Model

NVCSI is a MIPI CSI-2 Receiver and it expects the Transmitting device to follow the sequence defined in MIPI CCS spec version 1.0 (MIPI Specification for Camera Command Set). The initial/ stop/mode change sequences are shown below.

Figure 7.8 Initial Sequence (DPHY example)



Initial Sequence

Software should follow the steps below when initializing NVCSI for image captures:

1. Enable the NVCSI clock and nvcsi_lpclk. Remove the reset of the NVCSI, but keep the PP and CIL in reset state.
2. Power up the CSI pad.
3. Initial sensor to enter soft standby state with steps:
   a. PMIC programming to enable power (if required)
   b. Toggle XSHUTDOWN
   c. Wait several ms per sensor requirement.
   d. Enable XCLK.
   e. Wait several ms per sensor requirement.

   f. Toggle XCLR.
   g. Complete other operations if required by sensor requirement.
4. Set up the VI global setting and CSIMUX.
5. Prepare task descriptor for frame-based setting and submit to Falcon.
6. Program the CSI CIL configuration register.
7. Program the CSI PP configuration registers.
8. Remove reset of the NVCSI PP.
9. Remove reset of NVCSI CIL lanes.
10. Use the I2C to communicate to sensor:
    a. CSI interface configuration and initialize.
    b. Mode registers configuration.
    c. Activate from the soft standby mode to start the streaming.
11. Submit more task descriptors if required.

Stop Sequence

Software should follow the steps below when it needs to stop the sensor.

1. If required, follow these steps to make the boundary clean to guarantee the frame integrity.
   a. Software operation(s) to enable the frame end notification.
   b. Falcon gets a frame start notification.
   c. Interrupt to Software
2. Put the NVCSI PP and CIL in reset state.
3. Power down the CSI pad.
4. Use I2C to communicate to the sensor to enter soft standby.
5. Hardware standby for the sensor:
   a. Toggle XCLR.
   b. Wait several ms per sensor requirement.
   c. Toggle XSHUTDOWN.
   d. Wait several ms per sensor requirement.
   e. Stop INCLK.
   f. PMIC to stop the power of sensor (if required).

Mode Change without Standby Entry (not recommended)

Software should follow the steps below when it needs to do the mode change without putting the sensor in standby mode. This mode change sequence is not defined in CCS specification and does not support the lane number change.

1. Software operation(s) to enable the frame start notification.
2. Falcon gets a frame start notification.
3. Interrupt to Software.

4. Use I2C to communicate to the sensor a new mode setting.
5. Program the NVCSI PP registers (if required). NVCSI CIL register is not recommended to change.
6. Step 4 and 5 should be completed in one frame time.
7. New mode setting active from next frame.

Mode Change with Standby Entry

Software should follow the steps below when it needs to do the mode change with putting the sensor in standby mode.

1. If required, follow these steps to make the boundary clean to guarantee the frame integrity.
   a. Software operation(s) to enable the frame end notification.
   b. Falcon gets a frame start notification.
   c. Interrupt to Software
2. Put the NVCSI PP and CIL in reset state.
3. Use I2C to communicate to the sensor to enter soft standby.
4. The MIPI link, which is currently active, or will be active after standby, is not allowed to be Hi-Z during the standby phase.
5. Program the NVCSI PP registers and NVCSI CIL registers.
6. Soft reset the VI channel.
7. Remove the reset of the NVCSI PP and NVCSI CIL.
8. Use the I2C to communicate to the sensor for new mode settings and exit the soft standby mode.
9. Start normal image capture operation in VI.

7.2.1.2 NVCSI SCIL

The SCIL part of NVCSI contains all the NVCSI PHY interfacing and controls in one unit. This unifies data received from the D-PHY and C-PHY combination PHY (referred to as the combo brick) into the FIFO output of SCIL, so that the NVCSI CORE can work on data from the FIFO with less dependency on brick and PHY-related artifacts. NVCSI SCIL handles this brick, which includes the analog parts of MIPI D-PHY, v2.1, and MIPI C-PHY, v1.2.

The SCIL block includes the following functionality:

- Deskew calibration of the brick for D-PHY
- Polarity swapping and lane/trio swapping
- Reacting to and controlling brick for LP/HS and HS/LP transitions
- Handling and decoding of LP data received from the camera sensor
- Enabling/disabling the high-speed datapath of the brick
- Controlling various signals of the NVCSI combo brick to power down the brick in Ultra Low Power State (ULPS)

- Providing configuration registers for electrical parameter setting
- Establishing byte/word boundary
- Asserting appropriate error signal detected during data reception to NVCSI CORE block
- Byte alignment between multiple data lanes/trios
- Aligning data from brick and feeding into the FIFO between SCIL to NVCSI CORE
- Symbol translation, decoding, and demapping of 7-symbol-to-16-bit
- C-PHY edge delay calibration for clock recovery

The NVCSI SCIL (also known as SCIL or Super CIL) module controls the individual lanes of an NVCSI combo brick (or brick). SCIL aggregates data from individual lanes while supporting following brick configuration modes on two instances (AB and CD) of dedicated brick:

- D-PHY: 1x 4 lanes, 2x 2 lanes, 2x 1 lane, 1x 1 lane, 1x 2 lanes, 1x 1 lane + 1x 2 lanes
- C-PHY: 1x 4 trios, 1x 3 trios + 1x 1 trio, 2x 2 trios, 2x 1 trio, 1x 1 trio, 1x 3 trios, 1x 2 trios + 1x 1 trio, 1x 2 trios

AB and CD instances support two streams.

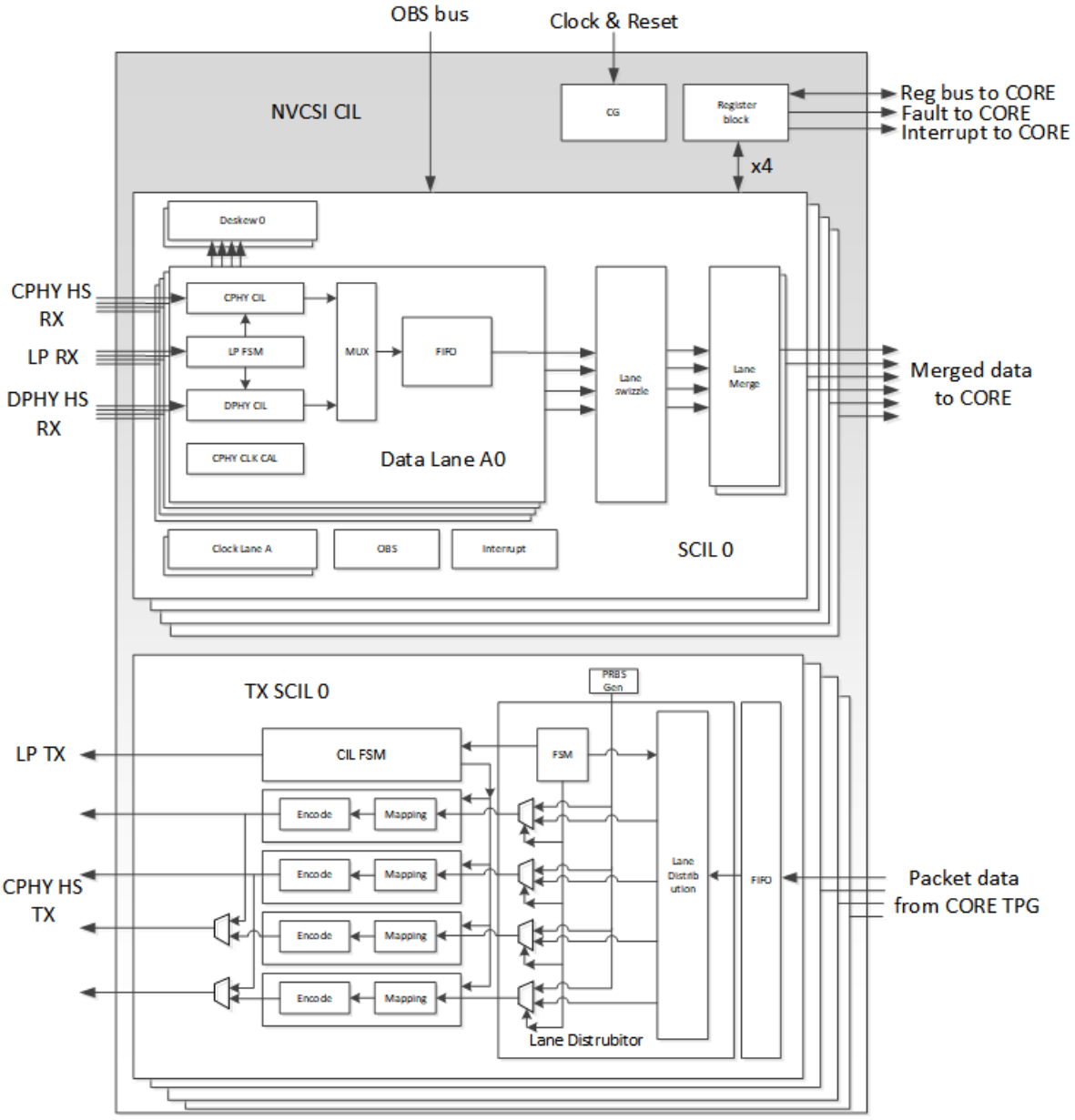
Two other instances (EF and GH) of dedicate brick supports subset of preceding configurations as listed:

- D-PHY: 1x 4 lanes, 1x 2 lanes, 1x 1 lane
  - The following D-PHY configurations are not supported: 2x 2 lanes, 2x 1 lane, 1x 1 lane + 1x 2 lanes
- C-PHY: 1x 4 trios, 1x 3 trios, 1x 2 trios, 1x 1 trio
  - The following C-PHY configurations are not supported: 1x 3 trios + 1x 1 trio, 2x 2 trios, 2x 1 trio, 1x 2 trios + 1x 1 trio

EF and GH instances support only one stream.

This SoC implements four bricks each supporting a maximum of four lanes with the preceding configurations. The data aggregated from physical lanes is pushed into an asynchronous FIFO, which interfaces to the NVCSI CORE logic block. Along with data, the SCIL block also asserts appropriate error signals to the NVCSI CORE.

Figure 7.9 NVCSI SCIL Block Diagram



One SCIL interfaces to one brick; this SoC implements four SCIL modules to control and receive data from four bricks. This configuration supports up to six cameras without an external aggregator. The data received from the brick generated by the camera sensors is aligned and provided to the NVCSI pixel parsers. MIPI D-PHY v2.1 (but supports up to 2.5 Gbps/lane) and MIPI C-PHY v1.2. SCIL implements the following layers as defined in CSI-2 specification:

- Low-Level Protocol: The low-level protocol (LLP) includes the means of establishing bit-level and byte-level synchronization for serial data transferred between the Start of Transmission (SoT) and the End of Transmission (EoT) events and for passing data to the NVCSI CORE. The

minimum data granularity of the LLP is one byte for D-PHY and two bytes for C-PHY. The LLP also includes assignment of bit-value interpretation within the byte, i.e., the "Endian" assignment.

- Lane Management: The CSI-2 interface is lane-scalable for increased performance. SCIL supports up to four data lanes per brick in the D-PHY configuration and up to four trios in the C-PHY configuration. SCIL collects bytes from the lanes/trios and merges them together into a recombined data stream that restores the original stream sequence from the camera sensor. For the C-PHY physical layer option, this layer exclusively collects byte pairs (i.e., 16-bits) from the trios.

In each supported PHY configuration, per data lane or trio, the brick provides high-speed data in the HS mode synchronous to the slow_clock/byte_clock, which is derived by the brick either from the recovered clock from input data (in C-PHY configuration) or from the clock received separately on dedicated pins (in D-PHY configuration). The brick provides this slow_clock/byte_clock to the SCIL to sample the HS mode data. In low power (LP) mode, per each lane or trio, the brick acts as a buffer, and SCIL samples the LP signaling to generate LP data. By default, at power-on, each lane or trio of the brick is set to LP mode. When predetermined LP signaling is detected, the SCIL transitions the brick from LP to HS mode. Image data from the camera sensor is received in HS mode. After completion of one packet transfer with appropriate handshake using LP signaling, the SCIL disables the HS path of the lane/trio and keeps the lane/trio in LP mode. SCIL waits for the next HS mode transition or ultra-low power state (ULPS) entry.

The following are main features of SCIL:

- Control of the brick HS path and transitions from LP mode to HS mode
- Power management of the brick is either software controlled (through register programming) or camera sensor initiated
- Polarity swapping in D-PHY and C-PHY configurations
- Data lane swapping in D-PHY and trio swapping in C-PHY
- Lane merging
- In D-PHY configuration:
  - Deskew calibration of the brick data lanes
  - Receives one byte in LPDT mode and registers it for software use
- Detection of SoT and EoT
- Word alignment through SYNC word search (allows one-bit error)
- Identification of deskew burst and normal HS data burst with appropriate actions
- In C-PHY configuration:
  - Termination of the recovered clock before lane swapping and merging
  - Translation of wire state information to symbols, with de-mapping functionality from seven symbols to 16-bit data
  - Detection of preamble and post amble, with programmable preamble sequence search
- Scalable bytes/words (from different lanes/trios) merging functionality to make unique data width toward the NVCSI CORE

- Watchdog timer to guard against no EoT detection
- Error detection and reporting on received data and mode transitions

Note that the SCIL in D-PHY configuration does not implement the bidirectional data turnaround feature of D-PHY, therefore no link turnaround procedure is supported.

For the D-PHY configuration, polarity swapping is allowed only on data lanes. There is no polarity swapping on clock lanes as listed in the following table. Furthermore, data lane swizzling is limited to data lanes only. Lane swapping is allowed among data lanes only. Clock lanes cannot be configured as data lanes, and vice versa.

Table 7.11 D-PHY and C-PHY Pin Sharing and Polarity Swapping

Pin Name	D-PHY		C-PHY	
	Default Lane Number	**Polarity Swapping**	**Default Trio Name**	**Polarity Swapping**
CSI_A_D0_P / CSI_TRIO0_A	Data Lane ‘d0_a’	Allowed between these pins only	trio_0	Allowed among these pins only
CSI_A_D0_N / CSI_TRIO0_B				
CSI_A_CLK_P / CSI_TRIO0_C	Clock Lane ‘a’	Allowed on a clock lane		
CSI_A_CLK_N / CSI_TRIO1_C			trio_1	Allowed among these pins only
CSI_A_D1_P / CSI_TRIO1_A	Data Lane ‘d1_a’	Allowed between these pins only		
CSI_A_D1_N / CSI_TRIO1_B				
CSI_B_D0_P / CSI_TRIO2_A	Data Lane ‘d0_b’	Allowed between these pins only	trio_2	Allowed among these pins only
CSI_B_D0_N / CSI_TRIO2_B				
CSI_B_CLK_P / CSI_TRIO2_C	Clock Lane ‘b’	Not allowed on a clock lane		
CSI_B_CLK_N / CSI_TRIO3_C			trio_3	Allowed among these pins only
CSI_B_D1_P / CSI_TRIO3_A	Data Lane ‘d1_b’	Allowed between these pins only		
CSI_B_D1_N / CSI_TRIO3_B				

Polarity swapping is allowed among any trio pins for C-PHY configuration. Trio (or lane) swapping is allowed among all trios, for example, any trio can become trio_0 (or tr0).

Each SCIL has four CIL blocks corresponding to four data lanes/trios.

7.2.1.2.1 D-PHY

Low-Power Receiver

During reset, the CIL disables the HS-RX path and enables the LP-RX path only. This makes the CIL watch the LP-RX behavior that determines the intended operation mode of the brick. The remote transmitter in the camera sensor controls the CIL.

Table 7.12 LP Line States

State Code	Line Voltage Levels		Low Power	
	Dp-line	Dn-line	Control mode	Escape mode
LP-00	LP low	LP low	Bridge	Space
LP-01	LP low	LP high	HS-Rqst	Mark-0
LP10	LP high	LP low	LP-Rqst	Mark-1
LP-11	LP high	LP high	Stop	N/A[1]

[1] If LP-11 occurs during Escape mode, the lane returns to the Stop state (Control mode LP-11).

After detecting the HS-Rqst, CIL enables HS-RX. HS mode entry is performed when CIL receives the following LP sequence of LP-11, LP-01, and LP-00. The receiver is in HS mode until CIL detects LP-11 (Stop state). CIL enters the Escape mode using a request within Control mode. CIL exits Escape mode when the Stop state is detected. If nothing is communicated on the lane, the data lane state is the Stop state and the mode is Control mode. CIL supports the following events starting from the Stop state:

- High-speed data transmission request (LP-11, LP-01, LP-00)
- Escape mode request (LP-11, LP-10, LP-00, LP-01, LP-00).

CIL implements low-power data transmission and ULPS detection Escape modes. CIL ignores other Escape modes and triggers listed in the table below. For better visibility to software, CIL provides the received Escape mode pattern to the software through a register along with the appropriate interrupt assertion. Software may use this information to perform any relevant action or may trigger an error or ignore it based on the camera sensor requirement.

Table 7.13 D-PHY Escape Entry Codes

Escape Mode Action	Command Type	Entry Command Pattern (first bit transmitted to last bit transmitted)
Low-power data transmission (LPDT)	Mode	11100001
Ultra-low power state (ULPS)	Mode	00011110

After the Escape entry command for the LPDT is received, the data received in this mode is transferred to the LPDT data register and alerts the software, which processes it further.

After the entry to Escape mode, if CIL receives a ULPS mode command it configures the brick to low-power mode. It gates other logic to save power. CIL starts a ULPS mode exit when it detects a Mark-1 state for T_{WAKEUP} (= 1 ms minimum) followed by Stop state.

Skew Calibration Function

In the D-PHY configuration, the CIL block implements polarity swapping and, in normal burst, searches for a sync byte and aligns incoming bits to meaningful byte boundaries. The value HS sync sequence distinguishes the bursts. After entry to HS mode, when CIL detects an HS Sync sequence to be FFh, it moves to the skew calibration state. In skew calibration state, CIL scans through all the values in deskew range on the HS path of brick to provide skew calibration result to software to determine optimal skew values. Software determines the optimal deskew value to minimize skew between the clock lane and the data lane. The skew-adjustment bit size is dependent on the brick design and is currently 6 bit in size.

HS Mode Operation

High-speed data transmission occurs in bursts, and each burst is extended with a leader and trailer sequence to aid receiver synchronization. The HS burst begins and ends with a Stop state.

Once the LP-01 -> LP-00 transition is detected, CIL enables receiver termination. During the leader sequence sync period, if CIL receives "011101b" possibly with a one-bit error (i.e., x11101, 0x1101, 01x101, 011x01, 0111x1, 01110x), then the burst should be considered as a normal mode burst. It should use the sync sequence to align the byte boundary. The data received after the sync symbol is forwarded to the 64-bit lane-data aggregator logic. This aggregator logic controls how the data to be written in asynchronous FIFO following it.

The lane merger block reads data from all active lanes (i.e., the lanes receiving data in the current configuration) asynchronous FIFO and combines it into 64-bit payload data or 32-bit packet header data to the NVCSI CORE logic in all the lane configurations. CIL should leave the HS burst state once it detects the EoT procedure, that is, detection of the Stop state on LP-RX. While exiting the HS burst state, the CIL should disable receiver termination and discard bits received from the brick after that.

Polarity Swapping

Polarity swapping is supported on the input-data lane pins for LS and HS modes to aid platform design flexibility. Polarity swapping for D-PHY is enabled by setting the POLARITY_SWIZZLE_DPHY[0/1]_A bits of the NVCSI_CIL_A_POLARITY_SWIZZLE_CTRL_0 register for D-PHY Lane-a0/a1 lanes of a brick. For example, when bit 0 of this register is set, polarity-swapping operation on D-PHY Lane-a0 is performed. Similarly, POLARITY_SWIZZLE_DPHY[0/1]_B bits of NVCSI_CIL_B_POLARITY_SWIZZLE_CTRL_0 register control the polarity operation for Lane-b0 and Lane-b1, respectively.

7.2.1.2.2 C-PHY

Low-Power Receiver

D-PHY and C-PHY CIL logic share the same low-power receiver functionality.

HS Mode Receiver Operation

C-PHY provides a synchronous connection between the host and the device. It consists of one or more sets of three wires (referred to here as a lane or trio) for high-speed data communication, which appears in bursts with an arbitrary number of payload data bytes. A minimum of three wires is required in the C-PHY configuration for data communication. In HS mode, each lane is terminated on both sides and driven by a low-swing three-phase signal.

HS Burst

High-speed data transmission occurs in bursts and each burst is extended with a preamble and postamble sequence to aid receiver synchronization. Normal mode burst is the same as a D-PHY burst, but SOT is replaced with a preamble and EOT is replaced with a postamble. The sync word is replaced with a symbol sequence of 3, 4, 4, 4, 4, 4, 3. A preamble consists of three parts: PreBegin, programmable Preamble, and PreEnd. PreBegin and PreEnd are composed of a sequence of symbol "3" with lengths of 7 to 448 symbols and 7 symbols, respectively.

The HS burst starts from, and ends with, a Stop state. Once LP-01 -> LP-00 transition is detected, CIL enables receiver termination. After PreEnd and during the sync period, if CIL receives a "4, 4, 4, 4, 4, 3" symbol sequence, then it uses a sync sequence to align the word/symbol boundary. The data received after the sync word is forwarded to the 64-bit lane-data aggregator logic.

This aggregator logic controls how the data is to be written in asynchronous FIFO following it. The lane merger block reads data from all active lane's (i.e., the lanes receiving data in the current configuration) asynchronous FIFO and combines it into a 64-bit payload data or 32-bit packet header data to the NVCSI CORE logic in all the lane configurations. CIL leaves the HS burst state once it detects the post/EoT procedure, that is, detection of the Stop state on LP-RX. While exiting the HS burst state, CIL disables receiver termination and discards bits received from the brick after that.

Polarity swapping is supported on the input-data lane pins for LS and HS modes to aid platform design flexibility. Register fields POLARITY_SWIZZLE_CPHY[0/1]_A of register NVCSI_CIL_A_POLARITY_SWIZZLE_CTRL_0 control the polarity of trio[0/1] respectively. Similarly, register field POLARITY_SWIZZLE_CPHY[0/1]_B of register NVCSI_CIL_B_POLARITY_SWIZZLE_CTRL_0 control the polarity of trio[2/3] respectively.

7.2.1.3 NVCSI Registers

7.2.1.3.1 NVCSI Stream0 Registers

NVCSI_STREAM_0_SW_RESET_CTRL_0

Offset: 0x4000
Byte Offset: 0x10000
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DISABLE	**cfg_swreset**: Reset the pixel parser. **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_0_SLCG_CTRL_0
Offset: 0x4001
Byte Offset: 0x10004
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DISABLE	**cfg_slcg_override**: Enable the SLCG override for the pixel parser. **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_0_PP_PHY_CTRL_0

Offset: 0x4002
Byte Offset: 0x10008
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DPHY	**cfg_phy_mode**: **0** = DPHY **1** = CPHY

NVCSI_STREAM_0_PP_EN_CTRL_0

Offset: 0x4003
Byte Offset: 0x1000c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	CIL	**cfg_src**: The pixel source of the pixel parser **0** = CIL **1** = TPG
0	DISABLE	**cfg_pp_en**: Pixel Parser streaming enable **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_0_VC0_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x4004
Byte Offset: 0x10010
Read/Write: R/W

Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc0_dt_nooverride_0**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC0_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x4005
Byte Offset: 0x10014
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc0_dt_nooverride_1**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC0_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x4006
Byte Offset: 0x10018
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc0_dt_nooverride_2**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC0_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x4007
Byte Offset: 0x1001c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc0_dt_nooverride_3**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC0_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x4008
Byte Offset: 0x10020
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc0_dt_nooverride_4**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC0_DT_OVERRIDE_0

Offset: 0x4009
Byte Offset: 0x10024
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc0_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc0_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc0_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_0_VC1_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x400a
Byte Offset: 0x10028
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc1_dt_nooverride_0**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC1_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x400b
Byte Offset: 0x1002c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc1_dt_nooverride_1**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC1_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x400c
Byte Offset: 0x10030
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc1_dt_nooverride_2**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC1_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x400d
Byte Offset: 0x10034
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc1_dt_nooverride_3**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC1_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x400e
Byte Offset: 0x10038
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc1_dt_nooverride_4**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC1_DT_OVERRIDE_0

Offset: 0x400f
Byte Offset: 0x1003c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc1_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc1_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc1_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_0_VC2_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x4010
Byte Offset: 0x10040
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc2_dt_nooverride_0**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC2_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x4011
Byte Offset: 0x10044
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc2_dt_nooverride_1**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC2_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x4012
Byte Offset: 0x10048
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc2_dt_nooverride_2**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC2_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x4013
Byte Offset: 0x1004c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc2_dt_nooverride_3**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC2_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x4014
Byte Offset: 0x10050
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc2_dt_nooverride_4**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC2_DT_OVERRIDE_0

Offset: 0x4015
Byte Offset: 0x10054
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc2_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc2_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc2_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_0_VC3_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x4016
Byte Offset: 0x10058
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc3_dt_nooverride_0**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC3_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x4017
Byte Offset: 0x1005c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc3_dt_nooverride_1**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC3_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x4018
Byte Offset: 0x10060
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc3_dt_nooverride_2**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC3_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x4019
Byte Offset: 0x10064
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc3_dt_nooverride_3**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC3_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x401a
Byte Offset: 0x10068
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc3_dt_nooverride_4**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC3_DT_OVERRIDE_0

Offset: 0x401b
Byte Offset: 0x1006c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc3_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc3_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc3_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_0_VC4_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x401c
Byte Offset: 0x10070
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc4_dt_nooverride_0**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC4_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x401d
Byte Offset: 0x10074
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc4_dt_nooverride_1**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC4_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x401e
Byte Offset: 0x10078
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc4_dt_nooverride_2**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC4_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x401f
Byte Offset: 0x1007c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc4_dt_nooverride_3**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC4_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x4020
Byte Offset: 0x10080
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc4_dt_nooverride_4**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC4_DT_OVERRIDE_0

Offset: 0x4021
Byte Offset: 0x10084
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc4_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc4_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc4_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_0_VC5_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x4022
Byte Offset: 0x10088
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc5_dt_nooverride_0**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC5_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x4023
Byte Offset: 0x1008c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc5_dt_nooverride_1**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC5_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x4024
Byte Offset: 0x10090
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc5_dt_nooverride_2**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC5_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x4025
Byte Offset: 0x10094
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc5_dt_nooverride_3**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC5_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x4026
Byte Offset: 0x10098
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc5_dt_nooverride_4**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC5_DT_OVERRIDE_0

Offset: 0x4027
Byte Offset: 0x1009c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc5_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc5_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc5_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_0_VC6_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x4028
Byte Offset: 0x100a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc6_dt_nooverride_0**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC6_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x4029
Byte Offset: 0x100a4
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc6_dt_nooverride_1**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC6_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x402a
Byte Offset: 0x100a8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc6_dt_nooverride_2**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC6_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x402b
Byte Offset: 0x100ac
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc6_dt_nooverride_3**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC6_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x402c
Byte Offset: 0x100b0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc6_dt_nooverride_4**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC6_DT_OVERRIDE_0

Offset: 0x402d
Byte Offset: 0x100b4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc6_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc6_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc6_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_0_VC7_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x402e
Byte Offset: 0x100b8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc7_dt_nooverride_0**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC7_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x402f
Byte Offset: 0x100bc
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc7_dt_nooverride_1**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC7_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x4030
Byte Offset: 0x100c0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc7_dt_nooverride_2**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC7_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x4031
Byte Offset: 0x100c4
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc7_dt_nooverride_3**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC7_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x4032
Byte Offset: 0x100c8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc7_dt_nooverride_4**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC7_DT_OVERRIDE_0

Offset: 0x4033
Byte Offset: 0x100cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc7_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc7_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc7_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_0_VC8_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x4034
Byte Offset: 0x100d0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc8_dt_nooverride_0**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC8_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x4035
Byte Offset: 0x100d4
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc8_dt_nooverride_1**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC8_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x4036
Byte Offset: 0x100d8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc8_dt_nooverride_2**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC8_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x4037
Byte Offset: 0x100dc
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc8_dt_nooverride_3**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC8_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x4038
Byte Offset: 0x100e0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc8_dt_nooverride_4**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC8_DT_OVERRIDE_0

Offset: 0x4039
Byte Offset: 0x100e4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc8_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc8_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc8_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_0_VC9_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x403a
Byte Offset: 0x100e8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc9_dt_nooverride_0**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC9_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x403b
Byte Offset: 0x100ec
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc9_dt_nooverride_1**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC9_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x403c
Byte Offset: 0x100f0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc9_dt_nooverride_2**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC9_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x403d
Byte Offset: 0x100f4
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc9_dt_nooverride_3**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC9_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x403e
Byte Offset: 0x100f8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc9_dt_nooverride_4**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC9_DT_OVERRIDE_0

Offset: 0x403f
Byte Offset: 0x100fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc9_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc9_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc9_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_0_VC10_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x4040
Byte Offset: 0x10100
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc10_dt_nooverride_0**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC10_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x4041
Byte Offset: 0x10104
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc10_dt_nooverride_1**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC10_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x4042
Byte Offset: 0x10108
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc10_dt_nooverride_2**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC10_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x4043
Byte Offset: 0x1010c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc10_dt_nooverride_3**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC10_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x4044
Byte Offset: 0x10110
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc10_dt_nooverride_4**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC10_DT_OVERRIDE_0

Offset: 0x4045
Byte Offset: 0x10114
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc10_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc10_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc10_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_0_VC11_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x4046
Byte Offset: 0x10118
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc11_dt_nooverride_0**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC11_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x4047
Byte Offset: 0x1011c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc11_dt_nooverride_1**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC11_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x4048
Byte Offset: 0x10120
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc11_dt_nooverride_2**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC11_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x4049
Byte Offset: 0x10124
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc11_dt_nooverride_3**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC11_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x404a
Byte Offset: 0x10128
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc11_dt_nooverride_4**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC11_DT_OVERRIDE_0

Offset: 0x404b
Byte Offset: 0x1012c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc11_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc11_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc11_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_0_VC12_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x404c
Byte Offset: 0x10130
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc12_dt_nooverride_0**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC12_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x404d
Byte Offset: 0x10134
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc12_dt_nooverride_1**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC12_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x404e
Byte Offset: 0x10138
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc12_dt_nooverride_2**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC12_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x404f
Byte Offset: 0x1013c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc12_dt_nooverride_3**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC12_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x4050
Byte Offset: 0x10140
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc12_dt_nooverride_4**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC12_DT_OVERRIDE_0

Offset: 0x4051
Byte Offset: 0x10144
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc12_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc12_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc12_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_0_VC13_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x4052
Byte Offset: 0x10148
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc13_dt_nooverride_0**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC13_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x4053
Byte Offset: 0x1014c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc13_dt_nooverride_1**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC13_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x4054
Byte Offset: 0x10150
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc13_dt_nooverride_2**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC13_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x4055
Byte Offset: 0x10154
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc13_dt_nooverride_3**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC13_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x4056
Byte Offset: 0x10158
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc13_dt_nooverride_4**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC13_DT_OVERRIDE_0

Offset: 0x4057
Byte Offset: 0x1015c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc13_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc13_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc13_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_0_VC14_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x4058
Byte Offset: 0x10160
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc14_dt_nooverride_0**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC14_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x4059
Byte Offset: 0x10164
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc14_dt_nooverride_1**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC14_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x405a
Byte Offset: 0x10168
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc14_dt_nooverride_2**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC14_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x405b
Byte Offset: 0x1016c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc14_dt_nooverride_3**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC14_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x405c
Byte Offset: 0x10170
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc14_dt_nooverride_4**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC14_DT_OVERRIDE_0

Offset: 0x405d
Byte Offset: 0x10174
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc14_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc14_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc14_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_0_VC15_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x405e
Byte Offset: 0x10178
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc15_dt_nooverride_0**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC15_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x405f
Byte Offset: 0x1017c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc15_dt_nooverride_1**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC15_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x4060
Byte Offset: 0x10180
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc15_dt_nooverride_2**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC15_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x4061
Byte Offset: 0x10184
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc15_dt_nooverride_3**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC15_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x4062
Byte Offset: 0x10188
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc15_dt_nooverride_4**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_0_VC15_DT_OVERRIDE_0

Offset: 0x4063
Byte Offset: 0x1018c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc15_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc15_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc15_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_0_PPFSM_TIMEOUT_CTRL_0

Offset: 0x4064
Byte Offset: 0x10190
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x7fffffff (0b0111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31	DISABLE	**cfg_timeout_en**: Enable Timeout counter for the PP FSM **0** = DISABLE **1** = ENABLE
30:0	0x7fffffff	**cfg_timeout_period**: Timeout period

NVCSI_STREAM_0_PH_CHK_CTRL_0

Offset: 0x4065
Byte Offset: 0x10194
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	ENABLE	**cfg_ph_16_vc**: 16 VC support **0** = DISABLE **1** = ENABLE
1	ENABLE	**cfg_ph_crc_chk_en**: PH CRC check enable (only for CPHY case), when this bit is set to 0, the packet header will still be decode when the CRC check fail, but the error will be set. **0** = DISABLE **1** = ENABLE
0	ENABLE	**cfg_ph_ecc_chk_en**: PH ECC check enable (only for DPHY case), when this bit is set to 0, the packet header will still be decode when the ECC check fail, but the error will be set. **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_0_VC0_DPCM_CTRL_0

Offset: 0x4066
Byte Offset: 0x10198
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc0_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_0_VC1_DPCM_CTRL_0

Offset: 0x4067
Byte Offset: 0x1019c

Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc1_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_0_VC2_DPCM_CTRL_0

Offset: 0x4068
Byte Offset: 0x101a0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc2_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_0_VC3_DPCM_CTRL_0

Offset: 0x4069

Byte Offset: 0x101a4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc3_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_0_VC4_DPCM_CTRL_0

Offset: 0x406a
Byte Offset: 0x101a8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc4_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_0_VC5_DPCM_CTRL_0

Offset: 0x406b
Byte Offset: 0x101ac
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc5_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_0_VC6_DPCM_CTRL_0

Offset: 0x406c
Byte Offset: 0x101b0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc6_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_0_VC7_DPCM_CTRL_0

Offset: 0x406d
Byte Offset: 0x101b4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc7_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_0_VC8_DPCM_CTRL_0

Offset: 0x406e
Byte Offset: 0x101b8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc8_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_0_VC9_DPCM_CTRL_0

Offset: 0x406f
Byte Offset: 0x101bc
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc9_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_0_VC10_DPCM_CTRL_0

Offset: 0x4070
Byte Offset: 0x101c0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc10_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_0_VC11_DPCM_CTRL_0

Offset: 0x4071
Byte Offset: 0x101c4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc11_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_0_VC12_DPCM_CTRL_0

Offset: 0x4072
Byte Offset: 0x101c8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc12_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_0_VC13_DPCM_CTRL_0

Offset: 0x4073
Byte Offset: 0x101cc
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc13_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_0_VC14_DPCM_CTRL_0

Offset: 0x4074
Byte Offset: 0x101d0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc14_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_0_VC15_DPCM_CTRL_0

Offset: 0x4075
Byte Offset: 0x101d4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc15_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_0_PF_CRC_0

Status register on packet data CRC
Offset: 0x4076
Byte Offset: 0x101d8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**calc_crc**: 16 bit CRC computed over current packet (has to match with the CRC in PF for a good packet)
15:0	0x0	**rx_crc**: 16 bit CRC from PF

NVCSI_STREAM_0_PH_WC_0

Status register on WC
Offset: 0x4077
Byte Offset: 0x101dc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**calc_wc**: 16 bit WC computed over current packet (has to match with the WC in PH for a good packet)
15:0	0x0	**rx_wc**: 16 bit WC from PH

NVCSI_STREAM_0_PH_DI_0

Status register on Data ID
Offset: 0x4078
Byte Offset: 0x101e0
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9:6	0x0	**rx_vc**: 4 bit VC from PH
5:0	0x0	**rx_dt**: 6 bit DTYPE from PH

NVCSI_STREAM_0_ERROR_STATUS2VI_MASK_0

Offset: 0x4079
Byte Offset: 0x101e4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15	0x0	**cfg_err_status2vi_mask_vc15**: for VC15 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
14	0x0	**cfg_err_status2vi_mask_vc14**: for VC14 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
13	0x0	**cfg_err_status2vi_mask_vc13**: for VC13 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
12	0x0	**cfg_err_status2vi_mask_vc12**: for VC12 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
11	0x0	**cfg_err_status2vi_mask_vc11**: for VC11 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
10	0x0	**cfg_err_status2vi_mask_vc10**: for VC10 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
9	0x0	**cfg_err_status2vi_mask_vc9**: for VC9 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
8	0x0	**cfg_err_status2vi_mask_vc8**: for VC8 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
7	0x0	**cfg_err_status2vi_mask_vc7**: for VC7 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
6	0x0	**cfg_err_status2vi_mask_vc6**: for VC6 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
5	0x0	**cfg_err_status2vi_mask_vc5**: for VC5 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
4	0x0	**cfg_err_status2vi_mask_vc4**: for VC4 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
3	0x0	**cfg_err_status2vi_mask_vc3**: for VC3 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
2	0x0	**cfg_err_status2vi_mask_vc2**: for VC2 : 0 = err_status2vi_vc2 will be send to VI at EOF; 1 = No error will be send to VI at EOF

Bit	Reset	Description
1	0x0	**cfg_err_status2vi_mask_vc1**: for VC1 : 0 = err_status2vi_vc1 will be send to VI at EOF; 1 = No error will be send to VI at EOF
0	0x0	**cfg_err_status2vi_mask_vc0**: for VC0 : 0 = err_status2vi_vc0 will be send to VI at EOF; 1 = No error will be send to VI at EOF

NVCSI_STREAM_0_ERROR_STATUS2VI_VC0_0

This is RO register for SW
Offset: 0x407a
Byte Offset: 0x101e8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc0**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_0_ERROR_STATUS2VI_VC1_0

This is RO register for SW
Offset: 0x407b
Byte Offset: 0x101ec
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc1**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_0_ERROR_STATUS2VI_VC2_0

This is RO register for SW
Offset: 0x407c
Byte Offset: 0x101f0
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc2**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_0_ERROR_STATUS2VI_VC3_0

This is RO register for SW
Offset: 0x407d
Byte Offset: 0x101f4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc3**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_0_ERROR_STATUS2VI_VC4_0

This is RO register for SW
Offset: 0x407e
Byte Offset: 0x101f8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc4**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_0_ERROR_STATUS2VI_VC5_0

This is RO register for SW
Offset: 0x407f
Byte Offset: 0x101fc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc5**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_0_ERROR_STATUS2VI_VC6_0

This is RO register for SW
Offset: 0x4080
Byte Offset: 0x10200
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc6**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_0_ERROR_STATUS2VI_VC7_0

This is RO register for SW
Offset: 0x4081
Byte Offset: 0x10204
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc7**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_0_ERROR_STATUS2VI_VC8_0

This is RO register for SW
Offset: 0x4082
Byte Offset: 0x10208
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc8**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_0_ERROR_STATUS2VI_VC9_0

This is RO register for SW
Offset: 0x4083
Byte Offset: 0x1020c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc9**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_0_ERROR_STATUS2VI_VC10_0

This is RO register for SW
Offset: 0x4084
Byte Offset: 0x10210
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc10**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_0_ERROR_STATUS2VI_VC11_0

This is RO register for SW
Offset: 0x4085
Byte Offset: 0x10214
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc11**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_0_ERROR_STATUS2VI_VC12_0

This is RO register for SW
Offset: 0x4086
Byte Offset: 0x10218
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc12**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_0_ERROR_STATUS2VI_VC13_0

This is RO register for SW
Offset: 0x4087
Byte Offset: 0x1021c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc13**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_0_ERROR_STATUS2VI_VC14_0

This is RO register for SW
Offset: 0x4088
Byte Offset: 0x10220
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc14**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_0_ERROR_STATUS2VI_VC15_0

This is RO register for SW
Offset: 0x4089
Byte Offset: 0x10224
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc15**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_0_INTR_STATUS_0

Offset: 0x408a
Byte Offset: 0x10228
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**intr_stat_novc**: Not VC related interrupt
15	0x0	**intr_stat_vc15**: VC15 event
14	0x0	**intr_stat_vc14**: VC14 event
13	0x0	**intr_stat_vc13**: VC13 event
12	0x0	**intr_stat_vc12**: VC12 event
11	0x0	**intr_stat_vc11**: VC11 event
10	0x0	**intr_stat_vc10**: VC10 event
9	0x0	**intr_stat_vc9**: VC9 event
8	0x0	**intr_stat_vc8**: VC8 event
7	0x0	**intr_stat_vc7**: VC7 event
6	0x0	**intr_stat_vc6**: VC6 event

Bit	Reset	Description
5	0x0	**intr_stat_vc5**: VC5 event
4	0x0	**intr_stat_vc4**: VC4 event
3	0x0	**intr_stat_vc3**: VC3 event
2	0x0	**intr_stat_vc2**: VC2 event
1	0x0	**intr_stat_vc1**: VC1 event
0	0x0	**intr_stat_vc0**: VC0 event

NVCSI_STREAM_0_INTR_STATUS_NOVC_0

Offset: 0x408b
Byte Offset: 0x1022c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**intr_stat_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**intr_stat_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_0_INTR_STATUS_VC0_0

Offset: 0x408c
Byte Offset: 0x10230
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc0**
4	0x0	**intr_stat_ph_single_crc_err_vc0**
3	0x0	**intr_stat_pd_wc_short_err_vc0**
2	0x0	**intr_stat_pd_crc_err_vc0**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc0**
0	0x0	**intr_stat_ppfsm_timeout_vc0**

NVCSI_STREAM_0_INTR_STATUS_VC1_0

Offset: 0x408d
Byte Offset: 0x10234
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc1**
4	0x0	**intr_stat_ph_single_crc_err_vc1**
3	0x0	**intr_stat_pd_wc_short_err_vc1**
2	0x0	**intr_stat_pd_crc_err_vc1**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc1**
0	0x0	**intr_stat_ppfsm_timeout_vc1**

NVCSI_STREAM_0_INTR_STATUS_VC2_0

Offset: 0x408e
Byte Offset: 0x10238
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc2**
4	0x0	**intr_stat_ph_single_crc_err_vc2**
3	0x0	**intr_stat_pd_wc_short_err_vc2**
2	0x0	**intr_stat_pd_crc_err_vc2**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc2**
0	0x0	**intr_stat_ppfsm_timeout_vc2**

NVCSI_STREAM_0_INTR_STATUS_VC3_0

Offset: 0x408f
Byte Offset: 0x1023c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc3**
4	0x0	**intr_stat_ph_single_crc_err_vc3**
3	0x0	**intr_stat_pd_wc_short_err_vc3**
2	0x0	**intr_stat_pd_crc_err_vc3**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc3**
0	0x0	**intr_stat_ppfsm_timeout_vc3**

NVCSI_STREAM_0_INTR_STATUS_VC4_0

Offset: 0x4090
Byte Offset: 0x10240
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc4**
4	0x0	**intr_stat_ph_single_crc_err_vc4**
3	0x0	**intr_stat_pd_wc_short_err_vc4**
2	0x0	**intr_stat_pd_crc_err_vc4**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc4**
0	0x0	**intr_stat_ppfsm_timeout_vc4**

NVCSI_STREAM_0_INTR_STATUS_VC5_0

Offset: 0x4091
Byte Offset: 0x10244
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc5**
4	0x0	**intr_stat_ph_single_crc_err_vc5**
3	0x0	**intr_stat_pd_wc_short_err_vc5**
2	0x0	**intr_stat_pd_crc_err_vc5**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc5**
0	0x0	**intr_stat_ppfsm_timeout_vc5**

NVCSI_STREAM_0_INTR_STATUS_VC6_0

Offset: 0x4092
Byte Offset: 0x10248
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc6**
4	0x0	**intr_stat_ph_single_crc_err_vc6**
3	0x0	**intr_stat_pd_wc_short_err_vc6**
2	0x0	**intr_stat_pd_crc_err_vc6**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc6**
0	0x0	**intr_stat_ppfsm_timeout_vc6**

NVCSI_STREAM_0_INTR_STATUS_VC7_0

Offset: 0x4093
Byte Offset: 0x1024c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc7**
4	0x0	**intr_stat_ph_single_crc_err_vc7**
3	0x0	**intr_stat_pd_wc_short_err_vc7**
2	0x0	**intr_stat_pd_crc_err_vc7**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc7**
0	0x0	**intr_stat_ppfsm_timeout_vc7**

NVCSI_STREAM_0_INTR_STATUS_VC8_0

Offset: 0x4094
Byte Offset: 0x10250
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc8**
4	0x0	**intr_stat_ph_single_crc_err_vc8**
3	0x0	**intr_stat_pd_wc_short_err_vc8**
2	0x0	**intr_stat_pd_crc_err_vc8**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc8**
0	0x0	**intr_stat_ppfsm_timeout_vc8**

NVCSI_STREAM_0_INTR_STATUS_VC9_0

Offset: 0x4095
Byte Offset: 0x10254
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc9**
4	0x0	**intr_stat_ph_single_crc_err_vc9**
3	0x0	**intr_stat_pd_wc_short_err_vc9**
2	0x0	**intr_stat_pd_crc_err_vc9**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc9**
0	0x0	**intr_stat_ppfsm_timeout_vc9**

NVCSI_STREAM_0_INTR_STATUS_VC10_0

Offset: 0x4096
Byte Offset: 0x10258
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc10**
4	0x0	**intr_stat_ph_single_crc_err_vc10**
3	0x0	**intr_stat_pd_wc_short_err_vc10**
2	0x0	**intr_stat_pd_crc_err_vc10**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc10**
0	0x0	**intr_stat_ppfsm_timeout_vc10**

NVCSI_STREAM_0_INTR_STATUS_VC11_0

Offset: 0x4097
Byte Offset: 0x1025c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc11**
4	0x0	**intr_stat_ph_single_crc_err_vc11**
3	0x0	**intr_stat_pd_wc_short_err_vc11**
2	0x0	**intr_stat_pd_crc_err_vc11**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc11**
0	0x0	**intr_stat_ppfsm_timeout_vc11**

NVCSI_STREAM_0_INTR_STATUS_VC12_0

Offset: 0x4098
Byte Offset: 0x10260
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc12**
4	0x0	**intr_stat_ph_single_crc_err_vc12**
3	0x0	**intr_stat_pd_wc_short_err_vc12**
2	0x0	**intr_stat_pd_crc_err_vc12**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc12**
0	0x0	**intr_stat_ppfsm_timeout_vc12**

NVCSI_STREAM_0_INTR_STATUS_VC13_0

Offset: 0x4099
Byte Offset: 0x10264
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc13**
4	0x0	**intr_stat_ph_single_crc_err_vc13**
3	0x0	**intr_stat_pd_wc_short_err_vc13**
2	0x0	**intr_stat_pd_crc_err_vc13**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc13**
0	0x0	**intr_stat_ppfsm_timeout_vc13**

NVCSI_STREAM_0_INTR_STATUS_VC14_0

Offset: 0x409a
Byte Offset: 0x10268
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc14**
4	0x0	**intr_stat_ph_single_crc_err_vc14**
3	0x0	**intr_stat_pd_wc_short_err_vc14**
2	0x0	**intr_stat_pd_crc_err_vc14**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc14**
0	0x0	**intr_stat_ppfsm_timeout_vc14**

NVCSI_STREAM_0_INTR_STATUS_VC15_0

Offset: 0x409b
Byte Offset: 0x1026c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc15**
4	0x0	**intr_stat_ph_single_crc_err_vc15**
3	0x0	**intr_stat_pd_wc_short_err_vc15**
2	0x0	**intr_stat_pd_crc_err_vc15**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc15**
0	0x0	**intr_stat_ppfsm_timeout_vc15**

NVCSI_STREAM_0_INTR_MASK_NOVC_0

Offset: 0x409c
Byte Offset: 0x10270
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**intr_mask_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**intr_mask_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_0_INTR_MASK_VC0_0

Offset: 0x409d
Byte Offset: 0x10274
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc0**
4	0x0	**intr_mask_ph_single_crc_err_vc0**
3	0x0	**intr_mask_pd_wc_short_err_vc0**
2	0x0	**intr_mask_pd_crc_err_vc0**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc0**
0	0x0	**intr_mask_ppfsm_timeout_vc0**

NVCSI_STREAM_0_INTR_MASK_VC1_0

Offset: 0x409e
Byte Offset: 0x10278
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc1**
4	0x0	**intr_mask_ph_single_crc_err_vc1**

Bit	Reset	Description
3	0x0	**intr_mask_pd_wc_short_err_vc1**
2	0x0	**intr_mask_pd_crc_err_vc1**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc1**
0	0x0	**intr_mask_ppfsm_timeout_vc1**

NVCSI_STREAM_0_INTR_MASK_VC2_0

Offset: 0x409f
Byte Offset: 0x1027c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc2**
4	0x0	**intr_mask_ph_single_crc_err_vc2**
3	0x0	**intr_mask_pd_wc_short_err_vc2**
2	0x0	**intr_mask_pd_crc_err_vc2**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc2**
0	0x0	**intr_mask_ppfsm_timeout_vc2**

NVCSI_STREAM_0_INTR_MASK_VC3_0

Offset: 0x40a0
Byte Offset: 0x10280
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc3**

Bit	Reset	Description
4	0x0	**intr_mask_ph_single_crc_err_vc3**
3	0x0	**intr_mask_pd_wc_short_err_vc3**
2	0x0	**intr_mask_pd_crc_err_vc3**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc3**
0	0x0	**intr_mask_ppfsm_timeout_vc3**

NVCSI_STREAM_0_INTR_MASK_VC4_0

Offset: 0x40a1
Byte Offset: 0x10284
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc4**
4	0x0	**intr_mask_ph_single_crc_err_vc4**
3	0x0	**intr_mask_pd_wc_short_err_vc4**
2	0x0	**intr_mask_pd_crc_err_vc4**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc4**
0	0x0	**intr_mask_ppfsm_timeout_vc4**

NVCSI_STREAM_0_INTR_MASK_VC5_0

Offset: 0x40a2
Byte Offset: 0x10288
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc5**
4	0x0	**intr_mask_ph_single_crc_err_vc5**
3	0x0	**intr_mask_pd_wc_short_err_vc5**
2	0x0	**intr_mask_pd_crc_err_vc5**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc5**
0	0x0	**intr_mask_ppfsm_timeout_vc5**

NVCSI_STREAM_0_INTR_MASK_VC6_0

Offset: 0x40a3
Byte Offset: 0x1028c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc6**
4	0x0	**intr_mask_ph_single_crc_err_vc6**
3	0x0	**intr_mask_pd_wc_short_err_vc6**
2	0x0	**intr_mask_pd_crc_err_vc6**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc6**
0	0x0	**intr_mask_ppfsm_timeout_vc6**

NVCSI_STREAM_0_INTR_MASK_VC7_0

Offset: 0x40a4
Byte Offset: 0x10290
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc7**
4	0x0	**intr_mask_ph_single_crc_err_vc7**
3	0x0	**intr_mask_pd_wc_short_err_vc7**
2	0x0	**intr_mask_pd_crc_err_vc7**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc7**
0	0x0	**intr_mask_ppfsm_timeout_vc7**

NVCSI_STREAM_0_INTR_MASK_VC8_0

Offset: 0x40a5
Byte Offset: 0x10294
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc8**
4	0x0	**intr_mask_ph_single_crc_err_vc8**
3	0x0	**intr_mask_pd_wc_short_err_vc8**
2	0x0	**intr_mask_pd_crc_err_vc8**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc8**
0	0x0	**intr_mask_ppfsm_timeout_vc8**

NVCSI_STREAM_0_INTR_MASK_VC9_0

Offset: 0x40a6
Byte Offset: 0x10298
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc9**
4	0x0	**intr_mask_ph_single_crc_err_vc9**
3	0x0	**intr_mask_pd_wc_short_err_vc9**
2	0x0	**intr_mask_pd_crc_err_vc9**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc9**
0	0x0	**intr_mask_ppfsm_timeout_vc9**

NVCSI_STREAM_0_INTR_MASK_VC10_0

Offset: 0x40a7
Byte Offset: 0x1029c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc10**
4	0x0	**intr_mask_ph_single_crc_err_vc10**
3	0x0	**intr_mask_pd_wc_short_err_vc10**
2	0x0	**intr_mask_pd_crc_err_vc10**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc10**
0	0x0	**intr_mask_ppfsm_timeout_vc10**

NVCSI_STREAM_0_INTR_MASK_VC11_0

Offset: 0x40a8
Byte Offset: 0x102a0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc11**
4	0x0	**intr_mask_ph_single_crc_err_vc11**
3	0x0	**intr_mask_pd_wc_short_err_vc11**
2	0x0	**intr_mask_pd_crc_err_vc11**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc11**
0	0x0	**intr_mask_ppfsm_timeout_vc11**

NVCSI_STREAM_0_INTR_MASK_VC12_0

Offset: 0x40a9
Byte Offset: 0x102a4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc12**
4	0x0	**intr_mask_ph_single_crc_err_vc12**
3	0x0	**intr_mask_pd_wc_short_err_vc12**
2	0x0	**intr_mask_pd_crc_err_vc12**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc12**
0	0x0	**intr_mask_ppfsm_timeout_vc12**

NVCSI_STREAM_0_INTR_MASK_VC13_0

Offset: 0x40aa
Byte Offset: 0x102a8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc13**
4	0x0	**intr_mask_ph_single_crc_err_vc13**
3	0x0	**intr_mask_pd_wc_short_err_vc13**
2	0x0	**intr_mask_pd_crc_err_vc13**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc13**
0	0x0	**intr_mask_ppfsm_timeout_vc13**

NVCSI_STREAM_0_INTR_MASK_VC14_0

Offset: 0x40ab
Byte Offset: 0x102ac
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc14**
4	0x0	**intr_mask_ph_single_crc_err_vc14**
3	0x0	**intr_mask_pd_wc_short_err_vc14**
2	0x0	**intr_mask_pd_crc_err_vc14**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc14**
0	0x0	**intr_mask_ppfsm_timeout_vc14**

NVCSI_STREAM_0_INTR_MASK_VC15_0

Offset: 0x40ac
Byte Offset: 0x102b0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc15**
4	0x0	**intr_mask_ph_single_crc_err_vc15**
3	0x0	**intr_mask_pd_wc_short_err_vc15**
2	0x0	**intr_mask_pd_crc_err_vc15**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc15**
0	0x0	**intr_mask_ppfsm_timeout_vc15**

NVCSI_STREAM_0_CORRECTABLE_ERR_INTR_STATUS_0

Offset: 0x40ad
Byte Offset: 0x102b4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**err_intr_stat_novc**: Not VC related interrupt
15	0x0	**err_intr_stat_vc15**: VC15 event
14	0x0	**err_intr_stat_vc14**: VC14 event
13	0x0	**err_intr_stat_vc13**: VC13 event
12	0x0	**err_intr_stat_vc12**: VC12 event
11	0x0	**err_intr_stat_vc11**: VC11 event
10	0x0	**err_intr_stat_vc10**: VC10 event
9	0x0	**err_intr_stat_vc9**: VC9 event
8	0x0	**err_intr_stat_vc8**: VC8 event
7	0x0	**err_intr_stat_vc7**: VC7 event

Bit	Reset	Description
6	0x0	**err_intr_stat_vc6**: VC6 event
5	0x0	**err_intr_stat_vc5**: VC5 event
4	0x0	**err_intr_stat_vc4**: VC4 event
3	0x0	**err_intr_stat_vc3**: VC3 event
2	0x0	**err_intr_stat_vc2**: VC2 event
1	0x0	**err_intr_stat_vc1**: VC1 event
0	0x0	**err_intr_stat_vc0**: VC0 event

NVCSI_STREAM_0_CORRECTABLE_ERR_INTR_STATUS_NOVC_0

Offset: 0x40ae
Byte Offset: 0x102b8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**err_intr_stat_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**err_intr_stat_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_0_CORRECTABLE_ERR_INTR_STATUS_VC0_0

Offset: 0x40af
Byte Offset: 0x102bc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc0**
4	0x0	**err_intr_stat_ph_single_crc_err_vc0**
3	0x0	**err_intr_stat_pd_wc_short_err_vc0**
2	0x0	**err_intr_stat_pd_crc_err_vc0**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc0**
0	0x0	**err_intr_stat_ppfsm_timeout_vc0**

NVCSI_STREAM_0_CORRECTABLE_ERR_INTR_STATUS_VC1_0

Offset: 0x40b0
Byte Offset: 0x102c0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc1**
4	0x0	**err_intr_stat_ph_single_crc_err_vc1**
3	0x0	**err_intr_stat_pd_wc_short_err_vc1**
2	0x0	**err_intr_stat_pd_crc_err_vc1**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc1**
0	0x0	**err_intr_stat_ppfsm_timeout_vc1**

NVCSI_STREAM_0_CORRECTABLE_ERR_INTR_STATUS_VC2_0

Offset: 0x40b1
Byte Offset: 0x102c4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc2**
4	0x0	**err_intr_stat_ph_single_crc_err_vc2**
3	0x0	**err_intr_stat_pd_wc_short_err_vc2**
2	0x0	**err_intr_stat_pd_crc_err_vc2**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc2**
0	0x0	**err_intr_stat_ppfsm_timeout_vc2**

NVCSI_STREAM_0_CORRECTABLE_ERR_INTR_STATUS_VC3_0

Offset: 0x40b2
Byte Offset: 0x102c8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc3**
4	0x0	**err_intr_stat_ph_single_crc_err_vc3**
3	0x0	**err_intr_stat_pd_wc_short_err_vc3**
2	0x0	**err_intr_stat_pd_crc_err_vc3**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc3**
0	0x0	**err_intr_stat_ppfsm_timeout_vc3**

NVCSI_STREAM_0_CORRECTABLE_ERR_INTR_STATUS_VC4_0

Offset: 0x40b3
Byte Offset: 0x102cc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc4**
4	0x0	**err_intr_stat_ph_single_crc_err_vc4**
3	0x0	**err_intr_stat_pd_wc_short_err_vc4**
2	0x0	**err_intr_stat_pd_crc_err_vc4**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc4**
0	0x0	**err_intr_stat_ppfsm_timeout_vc4**

NVCSI_STREAM_0_CORRECTABLE_ERR_INTR_STATUS_VC5_0

Offset: 0x40b4
Byte Offset: 0x102d0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc5**
4	0x0	**err_intr_stat_ph_single_crc_err_vc5**
3	0x0	**err_intr_stat_pd_wc_short_err_vc5**
2	0x0	**err_intr_stat_pd_crc_err_vc5**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc5**
0	0x0	**err_intr_stat_ppfsm_timeout_vc5**

NVCSI_STREAM_0_CORRECTABLE_ERR_INTR_STATUS_VC6_0

Offset: 0x40b5
Byte Offset: 0x102d4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc6**
4	0x0	**err_intr_stat_ph_single_crc_err_vc6**
3	0x0	**err_intr_stat_pd_wc_short_err_vc6**
2	0x0	**err_intr_stat_pd_crc_err_vc6**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc6**
0	0x0	**err_intr_stat_ppfsm_timeout_vc6**

NVCSI_STREAM_0_CORRECTABLE_ERR_INTR_STATUS_VC7_0

Offset: 0x40b6
Byte Offset: 0x102d8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc7**
4	0x0	**err_intr_stat_ph_single_crc_err_vc7**
3	0x0	**err_intr_stat_pd_wc_short_err_vc7**
2	0x0	**err_intr_stat_pd_crc_err_vc7**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc7**
0	0x0	**err_intr_stat_ppfsm_timeout_vc7**

NVCSI_STREAM_0_CORRECTABLE_ERR_INTR_STATUS_VC8_0

Offset: 0x40b7
Byte Offset: 0x102dc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc8**
4	0x0	**err_intr_stat_ph_single_crc_err_vc8**
3	0x0	**err_intr_stat_pd_wc_short_err_vc8**
2	0x0	**err_intr_stat_pd_crc_err_vc8**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc8**
0	0x0	**err_intr_stat_ppfsm_timeout_vc8**

NVCSI_STREAM_0_CORRECTABLE_ERR_INTR_STATUS_VC9_0

Offset: 0x40b8
Byte Offset: 0x102e0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc9**
4	0x0	**err_intr_stat_ph_single_crc_err_vc9**
3	0x0	**err_intr_stat_pd_wc_short_err_vc9**
2	0x0	**err_intr_stat_pd_crc_err_vc9**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc9**
0	0x0	**err_intr_stat_ppfsm_timeout_vc9**

NVCSI_STREAM_0_CORRECTABLE_ERR_INTR_STATUS_VC10_0

Offset: 0x40b9
Byte Offset: 0x102e4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc10**
4	0x0	**err_intr_stat_ph_single_crc_err_vc10**
3	0x0	**err_intr_stat_pd_wc_short_err_vc10**
2	0x0	**err_intr_stat_pd_crc_err_vc10**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc10**
0	0x0	**err_intr_stat_ppfsm_timeout_vc10**

NVCSI_STREAM_0_CORRECTABLE_ERR_INTR_STATUS_VC11_0

Offset: 0x40ba
Byte Offset: 0x102e8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc11**
4	0x0	**err_intr_stat_ph_single_crc_err_vc11**
3	0x0	**err_intr_stat_pd_wc_short_err_vc11**
2	0x0	**err_intr_stat_pd_crc_err_vc11**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc11**
0	0x0	**err_intr_stat_ppfsm_timeout_vc11**

NVCSI_STREAM_0_CORRECTABLE_ERR_INTR_STATUS_VC12_0

Offset: 0x40bb
Byte Offset: 0x102ec
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc12**
4	0x0	**err_intr_stat_ph_single_crc_err_vc12**
3	0x0	**err_intr_stat_pd_wc_short_err_vc12**
2	0x0	**err_intr_stat_pd_crc_err_vc12**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc12**
0	0x0	**err_intr_stat_ppfsm_timeout_vc12**

NVCSI_STREAM_0_CORRECTABLE_ERR_INTR_STATUS_VC13_0

Offset: 0x40bc
Byte Offset: 0x102f0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc13**
4	0x0	**err_intr_stat_ph_single_crc_err_vc13**
3	0x0	**err_intr_stat_pd_wc_short_err_vc13**
2	0x0	**err_intr_stat_pd_crc_err_vc13**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc13**
0	0x0	**err_intr_stat_ppfsm_timeout_vc13**

NVCSI_STREAM_0_CORRECTABLE_ERR_INTR_STATUS_VC14_0

Offset: 0x40bd
Byte Offset: 0x102f4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc14**
4	0x0	**err_intr_stat_ph_single_crc_err_vc14**
3	0x0	**err_intr_stat_pd_wc_short_err_vc14**
2	0x0	**err_intr_stat_pd_crc_err_vc14**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc14**
0	0x0	**err_intr_stat_ppfsm_timeout_vc14**

NVCSI_STREAM_0_CORRECTABLE_ERR_INTR_STATUS_VC15_0

Offset: 0x40be
Byte Offset: 0x102f8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc15**
4	0x0	**err_intr_stat_ph_single_crc_err_vc15**
3	0x0	**err_intr_stat_pd_wc_short_err_vc15**
2	0x0	**err_intr_stat_pd_crc_err_vc15**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc15**
0	0x0	**err_intr_stat_ppfsm_timeout_vc15**

NVCSI_STREAM_0_UNCORRECTABLE_ERR_INTR_STATUS_0

Offset: 0x40bf
Byte Offset: 0x102fc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**err_intr_stat_novc**: Not VC related interrupt
15	0x0	**err_intr_stat_vc15**: VC15 event
14	0x0	**err_intr_stat_vc14**: VC14 event
13	0x0	**err_intr_stat_vc13**: VC13 event
12	0x0	**err_intr_stat_vc12**: VC12 event
11	0x0	**err_intr_stat_vc11**: VC11 event
10	0x0	**err_intr_stat_vc10**: VC10 event
9	0x0	**err_intr_stat_vc9**: VC9 event
8	0x0	**err_intr_stat_vc8**: VC8 event
7	0x0	**err_intr_stat_vc7**: VC7 event
6	0x0	**err_intr_stat_vc6**: VC6 event
5	0x0	**err_intr_stat_vc5**: VC5 event
4	0x0	**err_intr_stat_vc4**: VC4 event
3	0x0	**err_intr_stat_vc3**: VC3 event
2	0x0	**err_intr_stat_vc2**: VC2 event
1	0x0	**err_intr_stat_vc1**: VC1 event
0	0x0	**err_intr_stat_vc0**: VC0 event

NVCSI_STREAM_0_UNCORRECTABLE_ERR_INTR_STATUS_NOVC_0

Offset: 0x40c0

Byte Offset: 0x10300
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**err_intr_stat_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**err_intr_stat_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_0_UNCORRECTABLE_ERR_INTR_STATUS_VC0_0

Offset: 0x40c1
Byte Offset: 0x10304
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc0**
4	0x0	**err_intr_stat_ph_single_crc_err_vc0**
3	0x0	**err_intr_stat_pd_wc_short_err_vc0**
2	0x0	**err_intr_stat_pd_crc_err_vc0**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc0**
0	0x0	**err_intr_stat_ppfsm_timeout_vc0**

NVCSI_STREAM_0_UNCORRECTABLE_ERR_INTR_STATUS_VC1_0

Offset: 0x40c2
Byte Offset: 0x10308
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc1**
4	0x0	**err_intr_stat_ph_single_crc_err_vc1**
3	0x0	**err_intr_stat_pd_wc_short_err_vc1**
2	0x0	**err_intr_stat_pd_crc_err_vc1**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc1**
0	0x0	**err_intr_stat_ppfsm_timeout_vc1**

NVCSI_STREAM_0_UNCORRECTABLE_ERR_INTR_STATUS_VC2_0

Offset: 0x40c3
Byte Offset: 0x1030c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc2**
4	0x0	**err_intr_stat_ph_single_crc_err_vc2**
3	0x0	**err_intr_stat_pd_wc_short_err_vc2**
2	0x0	**err_intr_stat_pd_crc_err_vc2**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc2**
0	0x0	**err_intr_stat_ppfsm_timeout_vc2**

NVCSI_STREAM_0_UNCORRECTABLE_ERR_INTR_STATUS_VC3_0

Offset: 0x40c4
Byte Offset: 0x10310
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc3**
4	0x0	**err_intr_stat_ph_single_crc_err_vc3**
3	0x0	**err_intr_stat_pd_wc_short_err_vc3**
2	0x0	**err_intr_stat_pd_crc_err_vc3**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc3**
0	0x0	**err_intr_stat_ppfsm_timeout_vc3**

NVCSI_STREAM_0_UNCORRECTABLE_ERR_INTR_STATUS_VC4_0

Offset: 0x40c5
Byte Offset: 0x10314
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc4**
4	0x0	**err_intr_stat_ph_single_crc_err_vc4**
3	0x0	**err_intr_stat_pd_wc_short_err_vc4**
2	0x0	**err_intr_stat_pd_crc_err_vc4**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc4**
0	0x0	**err_intr_stat_ppfsm_timeout_vc4**

NVCSI_STREAM_0_UNCORRECTABLE_ERR_INTR_STATUS_VC5_0

Offset: 0x40c6
Byte Offset: 0x10318
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc5**
4	0x0	**err_intr_stat_ph_single_crc_err_vc5**
3	0x0	**err_intr_stat_pd_wc_short_err_vc5**
2	0x0	**err_intr_stat_pd_crc_err_vc5**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc5**
0	0x0	**err_intr_stat_ppfsm_timeout_vc5**

NVCSI_STREAM_0_UNCORRECTABLE_ERR_INTR_STATUS_VC6_0

Offset: 0x40c7
Byte Offset: 0x1031c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc6**
4	0x0	**err_intr_stat_ph_single_crc_err_vc6**
3	0x0	**err_intr_stat_pd_wc_short_err_vc6**
2	0x0	**err_intr_stat_pd_crc_err_vc6**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc6**
0	0x0	**err_intr_stat_ppfsm_timeout_vc6**

NVCSI_STREAM_0_UNCORRECTABLE_ERR_INTR_STATUS_VC7_0

Offset: 0x40c8
Byte Offset: 0x10320
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc7**
4	0x0	**err_intr_stat_ph_single_crc_err_vc7**
3	0x0	**err_intr_stat_pd_wc_short_err_vc7**
2	0x0	**err_intr_stat_pd_crc_err_vc7**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc7**
0	0x0	**err_intr_stat_ppfsm_timeout_vc7**

NVCSI_STREAM_0_UNCORRECTABLE_ERR_INTR_STATUS_VC8_0

Offset: 0x40c9
Byte Offset: 0x10324
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc8**
4	0x0	**err_intr_stat_ph_single_crc_err_vc8**
3	0x0	**err_intr_stat_pd_wc_short_err_vc8**
2	0x0	**err_intr_stat_pd_crc_err_vc8**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc8**
0	0x0	**err_intr_stat_ppfsm_timeout_vc8**

NVCSI_STREAM_0_UNCORRECTABLE_ERR_INTR_STATUS_VC9_0

Offset: 0x40ca
Byte Offset: 0x10328
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc9**
4	0x0	**err_intr_stat_ph_single_crc_err_vc9**
3	0x0	**err_intr_stat_pd_wc_short_err_vc9**
2	0x0	**err_intr_stat_pd_crc_err_vc9**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc9**
0	0x0	**err_intr_stat_ppfsm_timeout_vc9**

NVCSI_STREAM_0_UNCORRECTABLE_ERR_INTR_STATUS_VC10_0

Offset: 0x40cb
Byte Offset: 0x1032c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc10**
4	0x0	**err_intr_stat_ph_single_crc_err_vc10**
3	0x0	**err_intr_stat_pd_wc_short_err_vc10**
2	0x0	**err_intr_stat_pd_crc_err_vc10**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc10**
0	0x0	**err_intr_stat_ppfsm_timeout_vc10**

NVCSI_STREAM_0_UNCORRECTABLE_ERR_INTR_STATUS_VC11_0

Offset: 0x40cc
Byte Offset: 0x10330
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc11**
4	0x0	**err_intr_stat_ph_single_crc_err_vc11**
3	0x0	**err_intr_stat_pd_wc_short_err_vc11**
2	0x0	**err_intr_stat_pd_crc_err_vc11**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc11**
0	0x0	**err_intr_stat_ppfsm_timeout_vc11**

NVCSI_STREAM_0_UNCORRECTABLE_ERR_INTR_STATUS_VC12_0

Offset: 0x40cd
Byte Offset: 0x10334
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc12**
4	0x0	**err_intr_stat_ph_single_crc_err_vc12**
3	0x0	**err_intr_stat_pd_wc_short_err_vc12**
2	0x0	**err_intr_stat_pd_crc_err_vc12**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc12**
0	0x0	**err_intr_stat_ppfsm_timeout_vc12**

NVCSI_STREAM_0_UNCORRECTABLE_ERR_INTR_STATUS_VC13_0

Offset: 0x40ce
Byte Offset: 0x10338
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc13**
4	0x0	**err_intr_stat_ph_single_crc_err_vc13**
3	0x0	**err_intr_stat_pd_wc_short_err_vc13**
2	0x0	**err_intr_stat_pd_crc_err_vc13**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc13**
0	0x0	**err_intr_stat_ppfsm_timeout_vc13**

NVCSI_STREAM_0_UNCORRECTABLE_ERR_INTR_STATUS_VC14_0

Offset: 0x40cf
Byte Offset: 0x1033c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc14**
4	0x0	**err_intr_stat_ph_single_crc_err_vc14**
3	0x0	**err_intr_stat_pd_wc_short_err_vc14**
2	0x0	**err_intr_stat_pd_crc_err_vc14**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc14**
0	0x0	**err_intr_stat_ppfsm_timeout_vc14**

NVCSI_STREAM_0_UNCORRECTABLE_ERR_INTR_STATUS_VC15_0

Offset: 0x40d0
Byte Offset: 0x10340
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc15**
4	0x0	**err_intr_stat_ph_single_crc_err_vc15**
3	0x0	**err_intr_stat_pd_wc_short_err_vc15**
2	0x0	**err_intr_stat_pd_crc_err_vc15**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc15**
0	0x0	**err_intr_stat_ppfsm_timeout_vc15**

NVCSI_STREAM_0_ERR_INTR_MASK_NOVC_0

Offset: 0x40d1
Byte Offset: 0x10344
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**err_intr_mask_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**err_intr_mask_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_0_ERR_INTR_MASK_VC0_0

Offset: 0x40d2
Byte Offset: 0x10348
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc0**

Bit	Reset	Description
4	0x0	**err_intr_mask_ph_single_crc_err_vc0**
3	0x0	**err_intr_mask_pd_wc_short_err_vc0**
2	0x0	**err_intr_mask_pd_crc_err_vc0**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc0**
0	0x0	**err_intr_mask_ppfsm_timeout_vc0**

NVCSI_STREAM_0_ERR_INTR_MASK_VC1_0

Offset: 0x40d3
Byte Offset: 0x1034c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc1**
4	0x0	**err_intr_mask_ph_single_crc_err_vc1**
3	0x0	**err_intr_mask_pd_wc_short_err_vc1**
2	0x0	**err_intr_mask_pd_crc_err_vc1**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc1**
0	0x0	**err_intr_mask_ppfsm_timeout_vc1**

NVCSI_STREAM_0_ERR_INTR_MASK_VC2_0

Offset: 0x40d4
Byte Offset: 0x10350
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc2**
4	0x0	**err_intr_mask_ph_single_crc_err_vc2**
3	0x0	**err_intr_mask_pd_wc_short_err_vc2**
2	0x0	**err_intr_mask_pd_crc_err_vc2**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc2**
0	0x0	**err_intr_mask_ppfsm_timeout_vc2**

NVCSI_STREAM_0_ERR_INTR_MASK_VC3_0

Offset: 0x40d5
Byte Offset: 0x10354
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc3**
4	0x0	**err_intr_mask_ph_single_crc_err_vc3**
3	0x0	**err_intr_mask_pd_wc_short_err_vc3**
2	0x0	**err_intr_mask_pd_crc_err_vc3**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc3**
0	0x0	**err_intr_mask_ppfsm_timeout_vc3**

NVCSI_STREAM_0_ERR_INTR_MASK_VC4_0

Offset: 0x40d6
Byte Offset: 0x10358
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc4**
4	0x0	**err_intr_mask_ph_single_crc_err_vc4**
3	0x0	**err_intr_mask_pd_wc_short_err_vc4**
2	0x0	**err_intr_mask_pd_crc_err_vc4**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc4**
0	0x0	**err_intr_mask_ppfsm_timeout_vc4**

NVCSI_STREAM_0_ERR_INTR_MASK_VC5_0

Offset: 0x40d7
Byte Offset: 0x1035c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc5**
4	0x0	**err_intr_mask_ph_single_crc_err_vc5**
3	0x0	**err_intr_mask_pd_wc_short_err_vc5**
2	0x0	**err_intr_mask_pd_crc_err_vc5**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc5**
0	0x0	**err_intr_mask_ppfsm_timeout_vc5**

NVCSI_STREAM_0_ERR_INTR_MASK_VC6_0

Offset: 0x40d8
Byte Offset: 0x10360
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc6**
4	0x0	**err_intr_mask_ph_single_crc_err_vc6**
3	0x0	**err_intr_mask_pd_wc_short_err_vc6**
2	0x0	**err_intr_mask_pd_crc_err_vc6**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc6**
0	0x0	**err_intr_mask_ppfsm_timeout_vc6**

NVCSI_STREAM_0_ERR_INTR_MASK_VC7_0

Offset: 0x40d9
Byte Offset: 0x10364
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc7**
4	0x0	**err_intr_mask_ph_single_crc_err_vc7**
3	0x0	**err_intr_mask_pd_wc_short_err_vc7**
2	0x0	**err_intr_mask_pd_crc_err_vc7**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc7**
0	0x0	**err_intr_mask_ppfsm_timeout_vc7**

NVCSI_STREAM_0_ERR_INTR_MASK_VC8_0

Offset: 0x40da
Byte Offset: 0x10368
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc8**
4	0x0	**err_intr_mask_ph_single_crc_err_vc8**
3	0x0	**err_intr_mask_pd_wc_short_err_vc8**
2	0x0	**err_intr_mask_pd_crc_err_vc8**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc8**
0	0x0	**err_intr_mask_ppfsm_timeout_vc8**

NVCSI_STREAM_0_ERR_INTR_MASK_VC9_0

Offset: 0x40db
Byte Offset: 0x1036c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc9**
4	0x0	**err_intr_mask_ph_single_crc_err_vc9**
3	0x0	**err_intr_mask_pd_wc_short_err_vc9**
2	0x0	**err_intr_mask_pd_crc_err_vc9**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc9**
0	0x0	**err_intr_mask_ppfsm_timeout_vc9**

NVCSI_STREAM_0_ERR_INTR_MASK_VC10_0

Offset: 0x40dc
Byte Offset: 0x10370
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc10**
4	0x0	**err_intr_mask_ph_single_crc_err_vc10**
3	0x0	**err_intr_mask_pd_wc_short_err_vc10**
2	0x0	**err_intr_mask_pd_crc_err_vc10**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc10**
0	0x0	**err_intr_mask_ppfsm_timeout_vc10**

NVCSI_STREAM_0_ERR_INTR_MASK_VC11_0

Offset: 0x40dd
Byte Offset: 0x10374
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc11**
4	0x0	**err_intr_mask_ph_single_crc_err_vc11**
3	0x0	**err_intr_mask_pd_wc_short_err_vc11**
2	0x0	**err_intr_mask_pd_crc_err_vc11**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc11**
0	0x0	**err_intr_mask_ppfsm_timeout_vc11**

NVCSI_STREAM_0_ERR_INTR_MASK_VC12_0

Offset: 0x40de
Byte Offset: 0x10378
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc12**
4	0x0	**err_intr_mask_ph_single_crc_err_vc12**
3	0x0	**err_intr_mask_pd_wc_short_err_vc12**
2	0x0	**err_intr_mask_pd_crc_err_vc12**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc12**
0	0x0	**err_intr_mask_ppfsm_timeout_vc12**

NVCSI_STREAM_0_ERR_INTR_MASK_VC13_0

Offset: 0x40df
Byte Offset: 0x1037c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc13**
4	0x0	**err_intr_mask_ph_single_crc_err_vc13**
3	0x0	**err_intr_mask_pd_wc_short_err_vc13**
2	0x0	**err_intr_mask_pd_crc_err_vc13**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc13**
0	0x0	**err_intr_mask_ppfsm_timeout_vc13**

NVCSI_STREAM_0_ERR_INTR_MASK_VC14_0

Offset: 0x40e0
Byte Offset: 0x10380
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc14**
4	0x0	**err_intr_mask_ph_single_crc_err_vc14**
3	0x0	**err_intr_mask_pd_wc_short_err_vc14**
2	0x0	**err_intr_mask_pd_crc_err_vc14**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc14**
0	0x0	**err_intr_mask_ppfsm_timeout_vc14**

NVCSI_STREAM_0_ERR_INTR_MASK_VC15_0

Offset: 0x40e1
Byte Offset: 0x10384
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc15**
4	0x0	**err_intr_mask_ph_single_crc_err_vc15**
3	0x0	**err_intr_mask_pd_wc_short_err_vc15**
2	0x0	**err_intr_mask_pd_crc_err_vc15**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc15**
0	0x0	**err_intr_mask_ppfsm_timeout_vc15**

NVCSI_STREAM_0_ERR_INTR_TYPE_NOVC_0

Offset: 0x40e2
Byte Offset: 0x10388
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**err_intr_type_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**err_intr_type_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_0_ERR_INTR_TYPE_VC0_0

Offset: 0x40e3
Byte Offset: 0x1038c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc0**
4	0x0	**err_intr_type_ph_single_crc_err_vc0**
3	0x0	**err_intr_type_pd_wc_short_err_vc0**
2	0x0	**err_intr_type_pd_crc_err_vc0**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc0**
0	0x0	**err_intr_type_ppfsm_timeout_vc0**

NVCSI_STREAM_0_ERR_INTR_TYPE_VC1_0

Offset: 0x40e4
Byte Offset: 0x10390
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc1**
4	0x0	**err_intr_type_ph_single_crc_err_vc1**

Bit	Reset	Description
3	0x0	**err_intr_type_pd_wc_short_err_vc1**
2	0x0	**err_intr_type_pd_crc_err_vc1**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc1**
0	0x0	**err_intr_type_ppfsm_timeout_vc1**

NVCSI_STREAM_0_ERR_INTR_TYPE_VC2_0

Offset: 0x40e5
Byte Offset: 0x10394
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc2**
4	0x0	**err_intr_type_ph_single_crc_err_vc2**
3	0x0	**err_intr_type_pd_wc_short_err_vc2**
2	0x0	**err_intr_type_pd_crc_err_vc2**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc2**
0	0x0	**err_intr_type_ppfsm_timeout_vc2**

NVCSI_STREAM_0_ERR_INTR_TYPE_VC3_0

Offset: 0x40e6
Byte Offset: 0x10398
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc3**

Bit	Reset	Description
4	0x0	**err_intr_type_ph_single_crc_err_vc3**
3	0x0	**err_intr_type_pd_wc_short_err_vc3**
2	0x0	**err_intr_type_pd_crc_err_vc3**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc3**
0	0x0	**err_intr_type_ppfsm_timeout_vc3**

NVCSI_STREAM_0_ERR_INTR_TYPE_VC4_0

Offset: 0x40e7
Byte Offset: 0x1039c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc4**
4	0x0	**err_intr_type_ph_single_crc_err_vc4**
3	0x0	**err_intr_type_pd_wc_short_err_vc4**
2	0x0	**err_intr_type_pd_crc_err_vc4**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc4**
0	0x0	**err_intr_type_ppfsm_timeout_vc4**

NVCSI_STREAM_0_ERR_INTR_TYPE_VC5_0

Offset: 0x40e8
Byte Offset: 0x103a0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc5**
4	0x0	**err_intr_type_ph_single_crc_err_vc5**
3	0x0	**err_intr_type_pd_wc_short_err_vc5**
2	0x0	**err_intr_type_pd_crc_err_vc5**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc5**
0	0x0	**err_intr_type_ppfsm_timeout_vc5**

NVCSI_STREAM_0_ERR_INTR_TYPE_VC6_0

Offset: 0x40e9
Byte Offset: 0x103a4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc6**
4	0x0	**err_intr_type_ph_single_crc_err_vc6**
3	0x0	**err_intr_type_pd_wc_short_err_vc6**
2	0x0	**err_intr_type_pd_crc_err_vc6**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc6**
0	0x0	**err_intr_type_ppfsm_timeout_vc6**

NVCSI_STREAM_0_ERR_INTR_TYPE_VC7_0

Offset: 0x40ea
Byte Offset: 0x103a8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc7**
4	0x0	**err_intr_type_ph_single_crc_err_vc7**
3	0x0	**err_intr_type_pd_wc_short_err_vc7**
2	0x0	**err_intr_type_pd_crc_err_vc7**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc7**
0	0x0	**err_intr_type_ppfsm_timeout_vc7**

NVCSI_STREAM_0_ERR_INTR_TYPE_VC8_0

Offset: 0x40eb
Byte Offset: 0x103ac
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc8**
4	0x0	**err_intr_type_ph_single_crc_err_vc8**
3	0x0	**err_intr_type_pd_wc_short_err_vc8**
2	0x0	**err_intr_type_pd_crc_err_vc8**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc8**
0	0x0	**err_intr_type_ppfsm_timeout_vc8**

NVCSI_STREAM_0_ERR_INTR_TYPE_VC9_0

Offset: 0x40ec
Byte Offset: 0x103b0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc9**
4	0x0	**err_intr_type_ph_single_crc_err_vc9**
3	0x0	**err_intr_type_pd_wc_short_err_vc9**
2	0x0	**err_intr_type_pd_crc_err_vc9**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc9**
0	0x0	**err_intr_type_ppfsm_timeout_vc9**

NVCSI_STREAM_0_ERR_INTR_TYPE_VC10_0

Offset: 0x40ed
Byte Offset: 0x103b4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc10**
4	0x0	**err_intr_type_ph_single_crc_err_vc10**
3	0x0	**err_intr_type_pd_wc_short_err_vc10**
2	0x0	**err_intr_type_pd_crc_err_vc10**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc10**
0	0x0	**err_intr_type_ppfsm_timeout_vc10**

NVCSI_STREAM_0_ERR_INTR_TYPE_VC11_0

Offset: 0x40ee
Byte Offset: 0x103b8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc11**
4	0x0	**err_intr_type_ph_single_crc_err_vc11**
3	0x0	**err_intr_type_pd_wc_short_err_vc11**
2	0x0	**err_intr_type_pd_crc_err_vc11**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc11**
0	0x0	**err_intr_type_ppfsm_timeout_vc11**

NVCSI_STREAM_0_ERR_INTR_TYPE_VC12_0

Offset: 0x40ef
Byte Offset: 0x103bc
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc12**
4	0x0	**err_intr_type_ph_single_crc_err_vc12**
3	0x0	**err_intr_type_pd_wc_short_err_vc12**
2	0x0	**err_intr_type_pd_crc_err_vc12**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc12**
0	0x0	**err_intr_type_ppfsm_timeout_vc12**

NVCSI_STREAM_0_ERR_INTR_TYPE_VC13_0

Offset: 0x40f0
Byte Offset: 0x103c0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc13**
4	0x0	**err_intr_type_ph_single_crc_err_vc13**
3	0x0	**err_intr_type_pd_wc_short_err_vc13**
2	0x0	**err_intr_type_pd_crc_err_vc13**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc13**
0	0x0	**err_intr_type_ppfsm_timeout_vc13**

NVCSI_STREAM_0_ERR_INTR_TYPE_VC14_0

Offset: 0x40f1
Byte Offset: 0x103c4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc14**
4	0x0	**err_intr_type_ph_single_crc_err_vc14**
3	0x0	**err_intr_type_pd_wc_short_err_vc14**
2	0x0	**err_intr_type_pd_crc_err_vc14**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc14**
0	0x0	**err_intr_type_ppfsm_timeout_vc14**

NVCSI_STREAM_0_ERR_INTR_TYPE_VC15_0

Offset: 0x40f2
Byte Offset: 0x103c8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc15**
4	0x0	**err_intr_type_ph_single_crc_err_vc15**
3	0x0	**err_intr_type_pd_wc_short_err_vc15**
2	0x0	**err_intr_type_pd_crc_err_vc15**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc15**
0	0x0	**err_intr_type_ppfsm_timeout_vc15**

NVCSI_STREAM_0_TPG_ENABLE_0

Offset: 0x40f3
Byte Offset: 0x103cc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	DISABLE	**TPG_ENABLE**: Enable the TPG path **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_0_TPG_VC_ENABLE_0

Offset: 0x40f4
Byte Offset: 0x103d0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15	DISABLE	**VC15_EN**: Enable bit for the VC15 **0** = DISABLE **1** = ENABLE
14	DISABLE	**VC14_EN**: Enable bit for the VC14 **0** = DISABLE **1** = ENABLE
13	DISABLE	**VC13_EN**: Enable bit for the VC13 **0** = DISABLE **1** = ENABLE
12	DISABLE	**VC12_EN**: Enable bit for the VC12 **0** = DISABLE **1** = ENABLE
11	DISABLE	**VC11_EN**: Enable bit for the VC11 **0** = DISABLE **1** = ENABLE
10	DISABLE	**VC10_EN**: Enable bit for the VC10 **0** = DISABLE **1** = ENABLE
9	DISABLE	**VC9_EN**: Enable bit for the VC9 **0** = DISABLE **1** = ENABLE
8	DISABLE	**VC8_EN**: Enable bit for the VC8 **0** = DISABLE **1** = ENABLE
7	DISABLE	**VC7_EN**: Enable bit for the VC7 **0** = DISABLE **1** = ENABLE
6	DISABLE	**VC6_EN**: Enable bit for the VC6 **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
5	DISABLE	**VC5_EN**: Enable bit for the VC5 **0** = DISABLE **1** = ENABLE
4	DISABLE	**VC4_EN**: Enable bit for the VC4 **0** = DISABLE **1** = ENABLE
3	DISABLE	**VC3_EN**: Enable bit for the VC3 **0** = DISABLE **1** = ENABLE
2	DISABLE	**VC2_EN**: Enable bit for the VC2 **0** = DISABLE **1** = ENABLE
1	DISABLE	**VC1_EN**: Enable bit for the VC1 **0** = DISABLE **1** = ENABLE
0	DISABLE	**VC0_EN**: Enable bit for the VC0 **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_0_LOAD_TPG_CFG_0

Offset: 0x40f5
Byte Offset: 0x103d4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**LOAD**: Load the shadow register

NVCSI_STREAM_0_TPG_CTRL_0

Offset: 0x40f6
Byte Offset: 0x103d8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000008a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1000,1010)

Bit	Reset	Description
7:4	0x8	**DATA_SPEED**: Control the data generation speed, valid range is 1~8.
3	ENABLE	**SKIP_LS_LE_PKT**: If the LS/LE packet need to generated. **0** = DISABLE **1** = ENABLE
2	DISABLE	**OVERRIDE_CRC**: Override the packet header CRC and payload CRC. **0** = DISABLE **1** = ENABLE
1	CORE	**DEST**: The TPG pattern is send to PP or send to CIL for TX. **0** = CIL **1** = CORE
0	DPHY	**PHY_MODE**: CPHY or DPHY packet structure for TPG **0** = DPHY **1** = CPHY

NVCSI_STREAM_0_TPG_VBLANK_0

Offset: 0x40f7
Byte Offset: 0x103dc
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000800 (0b0000,0000,0000,0000,0000,1000,0000,0000)

Bit	Reset	Description
31:0	0x800	**VBLANK**: The vblank value, unit is nvcsi clock cycles.

NVCSI_STREAM_0_TPG_HBLANK_0

Offset: 0x40f8
Byte Offset: 0x103e0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000800 (0b0000,0000,0000,0000,0000,1000,0000,0000)

Bit	Reset	Description
31:0	0x800	**HBLANK**: The hblank value, unit is nvcsi clock cycles.

NVCSI_STREAM_0_TPG_STATUS_0

Offset: 0x40f9
Byte Offset: 0x103e4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0001ffff (0bxxxx,xxxx,xxxx,xxx1,1111,1111,1111,1111)

Bit	Reset	Description
16	IDLE	**VC15_STATUS**: Indicate the VC15 status **0** = BUSY **1** = IDLE
15	IDLE	**VC14_STATUS**: Indicate the VC14 status **0** = BUSY **1** = IDLE
14	IDLE	**VC13_STATUS**: Indicate the VC13 status **0** = BUSY **1** = IDLE
13	IDLE	**VC12_STATUS**: Indicate the VC12 status **0** = BUSY **1** = IDLE

Bit	Reset	Description
12	IDLE	**VC11_STATUS**: Indicate the VC11 status **0** = BUSY **1** = IDLE
11	IDLE	**VC10_STATUS**: Indicate the VC10 status **0** = BUSY **1** = IDLE
10	IDLE	**VC9_STATUS**: Indicate the VC9 status **0** = BUSY **1** = IDLE
9	IDLE	**VC8_STATUS**: Indicate the VC8 status **0** = BUSY **1** = IDLE
8	IDLE	**VC7_STATUS**: Indicate the VC7 status **0** = BUSY **1** = IDLE
7	IDLE	**VC6_STATUS**: Indicate the VC6 status **0** = BUSY **1** = IDLE
6	IDLE	**VC5_STATUS**: Indicate the VC5 status **0** = BUSY **1** = IDLE
5	IDLE	**VC4_STATUS**: Indicate the VC4 status **0** = BUSY **1** = IDLE
4	IDLE	**VC3_STATUS**: Indicate the VC3 status **0** = BUSY **1** = IDLE
3	IDLE	**VC2_STATUS**: Indicate the VC2 status **0** = BUSY **1** = IDLE

Bit	Reset	Description
2	IDLE	**VC1_STATUS**: Indicate the VC1 status **0** = BUSY **1** = IDLE
1	IDLE	**VC0_STATUS**: Indicate the VC0 status **0** = BUSY **1** = IDLE
0	IDLE	**STATUS**: Indicate the TPG is in idle state, all packet has been send and all VC are disabled. **0** = BUSY **1** = IDLE

NVCSI_STREAM_0_TPG_PH_ECC_0

Offset: 0x40fa
Byte Offset: 0x103e8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:24	0x0	**LINE_ECC**: This field is for the long packet header ECC.
23:18	0x0	**EOL_ECC**: This field is for the EOL short packet ECC.
17:12	0x0	**SOL_ECC**: This field is for the SOL short packet ECC.
11:6	0x0	**EOF_ECC**: This field is for the EOF short packet ECC.
5:0	0x0	**SOF_ECC**: The TPG will not generate ECC for a packet. When using the TPG, SW should set the PP to skip the ecc check. To verify the ecc logic for safety BIST, SW can write a pre-calculated ECC for the TPG, when use with this mode, the TPG should generate a grescale pattern. This field is for the SOF short packet ECC.

NVCSI_STREAM_0_TPG_PF_CRC_0

Offset: 0x40fb
Byte Offset: 0x103ec
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PF_CRC**: This field is for the long packet payload CRC override.

NVCSI_STREAM_0_TPG_PH_SOF_CRC_0

Offset: 0x40fc
Byte Offset: 0x103f0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This field is for the CPHY SOF packet second packet header CRC override.
15:0	0x0	**CRC_0**: This field is for the CPHY SOF packet first packet header CRC override.

NVCSI_STREAM_0_TPG_PH_EOF_CRC_0

Offset: 0x40fd
Byte Offset: 0x103f4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This field is for the CPHY EOF packet second packet header CRC override.
15:0	0x0	**CRC_0**: This field is for the CPHY EOF packet first packet header CRC override.

NVCSI_STREAM_0_TPG_PH_SOL_CRC_0

Offset: 0x40fe
Byte Offset: 0x103f8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This field is for the CPHY SOL packet second packet header CRC override.
15:0	0x0	**CRC_0**: This field is for the CPHY SOL packet first packet header CRC override.

NVCSI_STREAM_0_TPG_PH_EOL_CRC_0

Offset: 0x40ff
Byte Offset: 0x103fc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This field is for the CPHY EOL packet second packet header CRC override.
15:0	0x0	**CRC_0**: This field is for the CPHY EOL packet first packet header CRC override.

NVCSI_STREAM_0_TPG_PH_LONG_PKT_CRC_0

Offset: 0x4100
Byte Offset: 0x10400
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This filed is for the CPHY long packet second packet header CRC override.
15:0	0x0	**CRC_0**: This filed is for the CPHY long packet first packet header CRC override.

NVCSI_STREAM_0_TPG_PKT_DELIMETER_0

Offset: 0x4101
Byte Offset: 0x10404
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0011)

Bit	Reset	Description
15:0	0x3	**NUM**: This field define the cycle number between two packet. Model the LP11 period between two packet, valid range is 1 to 65535.

NVCSI_STREAM_0_VC0_TPG_GAIN_CTRL_0

Offset: 0x4102
Byte Offset: 0x10408
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_0_VC1_TPG_GAIN_CTRL_0

Offset: 0x4103
Byte Offset: 0x1040c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_0_VC2_TPG_GAIN_CTRL_0

Offset: 0x4104
Byte Offset: 0x10410
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_0_VC3_TPG_GAIN_CTRL_0

Offset: 0x4105
Byte Offset: 0x10414
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_0_VC4_TPG_GAIN_CTRL_0

Offset: 0x4106
Byte Offset: 0x10418
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_0_VC5_TPG_GAIN_CTRL_0

Offset: 0x4107
Byte Offset: 0x1041c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_0_VC6_TPG_GAIN_CTRL_0

Offset: 0x4108
Byte Offset: 0x10420
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_0_VC7_TPG_GAIN_CTRL_0

Offset: 0x4109
Byte Offset: 0x10424
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_0_VC8_TPG_GAIN_CTRL_0

Offset: 0x410a
Byte Offset: 0x10428
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_0_VC9_TPG_GAIN_CTRL_0

Offset: 0x410b
Byte Offset: 0x1042c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_0_VC10_TPG_GAIN_CTRL_0

Offset: 0x410c
Byte Offset: 0x10430
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_0_VC11_TPG_GAIN_CTRL_0

Offset: 0x410d
Byte Offset: 0x10434
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_0_VC12_TPG_GAIN_CTRL_0

Offset: 0x410e
Byte Offset: 0x10438
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_0_VC13_TPG_GAIN_CTRL_0

Offset: 0x410f
Byte Offset: 0x1043c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_0_VC14_TPG_GAIN_CTRL_0

Offset: 0x4110
Byte Offset: 0x10440
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_0_VC15_TPG_GAIN_CTRL_0

Offset: 0x4111
Byte Offset: 0x10444
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_0_SPARE_0

Offset: 0x4112
Byte Offset: 0x10448
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**stream_reg**

7.2.1.3.2 NVCSI Stream1 Registers

NVCSI_STREAM_1_SW_RESET_CTRL_0

Offset: 0x6000
Byte Offset: 0x18000
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DISABLE	**cfg_swreset**: Reset the pixel parser. **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_1_SLCG_CTRL_0

Offset: 0x6001
Byte Offset: 0x18004
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DISABLE	**cfg_slcg_override**: Enable the SLCG override for the pixel parser. **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_1_PP_PHY_CTRL_0

Offset: 0x6002
Byte Offset: 0x18008
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DPHY	**cfg_phy_mode**: **0** = DPHY **1** = CPHY

NVCSI_STREAM_1_PP_EN_CTRL_0

Offset: 0x6003
Byte Offset: 0x1800c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	CIL	**cfg_src**: The pixel source of the pixel parser **0** = CIL **1** = TPG
0	DISABLE	**cfg_pp_en**: Pixel Parser streaming enable **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_1_VC0_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x6004
Byte Offset: 0x18010
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc0_dt_nooverride_0**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC0_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x6005
Byte Offset: 0x18014
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc0_dt_nooverride_1**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC0_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x6006
Byte Offset: 0x18018
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc0_dt_nooverride_2**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC0_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x6007
Byte Offset: 0x1801c
Read/Write: R/W
Parity Protection: N
Shadow: Y

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc0_dt_nooverride_3**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC0_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x6008
Byte Offset: 0x18020
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc0_dt_nooverride_4**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC0_DT_OVERRIDE_0

Offset: 0x6009
Byte Offset: 0x18024
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc0_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc0_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc0_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_1_VC1_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x600a
Byte Offset: 0x18028
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc1_dt_nooverride_0**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC1_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x600b
Byte Offset: 0x1802c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc1_dt_nooverride_1**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC1_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x600c
Byte Offset: 0x18030
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc1_dt_nooverride_2**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC1_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x600d
Byte Offset: 0x18034
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc1_dt_nooverride_3**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC1_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x600e
Byte Offset: 0x18038
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc1_dt_nooverride_4**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC1_DT_OVERRIDE_0

Offset: 0x600f
Byte Offset: 0x1803c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc1_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc1_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc1_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_1_VC2_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x6010
Byte Offset: 0x18040
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc2_dt_nooverride_0**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC2_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x6011
Byte Offset: 0x18044
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc2_dt_nooverride_1**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC2_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x6012
Byte Offset: 0x18048
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc2_dt_nooverride_2**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC2_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x6013
Byte Offset: 0x1804c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc2_dt_nooverride_3**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC2_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x6014
Byte Offset: 0x18050
Read/Write: R/W
Parity Protection: N
Shadow: Y

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc2_dt_nooverride_4**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC2_DT_OVERRIDE_0

Offset: 0x6015
Byte Offset: 0x18054
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc2_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc2_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc2_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_1_VC3_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x6016
Byte Offset: 0x18058
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc3_dt_nooverride_0**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC3_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x6017
Byte Offset: 0x1805c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc3_dt_nooverride_1**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC3_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x6018
Byte Offset: 0x18060
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc3_dt_nooverride_2**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC3_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x6019
Byte Offset: 0x18064
Read/Write: R/W
Parity Protection: N
Shadow: Y

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc3_dt_nooverride_3**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC3_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x601a
Byte Offset: 0x18068
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc3_dt_nooverride_4**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC3_DT_OVERRIDE_0

Offset: 0x601b
Byte Offset: 0x1806c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc3_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc3_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc3_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_1_VC4_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x601c
Byte Offset: 0x18070
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc4_dt_nooverride_0**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC4_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x601d
Byte Offset: 0x18074
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc4_dt_nooverride_1**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC4_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x601e
Byte Offset: 0x18078
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc4_dt_nooverride_2**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC4_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x601f
Byte Offset: 0x1807c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc4_dt_nooverride_3**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC4_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x6020
Byte Offset: 0x18080
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc4_dt_nooverride_4**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC4_DT_OVERRIDE_0

Offset: 0x6021
Byte Offset: 0x18084
Read/Write: R/W
Parity Protection: Y
Shadow: Y

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc4_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc4_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc4_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_1_VC5_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x6022
Byte Offset: 0x18088
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc5_dt_nooverride_0**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC5_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x6023
Byte Offset: 0x1808c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc5_dt_nooverride_1**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC5_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x6024
Byte Offset: 0x18090
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc5_dt_nooverride_2**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC5_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x6025
Byte Offset: 0x18094
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc5_dt_nooverride_3**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC5_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x6026
Byte Offset: 0x18098
Read/Write: R/W
Parity Protection: N
Shadow: Y

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc5_dt_nooverride_4**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC5_DT_OVERRIDE_0

Offset: 0x6027
Byte Offset: 0x1809c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc5_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc5_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc5_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_1_VC6_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x6028
Byte Offset: 0x180a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc6_dt_nooverride_0**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC6_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x6029
Byte Offset: 0x180a4
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc6_dt_nooverride_1**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC6_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x602a
Byte Offset: 0x180a8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc6_dt_nooverride_2**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC6_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x602b
Byte Offset: 0x180ac
Read/Write: R/W
Parity Protection: N
Shadow: Y

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc6_dt_nooverride_3**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC6_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x602c
Byte Offset: 0x180b0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc6_dt_nooverride_4**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC6_DT_OVERRIDE_0

Offset: 0x602d
Byte Offset: 0x180b4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc6_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc6_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc6_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_1_VC7_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x602e
Byte Offset: 0x180b8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc7_dt_nooverride_0**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC7_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x602f
Byte Offset: 0x180bc
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc7_dt_nooverride_1**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC7_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x6030
Byte Offset: 0x180c0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc7_dt_nooverride_2**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC7_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x6031
Byte Offset: 0x180c4
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc7_dt_nooverride_3**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC7_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x6032
Byte Offset: 0x180c8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc7_dt_nooverride_4**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC7_DT_OVERRIDE_0

Offset: 0x6033
Byte Offset: 0x180cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc7_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc7_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc7_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_1_VC8_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x6034
Byte Offset: 0x180d0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc8_dt_nooverride_0**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC8_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x6035
Byte Offset: 0x180d4
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc8_dt_nooverride_1**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC8_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x6036
Byte Offset: 0x180d8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc8_dt_nooverride_2**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC8_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x6037
Byte Offset: 0x180dc
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc8_dt_nooverride_3**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC8_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x6038
Byte Offset: 0x180e0
Read/Write: R/W
Parity Protection: N
Shadow: Y

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc8_dt_nooverride_4**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC8_DT_OVERRIDE_0

Offset: 0x6039
Byte Offset: 0x180e4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc8_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc8_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc8_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_1_VC9_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x603a
Byte Offset: 0x180e8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc9_dt_nooverride_0**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC9_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x603b
Byte Offset: 0x180ec
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc9_dt_nooverride_1**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC9_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x603c
Byte Offset: 0x180f0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc9_dt_nooverride_2**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC9_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x603d
Byte Offset: 0x180f4
Read/Write: R/W
Parity Protection: N
Shadow: Y

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc9_dt_nooverride_3**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC9_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x603e
Byte Offset: 0x180f8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc9_dt_nooverride_4**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC9_DT_OVERRIDE_0

Offset: 0x603f
Byte Offset: 0x180fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc9_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc9_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc9_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_1_VC10_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x6040
Byte Offset: 0x18100
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc10_dt_nooverride_0**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC10_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x6041
Byte Offset: 0x18104
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc10_dt_nooverride_1**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC10_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x6042
Byte Offset: 0x18108
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc10_dt_nooverride_2**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC10_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x6043
Byte Offset: 0x1810c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc10_dt_nooverride_3**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC10_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x6044
Byte Offset: 0x18110
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc10_dt_nooverride_4**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC10_DT_OVERRIDE_0

Offset: 0x6045
Byte Offset: 0x18114
Read/Write: R/W
Parity Protection: Y
Shadow: Y

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc10_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc10_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc10_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_1_VC11_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x6046
Byte Offset: 0x18118
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc11_dt_nooverride_0**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC11_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x6047
Byte Offset: 0x1811c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc11_dt_nooverride_1**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC11_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x6048
Byte Offset: 0x18120
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc11_dt_nooverride_2**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC11_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x6049
Byte Offset: 0x18124
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc11_dt_nooverride_3**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC11_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x604a
Byte Offset: 0x18128
Read/Write: R/W
Parity Protection: N
Shadow: Y

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc11_dt_nooverride_4**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC11_DT_OVERRIDE_0

Offset: 0x604b
Byte Offset: 0x1812c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc11_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc11_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc11_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_1_VC12_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x604c
Byte Offset: 0x18130
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc12_dt_nooverride_0**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC12_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x604d
Byte Offset: 0x18134
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc12_dt_nooverride_1**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC12_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x604e
Byte Offset: 0x18138
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc12_dt_nooverride_2**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC12_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x604f
Byte Offset: 0x1813c
Read/Write: R/W
Parity Protection: N
Shadow: Y

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc12_dt_nooverride_3**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC12_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x6050
Byte Offset: 0x18140
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc12_dt_nooverride_4**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC12_DT_OVERRIDE_0

Offset: 0x6051
Byte Offset: 0x18144
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc12_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc12_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc12_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_1_VC13_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x6052
Byte Offset: 0x18148
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc13_dt_nooverride_0**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC13_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x6053
Byte Offset: 0x1814c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc13_dt_nooverride_1**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC13_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x6054
Byte Offset: 0x18150
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc13_dt_nooverride_2**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC13_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x6055
Byte Offset: 0x18154
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc13_dt_nooverride_3**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC13_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x6056
Byte Offset: 0x18158
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc13_dt_nooverride_4**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC13_DT_OVERRIDE_0

Offset: 0x6057
Byte Offset: 0x1815c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc13_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc13_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc13_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_1_VC14_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x6058
Byte Offset: 0x18160
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc14_dt_nooverride_0**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC14_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x6059
Byte Offset: 0x18164
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc14_dt_nooverride_1**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC14_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x605a
Byte Offset: 0x18168
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc14_dt_nooverride_2**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC14_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x605b
Byte Offset: 0x1816c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc14_dt_nooverride_3**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC14_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x605c
Byte Offset: 0x18170
Read/Write: R/W
Parity Protection: N
Shadow: Y

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc14_dt_nooverride_4**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC14_DT_OVERRIDE_0

Offset: 0x605d
Byte Offset: 0x18174
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc14_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc14_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc14_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_1_VC15_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x605e
Byte Offset: 0x18178
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc15_dt_nooverride_0**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC15_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x605f
Byte Offset: 0x1817c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc15_dt_nooverride_1**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC15_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x6060
Byte Offset: 0x18180
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc15_dt_nooverride_2**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC15_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x6061
Byte Offset: 0x18184
Read/Write: R/W
Parity Protection: N
Shadow: Y

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc15_dt_nooverride_3**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC15_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x6062
Byte Offset: 0x18188
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc15_dt_nooverride_4**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_1_VC15_DT_OVERRIDE_0

Offset: 0x6063
Byte Offset: 0x1818c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc15_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc15_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc15_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_1_PPFSM_TIMEOUT_CTRL_0

Offset: 0x6064
Byte Offset: 0x18190
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x7fffffff (0b0111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31	DISABLE	**cfg_timeout_en**: Enable Timeout counter for the PP FSM **0** = DISABLE **1** = ENABLE
30:0	0x7fffffff	**cfg_timeout_period**: Timeout period

NVCSI_STREAM_1_PH_CHK_CTRL_0

Offset: 0x6065
Byte Offset: 0x18194
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	ENABLE	**cfg_ph_16_vc**: 16 VC support **0** = DISABLE **1** = ENABLE
1	ENABLE	**cfg_ph_crc_chk_en**: PH CRC check enable (only for CPHY case), when this bit is set to 0, the packet header will still be decode when the CRC check fail, but the error will be set. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
0	ENABLE	**cfg_ph_ecc_chk_en**: PH ECC check enable (only for DPHY case), when this bit is set to 0, the packet header will still be decode when the ECC check fail, but the error will be set. **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_1_VC0_DPCM_CTRL_0

Offset: 0x6066
Byte Offset: 0x18198
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc0_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_1_VC1_DPCM_CTRL_0

Offset: 0x6067
Byte Offset: 0x1819c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc1_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_1_VC2_DPCM_CTRL_0

Offset: 0x6068
Byte Offset: 0x181a0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc2_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_1_VC3_DPCM_CTRL_0

Offset: 0x6069
Byte Offset: 0x181a4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc3_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_1_VC4_DPCM_CTRL_0

Offset: 0x606a
Byte Offset: 0x181a8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc4_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_1_VC5_DPCM_CTRL_0

Offset: 0x606b
Byte Offset: 0x181ac
Read/Write: R/W
Parity Protection: Y
Shadow: N

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc5_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_1_VC6_DPCM_CTRL_0

Offset: 0x606c
Byte Offset: 0x181b0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc6_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_1_VC7_DPCM_CTRL_0

Offset: 0x606d
Byte Offset: 0x181b4
Read/Write: R/W
Parity Protection: Y

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc7_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_1_VC8_DPCM_CTRL_0

Offset: 0x606e
Byte Offset: 0x181b8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc8_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_1_VC9_DPCM_CTRL_0

Offset: 0x606f
Byte Offset: 0x181bc
Read/Write: R/W

Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc9_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_1_VC10_DPCM_CTRL_0

Offset: 0x6070
Byte Offset: 0x181c0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc10_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_1_VC11_DPCM_CTRL_0

Offset: 0x6071
Byte Offset: 0x181c4

Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc11_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_1_VC12_DPCM_CTRL_0

Offset: 0x6072
Byte Offset: 0x181c8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc12_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_1_VC13_DPCM_CTRL_0

Offset: 0x6073

Byte Offset: 0x181cc
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc13_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_1_VC14_DPCM_CTRL_0

Offset: 0x6074
Byte Offset: 0x181d0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc14_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_1_VC15_DPCM_CTRL_0

Offset: 0x6075
Byte Offset: 0x181d4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc15_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_1_PF_CRC_0

Status register on packet data CRC
Offset: 0x6076
Byte Offset: 0x181d8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**calc_crc**: 16 bit CRC computed over current packet (has to match with the CRC in PF for a good packet)
15:0	0x0	**rx_crc**: 16 bit CRC from PF

NVCSI_STREAM_1_PH_WC_0

Status register on WC
Offset: 0x6077
Byte Offset: 0x181dc

Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**calc_wc**: 16 bit WC computed over current packet (has to match with the WC in PH for a good packet)
15:0	0x0	**rx_wc**: 16 bit WC from PH

NVCSI_STREAM_1_PH_DI_0

Status register on Data ID
Offset: 0x6078
Byte Offset: 0x181e0
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9:6	0x0	**rx_vc**: 4 bit VC from PH
5:0	0x0	**rx_dt**: 6 bit DTYPE from PH

NVCSI_STREAM_1_ERROR_STATUS2VI_MASK_0

Offset: 0x6079
Byte Offset: 0x181e4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15	0x0	**cfg_err_status2vi_mask_vc15**: for VC15 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
14	0x0	**cfg_err_status2vi_mask_vc14**: for VC14 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
13	0x0	**cfg_err_status2vi_mask_vc13**: for VC13 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
12	0x0	**cfg_err_status2vi_mask_vc12**: for VC12 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
11	0x0	**cfg_err_status2vi_mask_vc11**: for VC11 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
10	0x0	**cfg_err_status2vi_mask_vc10**: for VC10 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
9	0x0	**cfg_err_status2vi_mask_vc9**: for VC9 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
8	0x0	**cfg_err_status2vi_mask_vc8**: for VC8 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
7	0x0	**cfg_err_status2vi_mask_vc7**: for VC7 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
6	0x0	**cfg_err_status2vi_mask_vc6**: for VC6 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
5	0x0	**cfg_err_status2vi_mask_vc5**: for VC5 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
4	0x0	**cfg_err_status2vi_mask_vc4**: for VC4 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
3	0x0	**cfg_err_status2vi_mask_vc3**: for VC3 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
2	0x0	**cfg_err_status2vi_mask_vc2**: for VC2 : 0 = err_status2vi_vc2 will be send to VI at EOF; 1 = No error will be send to VI at EOF
1	0x0	**cfg_err_status2vi_mask_vc1**: for VC1 : 0 = err_status2vi_vc1 will be send to VI at EOF; 1 = No error will be send to VI at EOF

Bit	Reset	Description
0	0x0	**cfg_err_status2vi_mask_vc0**: for VC0 : 0 = err_status2vi_vc0 will be send to VI at EOF; 1 = No error will be send to VI at EOF

NVCSI_STREAM_1_ERROR_STATUS2VI_VC0_0

This is RO register for SW
Offset: 0x607a
Byte Offset: 0x181e8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc0**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_1_ERROR_STATUS2VI_VC1_0

This is RO register for SW
Offset: 0x607b
Byte Offset: 0x181ec
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc1**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_1_ERROR_STATUS2VI_VC2_0

This is RO register for SW
Offset: 0x607c
Byte Offset: 0x181f0
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc2**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_1_ERROR_STATUS2VI_VC3_0

This is RO register for SW
Offset: 0x607d
Byte Offset: 0x181f4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc3**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_1_ERROR_STATUS2VI_VC4_0

This is RO register for SW
Offset: 0x607e

Byte Offset: 0x181f8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc4**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_1_ERROR_STATUS2VI_VC5_0

This is RO register for SW
Offset: 0x607f
Byte Offset: 0x181fc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc5**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_1_ERROR_STATUS2VI_VC6_0

This is RO register for SW
Offset: 0x6080
Byte Offset: 0x18200
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc6**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_1_ERROR_STATUS2VI_VC7_0

This is RO register for SW
Offset: 0x6081
Byte Offset: 0x18204
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc7**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_1_ERROR_STATUS2VI_VC8_0

This is RO register for SW
Offset: 0x6082
Byte Offset: 0x18208
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc8**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_1_ERROR_STATUS2VI_VC9_0

This is RO register for SW
Offset: 0x6083
Byte Offset: 0x1820c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc9**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_1_ERROR_STATUS2VI_VC10_0

This is RO register for SW
Offset: 0x6084
Byte Offset: 0x18210
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc10**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_1_ERROR_STATUS2VI_VC11_0

This is RO register for SW
Offset: 0x6085
Byte Offset: 0x18214
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc11**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_1_ERROR_STATUS2VI_VC12_0

This is RO register for SW
Offset: 0x6086
Byte Offset: 0x18218
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc12**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_1_ERROR_STATUS2VI_VC13_0

This is RO register for SW
Offset: 0x6087
Byte Offset: 0x1821c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc13**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_1_ERROR_STATUS2VI_VC14_0

This is RO register for SW
Offset: 0x6088
Byte Offset: 0x18220
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc14**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_1_ERROR_STATUS2VI_VC15_0

This is RO register for SW
Offset: 0x6089
Byte Offset: 0x18224
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc15**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_1_INTR_STATUS_0

Offset: 0x608a
Byte Offset: 0x18228
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**intr_stat_novc**: Not VC related interrupt
15	0x0	**intr_stat_vc15**: VC15 event

Bit	Reset	Description
14	0x0	**intr_stat_vc14**: VC14 event
13	0x0	**intr_stat_vc13**: VC13 event
12	0x0	**intr_stat_vc12**: VC12 event
11	0x0	**intr_stat_vc11**: VC11 event
10	0x0	**intr_stat_vc10**: VC10 event
9	0x0	**intr_stat_vc9**: VC9 event
8	0x0	**intr_stat_vc8**: VC8 event
7	0x0	**intr_stat_vc7**: VC7 event
6	0x0	**intr_stat_vc6**: VC6 event
5	0x0	**intr_stat_vc5**: VC5 event
4	0x0	**intr_stat_vc4**: VC4 event
3	0x0	**intr_stat_vc3**: VC3 event
2	0x0	**intr_stat_vc2**: VC2 event
1	0x0	**intr_stat_vc1**: VC1 event
0	0x0	**intr_stat_vc0**: VC0 event

NVCSI_STREAM_1_INTR_STATUS_NOVC_0

Offset: 0x608b
Byte Offset: 0x1822c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**intr_stat_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**intr_stat_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_1_INTR_STATUS_VC0_0

Offset: 0x608c
Byte Offset: 0x18230
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc0**
4	0x0	**intr_stat_ph_single_crc_err_vc0**
3	0x0	**intr_stat_pd_wc_short_err_vc0**
2	0x0	**intr_stat_pd_crc_err_vc0**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc0**
0	0x0	**intr_stat_ppfsm_timeout_vc0**

NVCSI_STREAM_1_INTR_STATUS_VC1_0

Offset: 0x608d
Byte Offset: 0x18234
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc1**

Bit	Reset	Description
4	0x0	**intr_stat_ph_single_crc_err_vc1**
3	0x0	**intr_stat_pd_wc_short_err_vc1**
2	0x0	**intr_stat_pd_crc_err_vc1**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc1**
0	0x0	**intr_stat_ppfsm_timeout_vc1**

NVCSI_STREAM_1_INTR_STATUS_VC2_0

Offset: 0x608e
Byte Offset: 0x18238
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc2**
4	0x0	**intr_stat_ph_single_crc_err_vc2**
3	0x0	**intr_stat_pd_wc_short_err_vc2**
2	0x0	**intr_stat_pd_crc_err_vc2**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc2**
0	0x0	**intr_stat_ppfsm_timeout_vc2**

NVCSI_STREAM_1_INTR_STATUS_VC3_0

Offset: 0x608f
Byte Offset: 0x1823c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc3**
4	0x0	**intr_stat_ph_single_crc_err_vc3**
3	0x0	**intr_stat_pd_wc_short_err_vc3**
2	0x0	**intr_stat_pd_crc_err_vc3**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc3**
0	0x0	**intr_stat_ppfsm_timeout_vc3**

NVCSI_STREAM_1_INTR_STATUS_VC4_0

Offset: 0x6090
Byte Offset: 0x18240
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc4**
4	0x0	**intr_stat_ph_single_crc_err_vc4**
3	0x0	**intr_stat_pd_wc_short_err_vc4**
2	0x0	**intr_stat_pd_crc_err_vc4**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc4**
0	0x0	**intr_stat_ppfsm_timeout_vc4**

NVCSI_STREAM_1_INTR_STATUS_VC5_0

Offset: 0x6091
Byte Offset: 0x18244
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc5**
4	0x0	**intr_stat_ph_single_crc_err_vc5**
3	0x0	**intr_stat_pd_wc_short_err_vc5**
2	0x0	**intr_stat_pd_crc_err_vc5**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc5**
0	0x0	**intr_stat_ppfsm_timeout_vc5**

NVCSI_STREAM_1_INTR_STATUS_VC6_0

Offset: 0x6092
Byte Offset: 0x18248
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc6**
4	0x0	**intr_stat_ph_single_crc_err_vc6**
3	0x0	**intr_stat_pd_wc_short_err_vc6**
2	0x0	**intr_stat_pd_crc_err_vc6**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc6**
0	0x0	**intr_stat_ppfsm_timeout_vc6**

NVCSI_STREAM_1_INTR_STATUS_VC7_0

Offset: 0x6093
Byte Offset: 0x1824c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc7**
4	0x0	**intr_stat_ph_single_crc_err_vc7**
3	0x0	**intr_stat_pd_wc_short_err_vc7**
2	0x0	**intr_stat_pd_crc_err_vc7**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc7**
0	0x0	**intr_stat_ppfsm_timeout_vc7**

NVCSI_STREAM_1_INTR_STATUS_VC8_0

Offset: 0x6094
Byte Offset: 0x18250
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc8**
4	0x0	**intr_stat_ph_single_crc_err_vc8**
3	0x0	**intr_stat_pd_wc_short_err_vc8**
2	0x0	**intr_stat_pd_crc_err_vc8**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc8**
0	0x0	**intr_stat_ppfsm_timeout_vc8**

NVCSI_STREAM_1_INTR_STATUS_VC9_0

Offset: 0x6095
Byte Offset: 0x18254
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc9**
4	0x0	**intr_stat_ph_single_crc_err_vc9**
3	0x0	**intr_stat_pd_wc_short_err_vc9**
2	0x0	**intr_stat_pd_crc_err_vc9**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc9**
0	0x0	**intr_stat_ppfsm_timeout_vc9**

NVCSI_STREAM_1_INTR_STATUS_VC10_0

Offset: 0x6096
Byte Offset: 0x18258
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc10**
4	0x0	**intr_stat_ph_single_crc_err_vc10**
3	0x0	**intr_stat_pd_wc_short_err_vc10**
2	0x0	**intr_stat_pd_crc_err_vc10**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc10**
0	0x0	**intr_stat_ppfsm_timeout_vc10**

NVCSI_STREAM_1_INTR_STATUS_VC11_0

Offset: 0x6097
Byte Offset: 0x1825c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc11**
4	0x0	**intr_stat_ph_single_crc_err_vc11**
3	0x0	**intr_stat_pd_wc_short_err_vc11**
2	0x0	**intr_stat_pd_crc_err_vc11**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc11**
0	0x0	**intr_stat_ppfsm_timeout_vc11**

NVCSI_STREAM_1_INTR_STATUS_VC12_0

Offset: 0x6098
Byte Offset: 0x18260
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc12**
4	0x0	**intr_stat_ph_single_crc_err_vc12**
3	0x0	**intr_stat_pd_wc_short_err_vc12**
2	0x0	**intr_stat_pd_crc_err_vc12**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc12**
0	0x0	**intr_stat_ppfsm_timeout_vc12**

NVCSI_STREAM_1_INTR_STATUS_VC13_0

Offset: 0x6099
Byte Offset: 0x18264
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc13**
4	0x0	**intr_stat_ph_single_crc_err_vc13**
3	0x0	**intr_stat_pd_wc_short_err_vc13**
2	0x0	**intr_stat_pd_crc_err_vc13**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc13**
0	0x0	**intr_stat_ppfsm_timeout_vc13**

NVCSI_STREAM_1_INTR_STATUS_VC14_0

Offset: 0x609a
Byte Offset: 0x18268
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc14**
4	0x0	**intr_stat_ph_single_crc_err_vc14**
3	0x0	**intr_stat_pd_wc_short_err_vc14**
2	0x0	**intr_stat_pd_crc_err_vc14**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc14**
0	0x0	**intr_stat_ppfsm_timeout_vc14**

NVCSI_STREAM_1_INTR_STATUS_VC15_0

Offset: 0x609b
Byte Offset: 0x1826c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc15**
4	0x0	**intr_stat_ph_single_crc_err_vc15**
3	0x0	**intr_stat_pd_wc_short_err_vc15**
2	0x0	**intr_stat_pd_crc_err_vc15**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc15**
0	0x0	**intr_stat_ppfsm_timeout_vc15**

NVCSI_STREAM_1_INTR_MASK_NOVC_0

Offset: 0x609c
Byte Offset: 0x18270
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**intr_mask_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**intr_mask_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_1_INTR_MASK_VC0_0

Offset: 0x609d
Byte Offset: 0x18274
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc0**
4	0x0	**intr_mask_ph_single_crc_err_vc0**

Bit	Reset	Description
3	0x0	**intr_mask_pd_wc_short_err_vc0**
2	0x0	**intr_mask_pd_crc_err_vc0**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc0**
0	0x0	**intr_mask_ppfsm_timeout_vc0**

NVCSI_STREAM_1_INTR_MASK_VC1_0

Offset: 0x609e
Byte Offset: 0x18278
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc1**
4	0x0	**intr_mask_ph_single_crc_err_vc1**
3	0x0	**intr_mask_pd_wc_short_err_vc1**
2	0x0	**intr_mask_pd_crc_err_vc1**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc1**
0	0x0	**intr_mask_ppfsm_timeout_vc1**

NVCSI_STREAM_1_INTR_MASK_VC2_0

Offset: 0x609f
Byte Offset: 0x1827c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc2**

Bit	Reset	Description
4	0x0	**intr_mask_ph_single_crc_err_vc2**
3	0x0	**intr_mask_pd_wc_short_err_vc2**
2	0x0	**intr_mask_pd_crc_err_vc2**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc2**
0	0x0	**intr_mask_ppfsm_timeout_vc2**

NVCSI_STREAM_1_INTR_MASK_VC3_0

Offset: 0x60a0
Byte Offset: 0x18280
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc3**
4	0x0	**intr_mask_ph_single_crc_err_vc3**
3	0x0	**intr_mask_pd_wc_short_err_vc3**
2	0x0	**intr_mask_pd_crc_err_vc3**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc3**
0	0x0	**intr_mask_ppfsm_timeout_vc3**

NVCSI_STREAM_1_INTR_MASK_VC4_0

Offset: 0x60a1
Byte Offset: 0x18284
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc4**
4	0x0	**intr_mask_ph_single_crc_err_vc4**
3	0x0	**intr_mask_pd_wc_short_err_vc4**
2	0x0	**intr_mask_pd_crc_err_vc4**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc4**
0	0x0	**intr_mask_ppfsm_timeout_vc4**

NVCSI_STREAM_1_INTR_MASK_VC5_0

Offset: 0x60a2
Byte Offset: 0x18288
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc5**
4	0x0	**intr_mask_ph_single_crc_err_vc5**
3	0x0	**intr_mask_pd_wc_short_err_vc5**
2	0x0	**intr_mask_pd_crc_err_vc5**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc5**
0	0x0	**intr_mask_ppfsm_timeout_vc5**

NVCSI_STREAM_1_INTR_MASK_VC6_0

Offset: 0x60a3
Byte Offset: 0x1828c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc6**
4	0x0	**intr_mask_ph_single_crc_err_vc6**
3	0x0	**intr_mask_pd_wc_short_err_vc6**
2	0x0	**intr_mask_pd_crc_err_vc6**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc6**
0	0x0	**intr_mask_ppfsm_timeout_vc6**

NVCSI_STREAM_1_INTR_MASK_VC7_0

Offset: 0x60a4
Byte Offset: 0x18290
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc7**
4	0x0	**intr_mask_ph_single_crc_err_vc7**
3	0x0	**intr_mask_pd_wc_short_err_vc7**
2	0x0	**intr_mask_pd_crc_err_vc7**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc7**
0	0x0	**intr_mask_ppfsm_timeout_vc7**

NVCSI_STREAM_1_INTR_MASK_VC8_0

Offset: 0x60a5
Byte Offset: 0x18294
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc8**
4	0x0	**intr_mask_ph_single_crc_err_vc8**
3	0x0	**intr_mask_pd_wc_short_err_vc8**
2	0x0	**intr_mask_pd_crc_err_vc8**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc8**
0	0x0	**intr_mask_ppfsm_timeout_vc8**

NVCSI_STREAM_1_INTR_MASK_VC9_0

Offset: 0x60a6
Byte Offset: 0x18298
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc9**
4	0x0	**intr_mask_ph_single_crc_err_vc9**
3	0x0	**intr_mask_pd_wc_short_err_vc9**
2	0x0	**intr_mask_pd_crc_err_vc9**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc9**
0	0x0	**intr_mask_ppfsm_timeout_vc9**

NVCSI_STREAM_1_INTR_MASK_VC10_0

Offset: 0x60a7
Byte Offset: 0x1829c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc10**
4	0x0	**intr_mask_ph_single_crc_err_vc10**
3	0x0	**intr_mask_pd_wc_short_err_vc10**
2	0x0	**intr_mask_pd_crc_err_vc10**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc10**
0	0x0	**intr_mask_ppfsm_timeout_vc10**

NVCSI_STREAM_1_INTR_MASK_VC11_0

Offset: 0x60a8
Byte Offset: 0x182a0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc11**
4	0x0	**intr_mask_ph_single_crc_err_vc11**
3	0x0	**intr_mask_pd_wc_short_err_vc11**
2	0x0	**intr_mask_pd_crc_err_vc11**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc11**
0	0x0	**intr_mask_ppfsm_timeout_vc11**

NVCSI_STREAM_1_INTR_MASK_VC12_0

Offset: 0x60a9
Byte Offset: 0x182a4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc12**
4	0x0	**intr_mask_ph_single_crc_err_vc12**
3	0x0	**intr_mask_pd_wc_short_err_vc12**
2	0x0	**intr_mask_pd_crc_err_vc12**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc12**
0	0x0	**intr_mask_ppfsm_timeout_vc12**

NVCSI_STREAM_1_INTR_MASK_VC13_0

Offset: 0x60aa
Byte Offset: 0x182a8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc13**
4	0x0	**intr_mask_ph_single_crc_err_vc13**
3	0x0	**intr_mask_pd_wc_short_err_vc13**
2	0x0	**intr_mask_pd_crc_err_vc13**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc13**
0	0x0	**intr_mask_ppfsm_timeout_vc13**

NVCSI_STREAM_1_INTR_MASK_VC14_0

Offset: 0x60ab
Byte Offset: 0x182ac
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc14**
4	0x0	**intr_mask_ph_single_crc_err_vc14**
3	0x0	**intr_mask_pd_wc_short_err_vc14**
2	0x0	**intr_mask_pd_crc_err_vc14**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc14**
0	0x0	**intr_mask_ppfsm_timeout_vc14**

NVCSI_STREAM_1_INTR_MASK_VC15_0

Offset: 0x60ac
Byte Offset: 0x182b0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc15**
4	0x0	**intr_mask_ph_single_crc_err_vc15**
3	0x0	**intr_mask_pd_wc_short_err_vc15**
2	0x0	**intr_mask_pd_crc_err_vc15**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc15**
0	0x0	**intr_mask_ppfsm_timeout_vc15**

NVCSI_STREAM_1_CORRECTABLE_ERR_INTR_STATUS_0

Offset: 0x60ad
Byte Offset: 0x182b4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**err_intr_stat_novc**: Not VC related interrupt
15	0x0	**err_intr_stat_vc15**: VC15 event
14	0x0	**err_intr_stat_vc14**: VC14 event
13	0x0	**err_intr_stat_vc13**: VC13 event
12	0x0	**err_intr_stat_vc12**: VC12 event
11	0x0	**err_intr_stat_vc11**: VC11 event
10	0x0	**err_intr_stat_vc10**: VC10 event
9	0x0	**err_intr_stat_vc9**: VC9 event
8	0x0	**err_intr_stat_vc8**: VC8 event
7	0x0	**err_intr_stat_vc7**: VC7 event
6	0x0	**err_intr_stat_vc6**: VC6 event
5	0x0	**err_intr_stat_vc5**: VC5 event
4	0x0	**err_intr_stat_vc4**: VC4 event
3	0x0	**err_intr_stat_vc3**: VC3 event
2	0x0	**err_intr_stat_vc2**: VC2 event
1	0x0	**err_intr_stat_vc1**: VC1 event
0	0x0	**err_intr_stat_vc0**: VC0 event

NVCSI_STREAM_1_CORRECTABLE_ERR_INTR_STATUS_NOVC_0

Offset: 0x60ae
Byte Offset: 0x182b8
Read/Write: R/W

Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**err_intr_stat_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**err_intr_stat_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_1_CORRECTABLE_ERR_INTR_STATUS_VC0_0

Offset: 0x60af
Byte Offset: 0x182bc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc0**
4	0x0	**err_intr_stat_ph_single_crc_err_vc0**
3	0x0	**err_intr_stat_pd_wc_short_err_vc0**
2	0x0	**err_intr_stat_pd_crc_err_vc0**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc0**
0	0x0	**err_intr_stat_ppfsm_timeout_vc0**

NVCSI_STREAM_1_CORRECTABLE_ERR_INTR_STATUS_VC1_0

Offset: 0x60b0
Byte Offset: 0x182c0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc1**
4	0x0	**err_intr_stat_ph_single_crc_err_vc1**
3	0x0	**err_intr_stat_pd_wc_short_err_vc1**
2	0x0	**err_intr_stat_pd_crc_err_vc1**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc1**
0	0x0	**err_intr_stat_ppfsm_timeout_vc1**

NVCSI_STREAM_1_CORRECTABLE_ERR_INTR_STATUS_VC2_0

Offset: 0x60b1
Byte Offset: 0x182c4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc2**
4	0x0	**err_intr_stat_ph_single_crc_err_vc2**
3	0x0	**err_intr_stat_pd_wc_short_err_vc2**
2	0x0	**err_intr_stat_pd_crc_err_vc2**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc2**
0	0x0	**err_intr_stat_ppfsm_timeout_vc2**

NVCSI_STREAM_1_CORRECTABLE_ERR_INTR_STATUS_VC3_0

Offset: 0x60b2
Byte Offset: 0x182c8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc3**
4	0x0	**err_intr_stat_ph_single_crc_err_vc3**
3	0x0	**err_intr_stat_pd_wc_short_err_vc3**
2	0x0	**err_intr_stat_pd_crc_err_vc3**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc3**
0	0x0	**err_intr_stat_ppfsm_timeout_vc3**

NVCSI_STREAM_1_CORRECTABLE_ERR_INTR_STATUS_VC4_0

Offset: 0x60b3
Byte Offset: 0x182cc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc4**
4	0x0	**err_intr_stat_ph_single_crc_err_vc4**
3	0x0	**err_intr_stat_pd_wc_short_err_vc4**
2	0x0	**err_intr_stat_pd_crc_err_vc4**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc4**
0	0x0	**err_intr_stat_ppfsm_timeout_vc4**

NVCSI_STREAM_1_CORRECTABLE_ERR_INTR_STATUS_VC5_0

Offset: 0x60b4
Byte Offset: 0x182d0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc5**
4	0x0	**err_intr_stat_ph_single_crc_err_vc5**
3	0x0	**err_intr_stat_pd_wc_short_err_vc5**
2	0x0	**err_intr_stat_pd_crc_err_vc5**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc5**
0	0x0	**err_intr_stat_ppfsm_timeout_vc5**

NVCSI_STREAM_1_CORRECTABLE_ERR_INTR_STATUS_VC6_0

Offset: 0x60b5
Byte Offset: 0x182d4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc6**
4	0x0	**err_intr_stat_ph_single_crc_err_vc6**
3	0x0	**err_intr_stat_pd_wc_short_err_vc6**
2	0x0	**err_intr_stat_pd_crc_err_vc6**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc6**
0	0x0	**err_intr_stat_ppfsm_timeout_vc6**

NVCSI_STREAM_1_CORRECTABLE_ERR_INTR_STATUS_VC7_0

Offset: 0x60b6
Byte Offset: 0x182d8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc7**
4	0x0	**err_intr_stat_ph_single_crc_err_vc7**
3	0x0	**err_intr_stat_pd_wc_short_err_vc7**
2	0x0	**err_intr_stat_pd_crc_err_vc7**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc7**
0	0x0	**err_intr_stat_ppfsm_timeout_vc7**

NVCSI_STREAM_1_CORRECTABLE_ERR_INTR_STATUS_VC8_0

Offset: 0x60b7
Byte Offset: 0x182dc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc8**
4	0x0	**err_intr_stat_ph_single_crc_err_vc8**
3	0x0	**err_intr_stat_pd_wc_short_err_vc8**
2	0x0	**err_intr_stat_pd_crc_err_vc8**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc8**
0	0x0	**err_intr_stat_ppfsm_timeout_vc8**

NVCSI_STREAM_1_CORRECTABLE_ERR_INTR_STATUS_VC9_0

Offset: 0x60b8
Byte Offset: 0x182e0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc9**
4	0x0	**err_intr_stat_ph_single_crc_err_vc9**
3	0x0	**err_intr_stat_pd_wc_short_err_vc9**
2	0x0	**err_intr_stat_pd_crc_err_vc9**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc9**
0	0x0	**err_intr_stat_ppfsm_timeout_vc9**

NVCSI_STREAM_1_CORRECTABLE_ERR_INTR_STATUS_VC10_0

Offset: 0x60b9
Byte Offset: 0x182e4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc10**
4	0x0	**err_intr_stat_ph_single_crc_err_vc10**
3	0x0	**err_intr_stat_pd_wc_short_err_vc10**
2	0x0	**err_intr_stat_pd_crc_err_vc10**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc10**
0	0x0	**err_intr_stat_ppfsm_timeout_vc10**

NVCSI_STREAM_1_CORRECTABLE_ERR_INTR_STATUS_VC11_0

Offset: 0x60ba
Byte Offset: 0x182e8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc11**
4	0x0	**err_intr_stat_ph_single_crc_err_vc11**
3	0x0	**err_intr_stat_pd_wc_short_err_vc11**
2	0x0	**err_intr_stat_pd_crc_err_vc11**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc11**
0	0x0	**err_intr_stat_ppfsm_timeout_vc11**

NVCSI_STREAM_1_CORRECTABLE_ERR_INTR_STATUS_VC12_0

Offset: 0x60bb
Byte Offset: 0x182ec
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc12**
4	0x0	**err_intr_stat_ph_single_crc_err_vc12**
3	0x0	**err_intr_stat_pd_wc_short_err_vc12**
2	0x0	**err_intr_stat_pd_crc_err_vc12**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc12**
0	0x0	**err_intr_stat_ppfsm_timeout_vc12**

NVCSI_STREAM_1_CORRECTABLE_ERR_INTR_STATUS_VC13_0

Offset: 0x60bc
Byte Offset: 0x182f0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc13**
4	0x0	**err_intr_stat_ph_single_crc_err_vc13**
3	0x0	**err_intr_stat_pd_wc_short_err_vc13**
2	0x0	**err_intr_stat_pd_crc_err_vc13**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc13**
0	0x0	**err_intr_stat_ppfsm_timeout_vc13**

NVCSI_STREAM_1_CORRECTABLE_ERR_INTR_STATUS_VC14_0

Offset: 0x60bd
Byte Offset: 0x182f4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc14**
4	0x0	**err_intr_stat_ph_single_crc_err_vc14**
3	0x0	**err_intr_stat_pd_wc_short_err_vc14**
2	0x0	**err_intr_stat_pd_crc_err_vc14**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc14**
0	0x0	**err_intr_stat_ppfsm_timeout_vc14**

NVCSI_STREAM_1_CORRECTABLE_ERR_INTR_STATUS_VC15_0

Offset: 0x60be
Byte Offset: 0x182f8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc15**
4	0x0	**err_intr_stat_ph_single_crc_err_vc15**
3	0x0	**err_intr_stat_pd_wc_short_err_vc15**
2	0x0	**err_intr_stat_pd_crc_err_vc15**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc15**
0	0x0	**err_intr_stat_ppfsm_timeout_vc15**

NVCSI_STREAM_1_UNCORRECTABLE_ERR_INTR_STATUS_0

Offset: 0x60bf
Byte Offset: 0x182fc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**err_intr_stat_novc**: Not VC related interrupt
15	0x0	**err_intr_stat_vc15**: VC15 event
14	0x0	**err_intr_stat_vc14**: VC14 event
13	0x0	**err_intr_stat_vc13**: VC13 event
12	0x0	**err_intr_stat_vc12**: VC12 event
11	0x0	**err_intr_stat_vc11**: VC11 event
10	0x0	**err_intr_stat_vc10**: VC10 event
9	0x0	**err_intr_stat_vc9**: VC9 event
8	0x0	**err_intr_stat_vc8**: VC8 event
7	0x0	**err_intr_stat_vc7**: VC7 event

Bit	Reset	Description
6	0x0	**err_intr_stat_vc6**: VC6 event
5	0x0	**err_intr_stat_vc5**: VC5 event
4	0x0	**err_intr_stat_vc4**: VC4 event
3	0x0	**err_intr_stat_vc3**: VC3 event
2	0x0	**err_intr_stat_vc2**: VC2 event
1	0x0	**err_intr_stat_vc1**: VC1 event
0	0x0	**err_intr_stat_vc0**: VC0 event

NVCSI_STREAM_1_UNCORRECTABLE_ERR_INTR_STATUS_NOVC_0

Offset: 0x60c0
Byte Offset: 0x18300
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**err_intr_stat_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**err_intr_stat_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_1_UNCORRECTABLE_ERR_INTR_STATUS_VC0_0

Offset: 0x60c1
Byte Offset: 0x18304
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc0**
4	0x0	**err_intr_stat_ph_single_crc_err_vc0**
3	0x0	**err_intr_stat_pd_wc_short_err_vc0**
2	0x0	**err_intr_stat_pd_crc_err_vc0**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc0**
0	0x0	**err_intr_stat_ppfsm_timeout_vc0**

NVCSI_STREAM_1_UNCORRECTABLE_ERR_INTR_STATUS_VC1_0

Offset: 0x60c2
Byte Offset: 0x18308
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc1**
4	0x0	**err_intr_stat_ph_single_crc_err_vc1**
3	0x0	**err_intr_stat_pd_wc_short_err_vc1**
2	0x0	**err_intr_stat_pd_crc_err_vc1**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc1**
0	0x0	**err_intr_stat_ppfsm_timeout_vc1**

NVCSI_STREAM_1_UNCORRECTABLE_ERR_INTR_STATUS_VC2_0

Offset: 0x60c3
Byte Offset: 0x1830c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc2**
4	0x0	**err_intr_stat_ph_single_crc_err_vc2**
3	0x0	**err_intr_stat_pd_wc_short_err_vc2**
2	0x0	**err_intr_stat_pd_crc_err_vc2**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc2**
0	0x0	**err_intr_stat_ppfsm_timeout_vc2**

NVCSI_STREAM_1_UNCORRECTABLE_ERR_INTR_STATUS_VC3_0

Offset: 0x60c4
Byte Offset: 0x18310
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc3**
4	0x0	**err_intr_stat_ph_single_crc_err_vc3**
3	0x0	**err_intr_stat_pd_wc_short_err_vc3**
2	0x0	**err_intr_stat_pd_crc_err_vc3**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc3**
0	0x0	**err_intr_stat_ppfsm_timeout_vc3**

NVCSI_STREAM_1_UNCORRECTABLE_ERR_INTR_STATUS_VC4_0

Offset: 0x60c5
Byte Offset: 0x18314
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc4**
4	0x0	**err_intr_stat_ph_single_crc_err_vc4**
3	0x0	**err_intr_stat_pd_wc_short_err_vc4**
2	0x0	**err_intr_stat_pd_crc_err_vc4**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc4**
0	0x0	**err_intr_stat_ppfsm_timeout_vc4**

NVCSI_STREAM_1_UNCORRECTABLE_ERR_INTR_STATUS_VC5_0

Offset: 0x60c6
Byte Offset: 0x18318
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc5**
4	0x0	**err_intr_stat_ph_single_crc_err_vc5**
3	0x0	**err_intr_stat_pd_wc_short_err_vc5**
2	0x0	**err_intr_stat_pd_crc_err_vc5**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc5**
0	0x0	**err_intr_stat_ppfsm_timeout_vc5**

NVCSI_STREAM_1_UNCORRECTABLE_ERR_INTR_STATUS_VC6_0

Offset: 0x60c7
Byte Offset: 0x1831c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc6**
4	0x0	**err_intr_stat_ph_single_crc_err_vc6**
3	0x0	**err_intr_stat_pd_wc_short_err_vc6**
2	0x0	**err_intr_stat_pd_crc_err_vc6**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc6**
0	0x0	**err_intr_stat_ppfsm_timeout_vc6**

NVCSI_STREAM_1_UNCORRECTABLE_ERR_INTR_STATUS_VC7_0

Offset: 0x60c8
Byte Offset: 0x18320
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc7**
4	0x0	**err_intr_stat_ph_single_crc_err_vc7**
3	0x0	**err_intr_stat_pd_wc_short_err_vc7**
2	0x0	**err_intr_stat_pd_crc_err_vc7**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc7**
0	0x0	**err_intr_stat_ppfsm_timeout_vc7**

NVCSI_STREAM_1_UNCORRECTABLE_ERR_INTR_STATUS_VC8_0

Offset: 0x60c9
Byte Offset: 0x18324
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc8**
4	0x0	**err_intr_stat_ph_single_crc_err_vc8**
3	0x0	**err_intr_stat_pd_wc_short_err_vc8**
2	0x0	**err_intr_stat_pd_crc_err_vc8**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc8**
0	0x0	**err_intr_stat_ppfsm_timeout_vc8**

NVCSI_STREAM_1_UNCORRECTABLE_ERR_INTR_STATUS_VC9_0

Offset: 0x60ca
Byte Offset: 0x18328
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc9**
4	0x0	**err_intr_stat_ph_single_crc_err_vc9**
3	0x0	**err_intr_stat_pd_wc_short_err_vc9**
2	0x0	**err_intr_stat_pd_crc_err_vc9**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc9**
0	0x0	**err_intr_stat_ppfsm_timeout_vc9**

NVCSI_STREAM_1_UNCORRECTABLE_ERR_INTR_STATUS_VC10_0

Offset: 0x60cb
Byte Offset: 0x1832c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc10**
4	0x0	**err_intr_stat_ph_single_crc_err_vc10**
3	0x0	**err_intr_stat_pd_wc_short_err_vc10**
2	0x0	**err_intr_stat_pd_crc_err_vc10**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc10**
0	0x0	**err_intr_stat_ppfsm_timeout_vc10**

NVCSI_STREAM_1_UNCORRECTABLE_ERR_INTR_STATUS_VC11_0

Offset: 0x60cc
Byte Offset: 0x18330
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc11**
4	0x0	**err_intr_stat_ph_single_crc_err_vc11**
3	0x0	**err_intr_stat_pd_wc_short_err_vc11**
2	0x0	**err_intr_stat_pd_crc_err_vc11**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc11**
0	0x0	**err_intr_stat_ppfsm_timeout_vc11**

NVCSI_STREAM_1_UNCORRECTABLE_ERR_INTR_STATUS_VC12_0

Offset: 0x60cd
Byte Offset: 0x18334
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc12**
4	0x0	**err_intr_stat_ph_single_crc_err_vc12**
3	0x0	**err_intr_stat_pd_wc_short_err_vc12**
2	0x0	**err_intr_stat_pd_crc_err_vc12**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc12**
0	0x0	**err_intr_stat_ppfsm_timeout_vc12**

NVCSI_STREAM_1_UNCORRECTABLE_ERR_INTR_STATUS_VC13_0

Offset: 0x60ce
Byte Offset: 0x18338
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc13**
4	0x0	**err_intr_stat_ph_single_crc_err_vc13**
3	0x0	**err_intr_stat_pd_wc_short_err_vc13**
2	0x0	**err_intr_stat_pd_crc_err_vc13**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc13**
0	0x0	**err_intr_stat_ppfsm_timeout_vc13**

NVCSI_STREAM_1_UNCORRECTABLE_ERR_INTR_STATUS_VC14_0

Offset: 0x60cf
Byte Offset: 0x1833c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc14**
4	0x0	**err_intr_stat_ph_single_crc_err_vc14**
3	0x0	**err_intr_stat_pd_wc_short_err_vc14**
2	0x0	**err_intr_stat_pd_crc_err_vc14**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc14**
0	0x0	**err_intr_stat_ppfsm_timeout_vc14**

NVCSI_STREAM_1_UNCORRECTABLE_ERR_INTR_STATUS_VC15_0

Offset: 0x60d0
Byte Offset: 0x18340
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc15**
4	0x0	**err_intr_stat_ph_single_crc_err_vc15**
3	0x0	**err_intr_stat_pd_wc_short_err_vc15**
2	0x0	**err_intr_stat_pd_crc_err_vc15**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc15**
0	0x0	**err_intr_stat_ppfsm_timeout_vc15**

NVCSI_STREAM_1_ERR_INTR_MASK_NOVC_0

Offset: 0x60d1
Byte Offset: 0x18344
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**err_intr_mask_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**err_intr_mask_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_1_ERR_INTR_MASK_VC0_0

Offset: 0x60d2
Byte Offset: 0x18348
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc0**
4	0x0	**err_intr_mask_ph_single_crc_err_vc0**
3	0x0	**err_intr_mask_pd_wc_short_err_vc0**
2	0x0	**err_intr_mask_pd_crc_err_vc0**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc0**
0	0x0	**err_intr_mask_ppfsm_timeout_vc0**

NVCSI_STREAM_1_ERR_INTR_MASK_VC1_0

Offset: 0x60d3
Byte Offset: 0x1834c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc1**

Bit	Reset	Description
4	0x0	**err_intr_mask_ph_single_crc_err_vc1**
3	0x0	**err_intr_mask_pd_wc_short_err_vc1**
2	0x0	**err_intr_mask_pd_crc_err_vc1**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc1**
0	0x0	**err_intr_mask_ppfsm_timeout_vc1**

NVCSI_STREAM_1_ERR_INTR_MASK_VC2_0

Offset: 0x60d4
Byte Offset: 0x18350
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc2**
4	0x0	**err_intr_mask_ph_single_crc_err_vc2**
3	0x0	**err_intr_mask_pd_wc_short_err_vc2**
2	0x0	**err_intr_mask_pd_crc_err_vc2**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc2**
0	0x0	**err_intr_mask_ppfsm_timeout_vc2**

NVCSI_STREAM_1_ERR_INTR_MASK_VC3_0

Offset: 0x60d5
Byte Offset: 0x18354
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc3**
4	0x0	**err_intr_mask_ph_single_crc_err_vc3**
3	0x0	**err_intr_mask_pd_wc_short_err_vc3**
2	0x0	**err_intr_mask_pd_crc_err_vc3**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc3**
0	0x0	**err_intr_mask_ppfsm_timeout_vc3**

NVCSI_STREAM_1_ERR_INTR_MASK_VC4_0

Offset: 0x60d6
Byte Offset: 0x18358
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc4**
4	0x0	**err_intr_mask_ph_single_crc_err_vc4**
3	0x0	**err_intr_mask_pd_wc_short_err_vc4**
2	0x0	**err_intr_mask_pd_crc_err_vc4**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc4**
0	0x0	**err_intr_mask_ppfsm_timeout_vc4**

NVCSI_STREAM_1_ERR_INTR_MASK_VC5_0

Offset: 0x60d7
Byte Offset: 0x1835c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc5**
4	0x0	**err_intr_mask_ph_single_crc_err_vc5**
3	0x0	**err_intr_mask_pd_wc_short_err_vc5**
2	0x0	**err_intr_mask_pd_crc_err_vc5**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc5**
0	0x0	**err_intr_mask_ppfsm_timeout_vc5**

NVCSI_STREAM_1_ERR_INTR_MASK_VC6_0

Offset: 0x60d8
Byte Offset: 0x18360
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc6**
4	0x0	**err_intr_mask_ph_single_crc_err_vc6**
3	0x0	**err_intr_mask_pd_wc_short_err_vc6**
2	0x0	**err_intr_mask_pd_crc_err_vc6**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc6**
0	0x0	**err_intr_mask_ppfsm_timeout_vc6**

NVCSI_STREAM_1_ERR_INTR_MASK_VC7_0

Offset: 0x60d9
Byte Offset: 0x18364
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc7**
4	0x0	**err_intr_mask_ph_single_crc_err_vc7**
3	0x0	**err_intr_mask_pd_wc_short_err_vc7**
2	0x0	**err_intr_mask_pd_crc_err_vc7**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc7**
0	0x0	**err_intr_mask_ppfsm_timeout_vc7**

NVCSI_STREAM_1_ERR_INTR_MASK_VC8_0

Offset: 0x60da
Byte Offset: 0x18368
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc8**
4	0x0	**err_intr_mask_ph_single_crc_err_vc8**
3	0x0	**err_intr_mask_pd_wc_short_err_vc8**
2	0x0	**err_intr_mask_pd_crc_err_vc8**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc8**
0	0x0	**err_intr_mask_ppfsm_timeout_vc8**

NVCSI_STREAM_1_ERR_INTR_MASK_VC9_0

Offset: 0x60db
Byte Offset: 0x1836c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc9**
4	0x0	**err_intr_mask_ph_single_crc_err_vc9**
3	0x0	**err_intr_mask_pd_wc_short_err_vc9**
2	0x0	**err_intr_mask_pd_crc_err_vc9**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc9**
0	0x0	**err_intr_mask_ppfsm_timeout_vc9**

NVCSI_STREAM_1_ERR_INTR_MASK_VC10_0

Offset: 0x60dc
Byte Offset: 0x18370
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc10**
4	0x0	**err_intr_mask_ph_single_crc_err_vc10**
3	0x0	**err_intr_mask_pd_wc_short_err_vc10**
2	0x0	**err_intr_mask_pd_crc_err_vc10**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc10**
0	0x0	**err_intr_mask_ppfsm_timeout_vc10**

NVCSI_STREAM_1_ERR_INTR_MASK_VC11_0

Offset: 0x60dd
Byte Offset: 0x18374
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc11**
4	0x0	**err_intr_mask_ph_single_crc_err_vc11**
3	0x0	**err_intr_mask_pd_wc_short_err_vc11**
2	0x0	**err_intr_mask_pd_crc_err_vc11**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc11**
0	0x0	**err_intr_mask_ppfsm_timeout_vc11**

NVCSI_STREAM_1_ERR_INTR_MASK_VC12_0

Offset: 0x60de
Byte Offset: 0x18378
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc12**
4	0x0	**err_intr_mask_ph_single_crc_err_vc12**
3	0x0	**err_intr_mask_pd_wc_short_err_vc12**
2	0x0	**err_intr_mask_pd_crc_err_vc12**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc12**
0	0x0	**err_intr_mask_ppfsm_timeout_vc12**

NVCSI_STREAM_1_ERR_INTR_MASK_VC13_0

Offset: 0x60df
Byte Offset: 0x1837c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc13**
4	0x0	**err_intr_mask_ph_single_crc_err_vc13**
3	0x0	**err_intr_mask_pd_wc_short_err_vc13**
2	0x0	**err_intr_mask_pd_crc_err_vc13**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc13**
0	0x0	**err_intr_mask_ppfsm_timeout_vc13**

NVCSI_STREAM_1_ERR_INTR_MASK_VC14_0

Offset: 0x60e0
Byte Offset: 0x18380
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc14**
4	0x0	**err_intr_mask_ph_single_crc_err_vc14**
3	0x0	**err_intr_mask_pd_wc_short_err_vc14**
2	0x0	**err_intr_mask_pd_crc_err_vc14**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc14**
0	0x0	**err_intr_mask_ppfsm_timeout_vc14**

NVCSI_STREAM_1_ERR_INTR_MASK_VC15_0

Offset: 0x60e1
Byte Offset: 0x18384
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc15**
4	0x0	**err_intr_mask_ph_single_crc_err_vc15**
3	0x0	**err_intr_mask_pd_wc_short_err_vc15**
2	0x0	**err_intr_mask_pd_crc_err_vc15**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc15**
0	0x0	**err_intr_mask_ppfsm_timeout_vc15**

NVCSI_STREAM_1_ERR_INTR_TYPE_NOVC_0

Offset: 0x60e2
Byte Offset: 0x18388
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**err_intr_type_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**err_intr_type_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_1_ERR_INTR_TYPE_VC0_0

Offset: 0x60e3
Byte Offset: 0x1838c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc0**
4	0x0	**err_intr_type_ph_single_crc_err_vc0**

Bit	Reset	Description
3	0x0	**err_intr_type_pd_wc_short_err_vc0**
2	0x0	**err_intr_type_pd_crc_err_vc0**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc0**
0	0x0	**err_intr_type_ppfsm_timeout_vc0**

NVCSI_STREAM_1_ERR_INTR_TYPE_VC1_0

Offset: 0x60e4
Byte Offset: 0x18390
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc1**
4	0x0	**err_intr_type_ph_single_crc_err_vc1**
3	0x0	**err_intr_type_pd_wc_short_err_vc1**
2	0x0	**err_intr_type_pd_crc_err_vc1**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc1**
0	0x0	**err_intr_type_ppfsm_timeout_vc1**

NVCSI_STREAM_1_ERR_INTR_TYPE_VC2_0

Offset: 0x60e5
Byte Offset: 0x18394
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc2**

Bit	Reset	Description
4	0x0	**err_intr_type_ph_single_crc_err_vc2**
3	0x0	**err_intr_type_pd_wc_short_err_vc2**
2	0x0	**err_intr_type_pd_crc_err_vc2**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc2**
0	0x0	**err_intr_type_ppfsm_timeout_vc2**

NVCSI_STREAM_1_ERR_INTR_TYPE_VC3_0

Offset: 0x60e6
Byte Offset: 0x18398
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc3**
4	0x0	**err_intr_type_ph_single_crc_err_vc3**
3	0x0	**err_intr_type_pd_wc_short_err_vc3**
2	0x0	**err_intr_type_pd_crc_err_vc3**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc3**
0	0x0	**err_intr_type_ppfsm_timeout_vc3**

NVCSI_STREAM_1_ERR_INTR_TYPE_VC4_0

Offset: 0x60e7
Byte Offset: 0x1839c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc4**
4	0x0	**err_intr_type_ph_single_crc_err_vc4**
3	0x0	**err_intr_type_pd_wc_short_err_vc4**
2	0x0	**err_intr_type_pd_crc_err_vc4**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc4**
0	0x0	**err_intr_type_ppfsm_timeout_vc4**

NVCSI_STREAM_1_ERR_INTR_TYPE_VC5_0

Offset: 0x60e8
Byte Offset: 0x183a0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc5**
4	0x0	**err_intr_type_ph_single_crc_err_vc5**
3	0x0	**err_intr_type_pd_wc_short_err_vc5**
2	0x0	**err_intr_type_pd_crc_err_vc5**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc5**
0	0x0	**err_intr_type_ppfsm_timeout_vc5**

NVCSI_STREAM_1_ERR_INTR_TYPE_VC6_0

Offset: 0x60e9
Byte Offset: 0x183a4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc6**
4	0x0	**err_intr_type_ph_single_crc_err_vc6**
3	0x0	**err_intr_type_pd_wc_short_err_vc6**
2	0x0	**err_intr_type_pd_crc_err_vc6**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc6**
0	0x0	**err_intr_type_ppfsm_timeout_vc6**

NVCSI_STREAM_1_ERR_INTR_TYPE_VC7_0

Offset: 0x60ea
Byte Offset: 0x183a8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc7**
4	0x0	**err_intr_type_ph_single_crc_err_vc7**
3	0x0	**err_intr_type_pd_wc_short_err_vc7**
2	0x0	**err_intr_type_pd_crc_err_vc7**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc7**
0	0x0	**err_intr_type_ppfsm_timeout_vc7**

NVCSI_STREAM_1_ERR_INTR_TYPE_VC8_0

Offset: 0x60eb
Byte Offset: 0x183ac
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc8**
4	0x0	**err_intr_type_ph_single_crc_err_vc8**
3	0x0	**err_intr_type_pd_wc_short_err_vc8**
2	0x0	**err_intr_type_pd_crc_err_vc8**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc8**
0	0x0	**err_intr_type_ppfsm_timeout_vc8**

NVCSI_STREAM_1_ERR_INTR_TYPE_VC9_0

Offset: 0x60ec
Byte Offset: 0x183b0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc9**
4	0x0	**err_intr_type_ph_single_crc_err_vc9**
3	0x0	**err_intr_type_pd_wc_short_err_vc9**
2	0x0	**err_intr_type_pd_crc_err_vc9**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc9**
0	0x0	**err_intr_type_ppfsm_timeout_vc9**

NVCSI_STREAM_1_ERR_INTR_TYPE_VC10_0

Offset: 0x60ed
Byte Offset: 0x183b4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc10**
4	0x0	**err_intr_type_ph_single_crc_err_vc10**
3	0x0	**err_intr_type_pd_wc_short_err_vc10**
2	0x0	**err_intr_type_pd_crc_err_vc10**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc10**
0	0x0	**err_intr_type_ppfsm_timeout_vc10**

NVCSI_STREAM_1_ERR_INTR_TYPE_VC11_0

Offset: 0x60ee
Byte Offset: 0x183b8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc11**
4	0x0	**err_intr_type_ph_single_crc_err_vc11**
3	0x0	**err_intr_type_pd_wc_short_err_vc11**
2	0x0	**err_intr_type_pd_crc_err_vc11**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc11**
0	0x0	**err_intr_type_ppfsm_timeout_vc11**

NVCSI_STREAM_1_ERR_INTR_TYPE_VC12_0

Offset: 0x60ef
Byte Offset: 0x183bc
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc12**
4	0x0	**err_intr_type_ph_single_crc_err_vc12**
3	0x0	**err_intr_type_pd_wc_short_err_vc12**
2	0x0	**err_intr_type_pd_crc_err_vc12**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc12**
0	0x0	**err_intr_type_ppfsm_timeout_vc12**

NVCSI_STREAM_1_ERR_INTR_TYPE_VC13_0

Offset: 0x60f0
Byte Offset: 0x183c0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc13**
4	0x0	**err_intr_type_ph_single_crc_err_vc13**
3	0x0	**err_intr_type_pd_wc_short_err_vc13**
2	0x0	**err_intr_type_pd_crc_err_vc13**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc13**
0	0x0	**err_intr_type_ppfsm_timeout_vc13**

NVCSI_STREAM_1_ERR_INTR_TYPE_VC14_0

Offset: 0x60f1
Byte Offset: 0x183c4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc14**
4	0x0	**err_intr_type_ph_single_crc_err_vc14**
3	0x0	**err_intr_type_pd_wc_short_err_vc14**
2	0x0	**err_intr_type_pd_crc_err_vc14**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc14**
0	0x0	**err_intr_type_ppfsm_timeout_vc14**

NVCSI_STREAM_1_ERR_INTR_TYPE_VC15_0

Offset: 0x60f2
Byte Offset: 0x183c8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc15**
4	0x0	**err_intr_type_ph_single_crc_err_vc15**
3	0x0	**err_intr_type_pd_wc_short_err_vc15**
2	0x0	**err_intr_type_pd_crc_err_vc15**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc15**
0	0x0	**err_intr_type_ppfsm_timeout_vc15**

NVCSI_STREAM_1_TPG_ENABLE_0

Offset: 0x60f3
Byte Offset: 0x183cc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	DISABLE	**TPG_ENABLE**: Enable the TPG path **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_1_TPG_VC_ENABLE_0

Offset: 0x60f4
Byte Offset: 0x183d0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15	DISABLE	**VC15_EN**: Enable bit for the VC15 **0** = DISABLE **1** = ENABLE
14	DISABLE	**VC14_EN**: Enable bit for the VC14 **0** = DISABLE **1** = ENABLE
13	DISABLE	**VC13_EN**: Enable bit for the VC13 **0** = DISABLE **1** = ENABLE
12	DISABLE	**VC12_EN**: Enable bit for the VC12 **0** = DISABLE **1** = ENABLE
11	DISABLE	**VC11_EN**: Enable bit for the VC11 **0** = DISABLE **1** = ENABLE
10	DISABLE	**VC10_EN**: Enable bit for the VC10 **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
9	DISABLE	**VC9_EN**: Enable bit for the VC9 **0** = DISABLE **1** = ENABLE
8	DISABLE	**VC8_EN**: Enable bit for the VC8 **0** = DISABLE **1** = ENABLE
7	DISABLE	**VC7_EN**: Enable bit for the VC7 **0** = DISABLE **1** = ENABLE
6	DISABLE	**VC6_EN**: Enable bit for the VC6 **0** = DISABLE **1** = ENABLE
5	DISABLE	**VC5_EN**: Enable bit for the VC5 **0** = DISABLE **1** = ENABLE
4	DISABLE	**VC4_EN**: Enable bit for the VC4 **0** = DISABLE **1** = ENABLE
3	DISABLE	**VC3_EN**: Enable bit for the VC3 **0** = DISABLE **1** = ENABLE
2	DISABLE	**VC2_EN**: Enable bit for the VC2 **0** = DISABLE **1** = ENABLE
1	DISABLE	**VC1_EN**: Enable bit for the VC1 **0** = DISABLE **1** = ENABLE
0	DISABLE	**VC0_EN**: Enable bit for the VC0 **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_1_LOAD_TPG_CFG_0

Offset: 0x60f5
Byte Offset: 0x183d4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**LOAD**: Load the shadow register

NVCSI_STREAM_1_TPG_CTRL_0

Offset: 0x60f6
Byte Offset: 0x183d8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000008a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1000,1010)

Bit	Reset	Description
7:4	0x8	**DATA_SPEED**: Control the data generation speed, valid range is 1~8.
3	ENABLE	**SKIP_LS_LE_PKT**: If the LS/LE packet need to generated. **0** = DISABLE **1** = ENABLE
2	DISABLE	**OVERRIDE_CRC**: Override the packet header CRC and payload CRC. **0** = DISABLE **1** = ENABLE
1	CORE	**DEST**: The TPG pattern is send to PP or send to CIL for TX. **0** = CIL **1** = CORE

Bit	Reset	Description
0	DPHY	**PHY_MODE**: CPHY or DPHY packet structure for TPG **0** = DPHY **1** = CPHY

NVCSI_STREAM_1_TPG_VBLANK_0

Offset: 0x60f7
Byte Offset: 0x183dc
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000800 (0b0000,0000,0000,0000,0000,1000,0000,0000)

Bit	Reset	Description
31:0	0x800	**VBLANK**: The vblank value, unit is nvcsi clock cycles.

NVCSI_STREAM_1_TPG_HBLANK_0

Offset: 0x60f8
Byte Offset: 0x183e0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000800 (0b0000,0000,0000,0000,0000,1000,0000,0000)

Bit	Reset	Description
31:0	0x800	**HBLANK**: The hblank value, unit is nvcsi clock cycles.

NVCSI_STREAM_1_TPG_STATUS_0

Offset: 0x60f9
Byte Offset: 0x183e4
Read/Write: RO
Parity Protection: N
Shadow: N

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0001ffff (0bxxxx,xxxx,xxxx,xxx1,1111,1111,1111,1111)

Bit	Reset	Description
16	IDLE	**VC15_STATUS**: Indicate the VC15 status **0** = BUSY **1** = IDLE
15	IDLE	**VC14_STATUS**: Indicate the VC14 status **0** = BUSY **1** = IDLE
14	IDLE	**VC13_STATUS**: Indicate the VC13 status **0** = BUSY **1** = IDLE
13	IDLE	**VC12_STATUS**: Indicate the VC12 status **0** = BUSY **1** = IDLE
12	IDLE	**VC11_STATUS**: Indicate the VC11 status **0** = BUSY **1** = IDLE
11	IDLE	**VC10_STATUS**: Indicate the VC10 status **0** = BUSY **1** = IDLE
10	IDLE	**VC9_STATUS**: Indicate the VC9 status **0** = BUSY **1** = IDLE
9	IDLE	**VC8_STATUS**: Indicate the VC8 status **0** = BUSY **1** = IDLE
8	IDLE	**VC7_STATUS**: Indicate the VC7 status **0** = BUSY **1** = IDLE

Bit	Reset	Description
7	IDLE	**VC6_STATUS**: Indicate the VC6 status **0** = BUSY **1** = IDLE
6	IDLE	**VC5_STATUS**: Indicate the VC5 status **0** = BUSY **1** = IDLE
5	IDLE	**VC4_STATUS**: Indicate the VC4 status **0** = BUSY **1** = IDLE
4	IDLE	**VC3_STATUS**: Indicate the VC3 status **0** = BUSY **1** = IDLE
3	IDLE	**VC2_STATUS**: Indicate the VC2 status **0** = BUSY **1** = IDLE
2	IDLE	**VC1_STATUS**: Indicate the VC1 status **0** = BUSY **1** = IDLE
1	IDLE	**VC0_STATUS**: Indicate the VC0 status **0** = BUSY **1** = IDLE
0	IDLE	**STATUS**: Indicate the TPG is in idle state, all packet has been send and all VC are disabled. **0** = BUSY **1** = IDLE

NVCSI_STREAM_1_TPG_PH_ECC_0

Offset: 0x60fa
Byte Offset: 0x183e8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:24	0x0	**LINE_ECC**: This field is for the long packet header ECC.
23:18	0x0	**EOL_ECC**: This field is for the EOL short packet ECC.
17:12	0x0	**SOL_ECC**: This field is for the SOL short packet ECC.
11:6	0x0	**EOF_ECC**: This field is for the EOF short packet ECC.
5:0	0x0	**SOF_ECC**: The TPG will not generate ECC for a packet. When using the TPG, SW should set the PP to skip the ecc check. To verify the ecc logic for safety BIST, SW can write a pre-calculated ECC for the TPG, when use with this mode, the TPG should generate a grescale pattern. This field is for the SOF short packet ECC.

NVCSI_STREAM_1_TPG_PF_CRC_0

Offset: 0x60fb
Byte Offset: 0x183ec
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PF_CRC**: This field is for the long packet payload CRC override.

NVCSI_STREAM_1_TPG_PH_SOF_CRC_0

Offset: 0x60fc
Byte Offset: 0x183f0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This field is for the CPHY SOF packet second packet header CRC override.
15:0	0x0	**CRC_0**: This field is for the CPHY SOF packet first packet header CRC override.

NVCSI_STREAM_1_TPG_PH_EOF_CRC_0

Offset: 0x60fd
Byte Offset: 0x183f4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This field is for the CPHY EOF packet second packet header CRC override.
15:0	0x0	**CRC_0**: This field is for the CPHY EOF packet first packet header CRC override.

NVCSI_STREAM_1_TPG_PH_SOL_CRC_0

Offset: 0x60fe
Byte Offset: 0x183f8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This field is for the CPHY SOL packet second packet header CRC override.
15:0	0x0	**CRC_0**: This field is for the CPHY SOL packet first packet header CRC override.

NVCSI_STREAM_1_TPG_PH_EOL_CRC_0

Offset: 0x60ff
Byte Offset: 0x183fc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This field is for the CPHY EOL packet second packet header CRC override.
15:0	0x0	**CRC_0**: This field is for the CPHY EOL packet first packet header CRC override.

NVCSI_STREAM_1_TPG_PH_LONG_PKT_CRC_0

Offset: 0x6100
Byte Offset: 0x18400
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This filed is for the CPHY long packet second packet header CRC override.
15:0	0x0	**CRC_0**: This filed is for the CPHY long packet first packet header CRC override.

NVCSI_STREAM_1_TPG_PKT_DELIMETER_0

Offset: 0x6101
Byte Offset: 0x18404
Read/Write: R/W
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0011)

Bit	Reset	Description
15:0	0x3	**NUM**: This field define the cycle number between two packet. Model the LP11 period between two packet, valid range is 1 to 65535.

NVCSI_STREAM_1_VC0_TPG_GAIN_CTRL_0

Offset: 0x6102
Byte Offset: 0x18408
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_1_VC1_TPG_GAIN_CTRL_0

Offset: 0x6103
Byte Offset: 0x1840c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_1_VC2_TPG_GAIN_CTRL_0

Offset: 0x6104
Byte Offset: 0x18410
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_1_VC3_TPG_GAIN_CTRL_0

Offset: 0x6105
Byte Offset: 0x18414
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_1_VC4_TPG_GAIN_CTRL_0

Offset: 0x6106
Byte Offset: 0x18418
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_1_VC5_TPG_GAIN_CTRL_0

Offset: 0x6107
Byte Offset: 0x1841c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_1_VC6_TPG_GAIN_CTRL_0

Offset: 0x6108
Byte Offset: 0x18420
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_1_VC7_TPG_GAIN_CTRL_0

Offset: 0x6109
Byte Offset: 0x18424
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_1_VC8_TPG_GAIN_CTRL_0

Offset: 0x610a
Byte Offset: 0x18428
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_1_VC9_TPG_GAIN_CTRL_0

Offset: 0x610b
Byte Offset: 0x1842c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_1_VC10_TPG_GAIN_CTRL_0

Offset: 0x610c
Byte Offset: 0x18430
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_1_VC11_TPG_GAIN_CTRL_0

Offset: 0x610d
Byte Offset: 0x18434
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_1_VC12_TPG_GAIN_CTRL_0

Offset: 0x610e
Byte Offset: 0x18438
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_1_VC13_TPG_GAIN_CTRL_0

Offset: 0x610f
Byte Offset: 0x1843c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_1_VC14_TPG_GAIN_CTRL_0

Offset: 0x6110
Byte Offset: 0x18440
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_1_VC15_TPG_GAIN_CTRL_0

Offset: 0x6111
Byte Offset: 0x18444
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_1_SPARE_0
Offset: 0x6112
Byte Offset: 0x18448
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**stream_reg**

7.2.1.3.3 NVCSI Stream2 Registers

NVCSI_STREAM_2_SW_RESET_CTRL_0

Offset: 0x8000
Byte Offset: 0x20000
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DISABLE	**cfg_swreset**: Reset the pixel parser. **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_2_SLCG_CTRL_0

Offset: 0x8001
Byte Offset: 0x20004
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DISABLE	**cfg_slcg_override**: Enable the SLCG override for the pixel parser. **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_2_PP_PHY_CTRL_0

Offset: 0x8002
Byte Offset: 0x20008
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DPHY	**cfg_phy_mode**: **0** = DPHY **1** = CPHY

NVCSI_STREAM_2_PP_EN_CTRL_0

Offset: 0x8003
Byte Offset: 0x2000c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	CIL	**cfg_src**: The pixel source of the pixel parser **0** = CIL **1** = TPG
0	DISABLE	**cfg_pp_en**: Pixel Parser streaming enable **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_2_VC0_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x8004
Byte Offset: 0x20010
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc0_dt_nooverride_0**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC0_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x8005
Byte Offset: 0x20014
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc0_dt_nooverride_1**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC0_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x8006
Byte Offset: 0x20018
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc0_dt_nooverride_2**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC0_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x8007
Byte Offset: 0x2001c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc0_dt_nooverride_3**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC0_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x8008
Byte Offset: 0x20020
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc0_dt_nooverride_4**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC0_DT_OVERRIDE_0

Offset: 0x8009
Byte Offset: 0x20024
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc0_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc0_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc0_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_2_VC1_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x800a
Byte Offset: 0x20028
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc1_dt_nooverride_0**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC1_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x800b
Byte Offset: 0x2002c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc1_dt_nooverride_1**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC1_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x800c
Byte Offset: 0x20030
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc1_dt_nooverride_2**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC1_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x800d
Byte Offset: 0x20034
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc1_dt_nooverride_3**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC1_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x800e
Byte Offset: 0x20038
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc1_dt_nooverride_4**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC1_DT_OVERRIDE_0

Offset: 0x800f
Byte Offset: 0x2003c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc1_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc1_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc1_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_2_VC2_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x8010
Byte Offset: 0x20040
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc2_dt_nooverride_0**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC2_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x8011
Byte Offset: 0x20044
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc2_dt_nooverride_1**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC2_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x8012
Byte Offset: 0x20048
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc2_dt_nooverride_2**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC2_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x8013
Byte Offset: 0x2004c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc2_dt_nooverride_3**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC2_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x8014
Byte Offset: 0x20050
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc2_dt_nooverride_4**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC2_DT_OVERRIDE_0

Offset: 0x8015
Byte Offset: 0x20054
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc2_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc2_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc2_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_2_VC3_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x8016
Byte Offset: 0x20058
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc3_dt_nooverride_0**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC3_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x8017
Byte Offset: 0x2005c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc3_dt_nooverride_1**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC3_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x8018
Byte Offset: 0x20060
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc3_dt_nooverride_2**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC3_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x8019
Byte Offset: 0x20064
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc3_dt_nooverride_3**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC3_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x801a
Byte Offset: 0x20068
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc3_dt_nooverride_4**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC3_DT_OVERRIDE_0

Offset: 0x801b
Byte Offset: 0x2006c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc3_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc3_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc3_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_2_VC4_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x801c
Byte Offset: 0x20070
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc4_dt_nooverride_0**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC4_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x801d
Byte Offset: 0x20074
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc4_dt_nooverride_1**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC4_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x801e
Byte Offset: 0x20078
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc4_dt_nooverride_2**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC4_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x801f
Byte Offset: 0x2007c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc4_dt_nooverride_3**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC4_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x8020
Byte Offset: 0x20080
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc4_dt_nooverride_4**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC4_DT_OVERRIDE_0

Offset: 0x8021
Byte Offset: 0x20084
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc4_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc4_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc4_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_2_VC5_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x8022
Byte Offset: 0x20088
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc5_dt_nooverride_0**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC5_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x8023
Byte Offset: 0x2008c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc5_dt_nooverride_1**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC5_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x8024
Byte Offset: 0x20090
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc5_dt_nooverride_2**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC5_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x8025
Byte Offset: 0x20094
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc5_dt_nooverride_3**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC5_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x8026
Byte Offset: 0x20098
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc5_dt_nooverride_4**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC5_DT_OVERRIDE_0

Offset: 0x8027
Byte Offset: 0x2009c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc5_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc5_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc5_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_2_VC6_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x8028
Byte Offset: 0x200a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc6_dt_nooverride_0**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC6_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x8029
Byte Offset: 0x200a4
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc6_dt_nooverride_1**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC6_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x802a
Byte Offset: 0x200a8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc6_dt_nooverride_2**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC6_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x802b
Byte Offset: 0x200ac
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc6_dt_nooverride_3**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC6_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x802c
Byte Offset: 0x200b0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc6_dt_nooverride_4**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC6_DT_OVERRIDE_0

Offset: 0x802d
Byte Offset: 0x200b4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc6_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc6_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc6_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_2_VC7_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x802e
Byte Offset: 0x200b8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc7_dt_nooverride_0**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC7_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x802f
Byte Offset: 0x200bc
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc7_dt_nooverride_1**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC7_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x8030
Byte Offset: 0x200c0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc7_dt_nooverride_2**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC7_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x8031
Byte Offset: 0x200c4
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc7_dt_nooverride_3**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC7_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x8032
Byte Offset: 0x200c8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc7_dt_nooverride_4**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC7_DT_OVERRIDE_0

Offset: 0x8033
Byte Offset: 0x200cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc7_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc7_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc7_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_2_VC8_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x8034
Byte Offset: 0x200d0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc8_dt_nooverride_0**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC8_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x8035
Byte Offset: 0x200d4
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc8_dt_nooverride_1**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC8_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x8036
Byte Offset: 0x200d8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc8_dt_nooverride_2**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC8_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x8037
Byte Offset: 0x200dc
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc8_dt_nooverride_3**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC8_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x8038
Byte Offset: 0x200e0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc8_dt_nooverride_4**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC8_DT_OVERRIDE_0

Offset: 0x8039
Byte Offset: 0x200e4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc8_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc8_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc8_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_2_VC9_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x803a
Byte Offset: 0x200e8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc9_dt_nooverride_0**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC9_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x803b
Byte Offset: 0x200ec
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc9_dt_nooverride_1**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC9_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x803c
Byte Offset: 0x200f0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc9_dt_nooverride_2**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC9_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x803d
Byte Offset: 0x200f4
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc9_dt_nooverride_3**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC9_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x803e
Byte Offset: 0x200f8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc9_dt_nooverride_4**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC9_DT_OVERRIDE_0

Offset: 0x803f
Byte Offset: 0x200fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc9_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc9_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc9_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_2_VC10_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x8040
Byte Offset: 0x20100
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc10_dt_nooverride_0**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC10_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x8041
Byte Offset: 0x20104
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc10_dt_nooverride_1**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC10_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x8042
Byte Offset: 0x20108
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc10_dt_nooverride_2**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC10_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x8043
Byte Offset: 0x2010c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc10_dt_nooverride_3**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC10_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x8044
Byte Offset: 0x20110
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc10_dt_nooverride_4**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC10_DT_OVERRIDE_0

Offset: 0x8045
Byte Offset: 0x20114
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc10_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc10_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc10_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_2_VC11_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x8046
Byte Offset: 0x20118
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc11_dt_nooverride_0**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC11_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x8047
Byte Offset: 0x2011c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc11_dt_nooverride_1**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC11_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x8048
Byte Offset: 0x20120
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc11_dt_nooverride_2**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC11_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x8049
Byte Offset: 0x20124
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc11_dt_nooverride_3**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC11_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x804a
Byte Offset: 0x20128
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc11_dt_nooverride_4**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC11_DT_OVERRIDE_0

Offset: 0x804b
Byte Offset: 0x2012c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc11_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc11_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc11_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_2_VC12_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x804c
Byte Offset: 0x20130
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc12_dt_nooverride_0**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC12_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x804d
Byte Offset: 0x20134
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc12_dt_nooverride_1**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC12_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x804e
Byte Offset: 0x20138
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc12_dt_nooverride_2**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC12_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x804f
Byte Offset: 0x2013c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc12_dt_nooverride_3**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC12_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x8050
Byte Offset: 0x20140
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc12_dt_nooverride_4**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC12_DT_OVERRIDE_0

Offset: 0x8051
Byte Offset: 0x20144
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc12_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc12_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc12_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_2_VC13_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x8052
Byte Offset: 0x20148
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc13_dt_nooverride_0**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC13_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x8053
Byte Offset: 0x2014c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc13_dt_nooverride_1**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC13_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x8054
Byte Offset: 0x20150
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc13_dt_nooverride_2**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC13_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x8055
Byte Offset: 0x20154
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc13_dt_nooverride_3**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC13_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x8056
Byte Offset: 0x20158
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc13_dt_nooverride_4**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC13_DT_OVERRIDE_0

Offset: 0x8057
Byte Offset: 0x2015c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc13_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc13_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc13_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_2_VC14_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x8058
Byte Offset: 0x20160
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc14_dt_nooverride_0**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC14_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x8059
Byte Offset: 0x20164
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc14_dt_nooverride_1**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC14_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x805a
Byte Offset: 0x20168
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc14_dt_nooverride_2**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC14_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x805b
Byte Offset: 0x2016c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc14_dt_nooverride_3**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC14_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x805c
Byte Offset: 0x20170
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc14_dt_nooverride_4**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC14_DT_OVERRIDE_0

Offset: 0x805d
Byte Offset: 0x20174
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc14_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc14_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc14_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_2_VC15_DT_NOOVERRIDE_CTRL_0_0

Offset: 0x805e
Byte Offset: 0x20178
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc15_dt_nooverride_0**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC15_DT_NOOVERRIDE_CTRL_1_0

Offset: 0x805f
Byte Offset: 0x2017c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc15_dt_nooverride_1**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC15_DT_NOOVERRIDE_CTRL_2_0

Offset: 0x8060
Byte Offset: 0x20180
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc15_dt_nooverride_2**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC15_DT_NOOVERRIDE_CTRL_3_0

Offset: 0x8061
Byte Offset: 0x20184
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc15_dt_nooverride_3**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC15_DT_NOOVERRIDE_CTRL_4_0

Offset: 0x8062
Byte Offset: 0x20188
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc15_dt_nooverride_4**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_2_VC15_DT_OVERRIDE_0

Offset: 0x8063
Byte Offset: 0x2018c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc15_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc15_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc15_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_2_PPFSM_TIMEOUT_CTRL_0

Offset: 0x8064
Byte Offset: 0x20190
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x7fffffff (0b0111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31	DISABLE	**cfg_timeout_en**: Enable Timeout counter for the PP FSM **0** = DISABLE **1** = ENABLE
30:0	0x7fffffff	**cfg_timeout_period**: Timeout period

NVCSI_STREAM_2_PH_CHK_CTRL_0

Offset: 0x8065
Byte Offset: 0x20194
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	ENABLE	**cfg_ph_16_vc**: 16 VC support **0** = DISABLE **1** = ENABLE
1	ENABLE	**cfg_ph_crc_chk_en**: PH CRC check enable (only for CPHY case), when this bit is set to 0, the packet header will still be decode when the CRC check fail, but the error will be set. **0** = DISABLE **1** = ENABLE
0	ENABLE	**cfg_ph_ecc_chk_en**: PH ECC check enable (only for DPHY case), when this bit is set to 0, the packet header will still be decode when the ECC check fail, but the error will be set. **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_2_VC0_DPCM_CTRL_0

Offset: 0x8066
Byte Offset: 0x20198
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc0_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_2_VC1_DPCM_CTRL_0

Offset: 0x8067
Byte Offset: 0x2019c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc1_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_2_VC2_DPCM_CTRL_0

Offset: 0x8068
Byte Offset: 0x201a0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc2_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_2_VC3_DPCM_CTRL_0

Offset: 0x8069
Byte Offset: 0x201a4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc3_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_2_VC4_DPCM_CTRL_0

Offset: 0x806a
Byte Offset: 0x201a8
Read/Write: R/W
Parity Protection: Y
Shadow: N

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc4_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_2_VC5_DPCM_CTRL_0

Offset: 0x806b
Byte Offset: 0x201ac
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc5_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_2_VC6_DPCM_CTRL_0

Offset: 0x806c
Byte Offset: 0x201b0
Read/Write: R/W
Parity Protection: Y

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc6_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_2_VC7_DPCM_CTRL_0

Offset: 0x806d
Byte Offset: 0x201b4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc7_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_2_VC8_DPCM_CTRL_0

Offset: 0x806e
Byte Offset: 0x201b8
Read/Write: R/W

Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc8_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_2_VC9_DPCM_CTRL_0

Offset: 0x806f
Byte Offset: 0x201bc
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc9_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_2_VC10_DPCM_CTRL_0

Offset: 0x8070
Byte Offset: 0x201c0

Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc10_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_2_VC11_DPCM_CTRL_0

Offset: 0x8071
Byte Offset: 0x201c4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc11_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_2_VC12_DPCM_CTRL_0

Offset: 0x8072

Byte Offset: 0x201c8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc12_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_2_VC13_DPCM_CTRL_0

Offset: 0x8073
Byte Offset: 0x201cc
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc13_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_2_VC14_DPCM_CTRL_0

Offset: 0x8074
Byte Offset: 0x201d0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc14_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_2_VC15_DPCM_CTRL_0

Offset: 0x8075
Byte Offset: 0x201d4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc15_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_2_PF_CRC_0
Status register on packet data CRC
Offset: 0x8076
Byte Offset: 0x201d8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**calc_crc**: 16 bit CRC computed over current packet (has to match with the CRC in PF for a good packet)
15:0	0x0	**rx_crc**: 16 bit CRC from PF

NVCSI_STREAM_2_PH_WC_0
Status register on WC
Offset: 0x8077
Byte Offset: 0x201dc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**calc_wc**: 16 bit WC computed over current packet (has to match with the WC in PH for a good packet)
15:0	0x0	**rx_wc**: 16 bit WC from PH

NVCSI_STREAM_2_PH_DI_0
Status register on Data ID
Offset: 0x8078
Byte Offset: 0x201e0
Read/Write: RO
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9:6	0x0	**rx_vc**: 4 bit VC from PH
5:0	0x0	**rx_dt**: 6 bit DTYPE from PH

NVCSI_STREAM_2_ERROR_STATUS2VI_MASK_0

Offset: 0x8079
Byte Offset: 0x201e4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15	0x0	**cfg_err_status2vi_mask_vc15**: for VC15 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
14	0x0	**cfg_err_status2vi_mask_vc14**: for VC14 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
13	0x0	**cfg_err_status2vi_mask_vc13**: for VC13 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
12	0x0	**cfg_err_status2vi_mask_vc12**: for VC12 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
11	0x0	**cfg_err_status2vi_mask_vc11**: for VC11 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
10	0x0	**cfg_err_status2vi_mask_vc10**: for VC10 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
9	0x0	**cfg_err_status2vi_mask_vc9**: for VC9 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF

Bit	Reset	Description
8	0x0	**cfg_err_status2vi_mask_vc8**: for VC8 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
7	0x0	**cfg_err_status2vi_mask_vc7**: for VC7 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
6	0x0	**cfg_err_status2vi_mask_vc6**: for VC6 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
5	0x0	**cfg_err_status2vi_mask_vc5**: for VC5 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
4	0x0	**cfg_err_status2vi_mask_vc4**: for VC4 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
3	0x0	**cfg_err_status2vi_mask_vc3**: for VC3 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
2	0x0	**cfg_err_status2vi_mask_vc2**: for VC2 : 0 = err_status2vi_vc2 will be send to VI at EOF; 1 = No error will be send to VI at EOF
1	0x0	**cfg_err_status2vi_mask_vc1**: for VC1 : 0 = err_status2vi_vc1 will be send to VI at EOF; 1 = No error will be send to VI at EOF
0	0x0	**cfg_err_status2vi_mask_vc0**: for VC0 : 0 = err_status2vi_vc0 will be send to VI at EOF; 1 = No error will be send to VI at EOF

NVCSI_STREAM_2_ERROR_STATUS2VI_VC0_0

This is RO register for SW
Offset: 0x807a
Byte Offset: 0x201e8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc0**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_2_ERROR_STATUS2VI_VC1_0

This is RO register for SW
Offset: 0x807b
Byte Offset: 0x201ec
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc1**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_2_ERROR_STATUS2VI_VC2_0

This is RO register for SW
Offset: 0x807c
Byte Offset: 0x201f0
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc2**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_2_ERROR_STATUS2VI_VC3_0

This is RO register for SW
Offset: 0x807d
Byte Offset: 0x201f4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc3**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_2_ERROR_STATUS2VI_VC4_0

This is RO register for SW
Offset: 0x807e
Byte Offset: 0x201f8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc4**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_2_ERROR_STATUS2VI_VC5_0

This is RO register for SW
Offset: 0x807f
Byte Offset: 0x201fc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc5**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_2_ERROR_STATUS2VI_VC6_0

This is RO register for SW
Offset: 0x8080
Byte Offset: 0x20200
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc6**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_2_ERROR_STATUS2VI_VC7_0

This is RO register for SW
Offset: 0x8081
Byte Offset: 0x20204
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc7**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_2_ERROR_STATUS2VI_VC8_0

This is RO register for SW
Offset: 0x8082
Byte Offset: 0x20208
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc8**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_2_ERROR_STATUS2VI_VC9_0

This is RO register for SW
Offset: 0x8083
Byte Offset: 0x2020c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc9**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_2_ERROR_STATUS2VI_VC10_0

This is RO register for SW
Offset: 0x8084
Byte Offset: 0x20210
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc10**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_2_ERROR_STATUS2VI_VC11_0

This is RO register for SW
Offset: 0x8085
Byte Offset: 0x20214
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc11**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_2_ERROR_STATUS2VI_VC12_0

This is RO register for SW
Offset: 0x8086
Byte Offset: 0x20218
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc12**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_2_ERROR_STATUS2VI_VC13_0

This is RO register for SW
Offset: 0x8087
Byte Offset: 0x2021c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc13**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_2_ERROR_STATUS2VI_VC14_0

This is RO register for SW
Offset: 0x8088
Byte Offset: 0x20220
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc14**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_2_ERROR_STATUS2VI_VC15_0

This is RO register for SW
Offset: 0x8089
Byte Offset: 0x20224
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc15**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_2_INTR_STATUS_0

Offset: 0x808a
Byte Offset: 0x20228
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**intr_stat_novc**: Not VC related interrupt
15	0x0	**intr_stat_vc15**: VC15 event

Bit	Reset	Description
14	0x0	**intr_stat_vc14**: VC14 event
13	0x0	**intr_stat_vc13**: VC13 event
12	0x0	**intr_stat_vc12**: VC12 event
11	0x0	**intr_stat_vc11**: VC11 event
10	0x0	**intr_stat_vc10**: VC10 event
9	0x0	**intr_stat_vc9**: VC9 event
8	0x0	**intr_stat_vc8**: VC8 event
7	0x0	**intr_stat_vc7**: VC7 event
6	0x0	**intr_stat_vc6**: VC6 event
5	0x0	**intr_stat_vc5**: VC5 event
4	0x0	**intr_stat_vc4**: VC4 event
3	0x0	**intr_stat_vc3**: VC3 event
2	0x0	**intr_stat_vc2**: VC2 event
1	0x0	**intr_stat_vc1**: VC1 event
0	0x0	**intr_stat_vc0**: VC0 event

NVCSI_STREAM_2_INTR_STATUS_NOVC_0

Offset: 0x808b
Byte Offset: 0x2022c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**intr_stat_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**intr_stat_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_2_INTR_STATUS_VC0_0

Offset: 0x808c
Byte Offset: 0x20230
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc0**
4	0x0	**intr_stat_ph_single_crc_err_vc0**
3	0x0	**intr_stat_pd_wc_short_err_vc0**
2	0x0	**intr_stat_pd_crc_err_vc0**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc0**
0	0x0	**intr_stat_ppfsm_timeout_vc0**

NVCSI_STREAM_2_INTR_STATUS_VC1_0

Offset: 0x808d
Byte Offset: 0x20234
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc1**

Bit	Reset	Description
4	0x0	**intr_stat_ph_single_crc_err_vc1**
3	0x0	**intr_stat_pd_wc_short_err_vc1**
2	0x0	**intr_stat_pd_crc_err_vc1**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc1**
0	0x0	**intr_stat_ppfsm_timeout_vc1**

NVCSI_STREAM_2_INTR_STATUS_VC2_0

Offset: 0x808e
Byte Offset: 0x20238
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc2**
4	0x0	**intr_stat_ph_single_crc_err_vc2**
3	0x0	**intr_stat_pd_wc_short_err_vc2**
2	0x0	**intr_stat_pd_crc_err_vc2**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc2**
0	0x0	**intr_stat_ppfsm_timeout_vc2**

NVCSI_STREAM_2_INTR_STATUS_VC3_0

Offset: 0x808f
Byte Offset: 0x2023c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc3**
4	0x0	**intr_stat_ph_single_crc_err_vc3**
3	0x0	**intr_stat_pd_wc_short_err_vc3**
2	0x0	**intr_stat_pd_crc_err_vc3**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc3**
0	0x0	**intr_stat_ppfsm_timeout_vc3**

NVCSI_STREAM_2_INTR_STATUS_VC4_0

Offset: 0x8090
Byte Offset: 0x20240
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc4**
4	0x0	**intr_stat_ph_single_crc_err_vc4**
3	0x0	**intr_stat_pd_wc_short_err_vc4**
2	0x0	**intr_stat_pd_crc_err_vc4**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc4**
0	0x0	**intr_stat_ppfsm_timeout_vc4**

NVCSI_STREAM_2_INTR_STATUS_VC5_0

Offset: 0x8091
Byte Offset: 0x20244
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc5**
4	0x0	**intr_stat_ph_single_crc_err_vc5**
3	0x0	**intr_stat_pd_wc_short_err_vc5**
2	0x0	**intr_stat_pd_crc_err_vc5**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc5**
0	0x0	**intr_stat_ppfsm_timeout_vc5**

NVCSI_STREAM_2_INTR_STATUS_VC6_0

Offset: 0x8092
Byte Offset: 0x20248
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc6**
4	0x0	**intr_stat_ph_single_crc_err_vc6**
3	0x0	**intr_stat_pd_wc_short_err_vc6**
2	0x0	**intr_stat_pd_crc_err_vc6**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc6**
0	0x0	**intr_stat_ppfsm_timeout_vc6**

NVCSI_STREAM_2_INTR_STATUS_VC7_0

Offset: 0x8093
Byte Offset: 0x2024c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc7**
4	0x0	**intr_stat_ph_single_crc_err_vc7**
3	0x0	**intr_stat_pd_wc_short_err_vc7**
2	0x0	**intr_stat_pd_crc_err_vc7**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc7**
0	0x0	**intr_stat_ppfsm_timeout_vc7**

NVCSI_STREAM_2_INTR_STATUS_VC8_0

Offset: 0x8094
Byte Offset: 0x20250
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc8**
4	0x0	**intr_stat_ph_single_crc_err_vc8**
3	0x0	**intr_stat_pd_wc_short_err_vc8**
2	0x0	**intr_stat_pd_crc_err_vc8**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc8**
0	0x0	**intr_stat_ppfsm_timeout_vc8**

NVCSI_STREAM_2_INTR_STATUS_VC9_0

Offset: 0x8095
Byte Offset: 0x20254
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc9**
4	0x0	**intr_stat_ph_single_crc_err_vc9**
3	0x0	**intr_stat_pd_wc_short_err_vc9**
2	0x0	**intr_stat_pd_crc_err_vc9**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc9**
0	0x0	**intr_stat_ppfsm_timeout_vc9**

NVCSI_STREAM_2_INTR_STATUS_VC10_0

Offset: 0x8096
Byte Offset: 0x20258
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc10**
4	0x0	**intr_stat_ph_single_crc_err_vc10**
3	0x0	**intr_stat_pd_wc_short_err_vc10**
2	0x0	**intr_stat_pd_crc_err_vc10**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc10**
0	0x0	**intr_stat_ppfsm_timeout_vc10**

NVCSI_STREAM_2_INTR_STATUS_VC11_0

Offset: 0x8097
Byte Offset: 0x2025c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc11**
4	0x0	**intr_stat_ph_single_crc_err_vc11**
3	0x0	**intr_stat_pd_wc_short_err_vc11**
2	0x0	**intr_stat_pd_crc_err_vc11**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc11**
0	0x0	**intr_stat_ppfsm_timeout_vc11**

NVCSI_STREAM_2_INTR_STATUS_VC12_0

Offset: 0x8098
Byte Offset: 0x20260
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc12**
4	0x0	**intr_stat_ph_single_crc_err_vc12**
3	0x0	**intr_stat_pd_wc_short_err_vc12**
2	0x0	**intr_stat_pd_crc_err_vc12**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc12**
0	0x0	**intr_stat_ppfsm_timeout_vc12**

NVCSI_STREAM_2_INTR_STATUS_VC13_0

Offset: 0x8099
Byte Offset: 0x20264
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc13**
4	0x0	**intr_stat_ph_single_crc_err_vc13**
3	0x0	**intr_stat_pd_wc_short_err_vc13**
2	0x0	**intr_stat_pd_crc_err_vc13**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc13**
0	0x0	**intr_stat_ppfsm_timeout_vc13**

NVCSI_STREAM_2_INTR_STATUS_VC14_0

Offset: 0x809a
Byte Offset: 0x20268
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc14**
4	0x0	**intr_stat_ph_single_crc_err_vc14**
3	0x0	**intr_stat_pd_wc_short_err_vc14**
2	0x0	**intr_stat_pd_crc_err_vc14**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc14**
0	0x0	**intr_stat_ppfsm_timeout_vc14**

NVCSI_STREAM_2_INTR_STATUS_VC15_0

Offset: 0x809b
Byte Offset: 0x2026c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc15**
4	0x0	**intr_stat_ph_single_crc_err_vc15**
3	0x0	**intr_stat_pd_wc_short_err_vc15**
2	0x0	**intr_stat_pd_crc_err_vc15**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc15**
0	0x0	**intr_stat_ppfsm_timeout_vc15**

NVCSI_STREAM_2_INTR_MASK_NOVC_0

Offset: 0x809c
Byte Offset: 0x20270
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**intr_mask_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**intr_mask_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_2_INTR_MASK_VC0_0

Offset: 0x809d
Byte Offset: 0x20274
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc0**
4	0x0	**intr_mask_ph_single_crc_err_vc0**

Bit	Reset	Description
3	0x0	**intr_mask_pd_wc_short_err_vc0**
2	0x0	**intr_mask_pd_crc_err_vc0**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc0**
0	0x0	**intr_mask_ppfsm_timeout_vc0**

NVCSI_STREAM_2_INTR_MASK_VC1_0

Offset: 0x809e
Byte Offset: 0x20278
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc1**
4	0x0	**intr_mask_ph_single_crc_err_vc1**
3	0x0	**intr_mask_pd_wc_short_err_vc1**
2	0x0	**intr_mask_pd_crc_err_vc1**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc1**
0	0x0	**intr_mask_ppfsm_timeout_vc1**

NVCSI_STREAM_2_INTR_MASK_VC2_0

Offset: 0x809f
Byte Offset: 0x2027c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc2**

Bit	Reset	Description
4	0x0	**intr_mask_ph_single_crc_err_vc2**
3	0x0	**intr_mask_pd_wc_short_err_vc2**
2	0x0	**intr_mask_pd_crc_err_vc2**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc2**
0	0x0	**intr_mask_ppfsm_timeout_vc2**

NVCSI_STREAM_2_INTR_MASK_VC3_0

Offset: 0x80a0
Byte Offset: 0x20280
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc3**
4	0x0	**intr_mask_ph_single_crc_err_vc3**
3	0x0	**intr_mask_pd_wc_short_err_vc3**
2	0x0	**intr_mask_pd_crc_err_vc3**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc3**
0	0x0	**intr_mask_ppfsm_timeout_vc3**

NVCSI_STREAM_2_INTR_MASK_VC4_0

Offset: 0x80a1
Byte Offset: 0x20284
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc4**
4	0x0	**intr_mask_ph_single_crc_err_vc4**
3	0x0	**intr_mask_pd_wc_short_err_vc4**
2	0x0	**intr_mask_pd_crc_err_vc4**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc4**
0	0x0	**intr_mask_ppfsm_timeout_vc4**

NVCSI_STREAM_2_INTR_MASK_VC5_0

Offset: 0x80a2
Byte Offset: 0x20288
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc5**
4	0x0	**intr_mask_ph_single_crc_err_vc5**
3	0x0	**intr_mask_pd_wc_short_err_vc5**
2	0x0	**intr_mask_pd_crc_err_vc5**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc5**
0	0x0	**intr_mask_ppfsm_timeout_vc5**

NVCSI_STREAM_2_INTR_MASK_VC6_0

Offset: 0x80a3
Byte Offset: 0x2028c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc6**
4	0x0	**intr_mask_ph_single_crc_err_vc6**
3	0x0	**intr_mask_pd_wc_short_err_vc6**
2	0x0	**intr_mask_pd_crc_err_vc6**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc6**
0	0x0	**intr_mask_ppfsm_timeout_vc6**

NVCSI_STREAM_2_INTR_MASK_VC7_0

Offset: 0x80a4
Byte Offset: 0x20290
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc7**
4	0x0	**intr_mask_ph_single_crc_err_vc7**
3	0x0	**intr_mask_pd_wc_short_err_vc7**
2	0x0	**intr_mask_pd_crc_err_vc7**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc7**
0	0x0	**intr_mask_ppfsm_timeout_vc7**

NVCSI_STREAM_2_INTR_MASK_VC8_0

Offset: 0x80a5
Byte Offset: 0x20294
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc8**
4	0x0	**intr_mask_ph_single_crc_err_vc8**
3	0x0	**intr_mask_pd_wc_short_err_vc8**
2	0x0	**intr_mask_pd_crc_err_vc8**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc8**
0	0x0	**intr_mask_ppfsm_timeout_vc8**

NVCSI_STREAM_2_INTR_MASK_VC9_0

Offset: 0x80a6
Byte Offset: 0x20298
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc9**
4	0x0	**intr_mask_ph_single_crc_err_vc9**
3	0x0	**intr_mask_pd_wc_short_err_vc9**
2	0x0	**intr_mask_pd_crc_err_vc9**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc9**
0	0x0	**intr_mask_ppfsm_timeout_vc9**

NVCSI_STREAM_2_INTR_MASK_VC10_0

Offset: 0x80a7
Byte Offset: 0x2029c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc10**
4	0x0	**intr_mask_ph_single_crc_err_vc10**
3	0x0	**intr_mask_pd_wc_short_err_vc10**
2	0x0	**intr_mask_pd_crc_err_vc10**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc10**
0	0x0	**intr_mask_ppfsm_timeout_vc10**

NVCSI_STREAM_2_INTR_MASK_VC11_0

Offset: 0x80a8
Byte Offset: 0x202a0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc11**
4	0x0	**intr_mask_ph_single_crc_err_vc11**
3	0x0	**intr_mask_pd_wc_short_err_vc11**
2	0x0	**intr_mask_pd_crc_err_vc11**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc11**
0	0x0	**intr_mask_ppfsm_timeout_vc11**

NVCSI_STREAM_2_INTR_MASK_VC12_0

Offset: 0x80a9
Byte Offset: 0x202a4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc12**
4	0x0	**intr_mask_ph_single_crc_err_vc12**
3	0x0	**intr_mask_pd_wc_short_err_vc12**
2	0x0	**intr_mask_pd_crc_err_vc12**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc12**
0	0x0	**intr_mask_ppfsm_timeout_vc12**

NVCSI_STREAM_2_INTR_MASK_VC13_0

Offset: 0x80aa
Byte Offset: 0x202a8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc13**
4	0x0	**intr_mask_ph_single_crc_err_vc13**
3	0x0	**intr_mask_pd_wc_short_err_vc13**
2	0x0	**intr_mask_pd_crc_err_vc13**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc13**
0	0x0	**intr_mask_ppfsm_timeout_vc13**

NVCSI_STREAM_2_INTR_MASK_VC14_0

Offset: 0x80ab
Byte Offset: 0x202ac
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc14**
4	0x0	**intr_mask_ph_single_crc_err_vc14**
3	0x0	**intr_mask_pd_wc_short_err_vc14**
2	0x0	**intr_mask_pd_crc_err_vc14**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc14**
0	0x0	**intr_mask_ppfsm_timeout_vc14**

NVCSI_STREAM_2_INTR_MASK_VC15_0

Offset: 0x80ac
Byte Offset: 0x202b0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc15**
4	0x0	**intr_mask_ph_single_crc_err_vc15**
3	0x0	**intr_mask_pd_wc_short_err_vc15**
2	0x0	**intr_mask_pd_crc_err_vc15**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc15**
0	0x0	**intr_mask_ppfsm_timeout_vc15**

NVCSI_STREAM_2_CORRECTABLE_ERR_INTR_STATUS_0

Offset: 0x80ad
Byte Offset: 0x202b4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**err_intr_stat_novc**: Not VC related interrupt
15	0x0	**err_intr_stat_vc15**: VC15 event
14	0x0	**err_intr_stat_vc14**: VC14 event
13	0x0	**err_intr_stat_vc13**: VC13 event
12	0x0	**err_intr_stat_vc12**: VC12 event
11	0x0	**err_intr_stat_vc11**: VC11 event
10	0x0	**err_intr_stat_vc10**: VC10 event
9	0x0	**err_intr_stat_vc9**: VC9 event
8	0x0	**err_intr_stat_vc8**: VC8 event
7	0x0	**err_intr_stat_vc7**: VC7 event
6	0x0	**err_intr_stat_vc6**: VC6 event
5	0x0	**err_intr_stat_vc5**: VC5 event
4	0x0	**err_intr_stat_vc4**: VC4 event
3	0x0	**err_intr_stat_vc3**: VC3 event
2	0x0	**err_intr_stat_vc2**: VC2 event
1	0x0	**err_intr_stat_vc1**: VC1 event
0	0x0	**err_intr_stat_vc0**: VC0 event

NVCSI_STREAM_2_CORRECTABLE_ERR_INTR_STATUS_NOVC_0

Offset: 0x80ae
Byte Offset: 0x202b8
Read/Write: R/W

Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**err_intr_stat_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**err_intr_stat_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_2_CORRECTABLE_ERR_INTR_STATUS_VC0_0

Offset: 0x80af
Byte Offset: 0x202bc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc0**
4	0x0	**err_intr_stat_ph_single_crc_err_vc0**
3	0x0	**err_intr_stat_pd_wc_short_err_vc0**
2	0x0	**err_intr_stat_pd_crc_err_vc0**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc0**
0	0x0	**err_intr_stat_ppfsm_timeout_vc0**

NVCSI_STREAM_2_CORRECTABLE_ERR_INTR_STATUS_VC1_0

Offset: 0x80b0
Byte Offset: 0x202c0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc1**
4	0x0	**err_intr_stat_ph_single_crc_err_vc1**
3	0x0	**err_intr_stat_pd_wc_short_err_vc1**
2	0x0	**err_intr_stat_pd_crc_err_vc1**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc1**
0	0x0	**err_intr_stat_ppfsm_timeout_vc1**

NVCSI_STREAM_2_CORRECTABLE_ERR_INTR_STATUS_VC2_0

Offset: 0x80b1
Byte Offset: 0x202c4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc2**
4	0x0	**err_intr_stat_ph_single_crc_err_vc2**
3	0x0	**err_intr_stat_pd_wc_short_err_vc2**
2	0x0	**err_intr_stat_pd_crc_err_vc2**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc2**
0	0x0	**err_intr_stat_ppfsm_timeout_vc2**

NVCSI_STREAM_2_CORRECTABLE_ERR_INTR_STATUS_VC3_0

Offset: 0x80b2
Byte Offset: 0x202c8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc3**
4	0x0	**err_intr_stat_ph_single_crc_err_vc3**
3	0x0	**err_intr_stat_pd_wc_short_err_vc3**
2	0x0	**err_intr_stat_pd_crc_err_vc3**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc3**
0	0x0	**err_intr_stat_ppfsm_timeout_vc3**

NVCSI_STREAM_2_CORRECTABLE_ERR_INTR_STATUS_VC4_0

Offset: 0x80b3
Byte Offset: 0x202cc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc4**
4	0x0	**err_intr_stat_ph_single_crc_err_vc4**
3	0x0	**err_intr_stat_pd_wc_short_err_vc4**
2	0x0	**err_intr_stat_pd_crc_err_vc4**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc4**
0	0x0	**err_intr_stat_ppfsm_timeout_vc4**

NVCSI_STREAM_2_CORRECTABLE_ERR_INTR_STATUS_VC5_0

Offset: 0x80b4
Byte Offset: 0x202d0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc5**
4	0x0	**err_intr_stat_ph_single_crc_err_vc5**
3	0x0	**err_intr_stat_pd_wc_short_err_vc5**
2	0x0	**err_intr_stat_pd_crc_err_vc5**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc5**
0	0x0	**err_intr_stat_ppfsm_timeout_vc5**

NVCSI_STREAM_2_CORRECTABLE_ERR_INTR_STATUS_VC6_0

Offset: 0x80b5
Byte Offset: 0x202d4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc6**
4	0x0	**err_intr_stat_ph_single_crc_err_vc6**
3	0x0	**err_intr_stat_pd_wc_short_err_vc6**
2	0x0	**err_intr_stat_pd_crc_err_vc6**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc6**
0	0x0	**err_intr_stat_ppfsm_timeout_vc6**

NVCSI_STREAM_2_CORRECTABLE_ERR_INTR_STATUS_VC7_0

Offset: 0x80b6
Byte Offset: 0x202d8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc7**
4	0x0	**err_intr_stat_ph_single_crc_err_vc7**
3	0x0	**err_intr_stat_pd_wc_short_err_vc7**
2	0x0	**err_intr_stat_pd_crc_err_vc7**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc7**
0	0x0	**err_intr_stat_ppfsm_timeout_vc7**

NVCSI_STREAM_2_CORRECTABLE_ERR_INTR_STATUS_VC8_0

Offset: 0x80b7
Byte Offset: 0x202dc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc8**
4	0x0	**err_intr_stat_ph_single_crc_err_vc8**
3	0x0	**err_intr_stat_pd_wc_short_err_vc8**
2	0x0	**err_intr_stat_pd_crc_err_vc8**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc8**
0	0x0	**err_intr_stat_ppfsm_timeout_vc8**

NVCSI_STREAM_2_CORRECTABLE_ERR_INTR_STATUS_VC9_0

Offset: 0x80b8
Byte Offset: 0x202e0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc9**
4	0x0	**err_intr_stat_ph_single_crc_err_vc9**
3	0x0	**err_intr_stat_pd_wc_short_err_vc9**
2	0x0	**err_intr_stat_pd_crc_err_vc9**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc9**
0	0x0	**err_intr_stat_ppfsm_timeout_vc9**

NVCSI_STREAM_2_CORRECTABLE_ERR_INTR_STATUS_VC10_0

Offset: 0x80b9
Byte Offset: 0x202e4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc10**
4	0x0	**err_intr_stat_ph_single_crc_err_vc10**
3	0x0	**err_intr_stat_pd_wc_short_err_vc10**
2	0x0	**err_intr_stat_pd_crc_err_vc10**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc10**
0	0x0	**err_intr_stat_ppfsm_timeout_vc10**

NVCSI_STREAM_2_CORRECTABLE_ERR_INTR_STATUS_VC11_0

Offset: 0x80ba
Byte Offset: 0x202e8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc11**
4	0x0	**err_intr_stat_ph_single_crc_err_vc11**
3	0x0	**err_intr_stat_pd_wc_short_err_vc11**
2	0x0	**err_intr_stat_pd_crc_err_vc11**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc11**
0	0x0	**err_intr_stat_ppfsm_timeout_vc11**

NVCSI_STREAM_2_CORRECTABLE_ERR_INTR_STATUS_VC12_0

Offset: 0x80bb
Byte Offset: 0x202ec
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc12**
4	0x0	**err_intr_stat_ph_single_crc_err_vc12**
3	0x0	**err_intr_stat_pd_wc_short_err_vc12**
2	0x0	**err_intr_stat_pd_crc_err_vc12**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc12**
0	0x0	**err_intr_stat_ppfsm_timeout_vc12**

NVCSI_STREAM_2_CORRECTABLE_ERR_INTR_STATUS_VC13_0

Offset: 0x80bc
Byte Offset: 0x202f0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc13**
4	0x0	**err_intr_stat_ph_single_crc_err_vc13**
3	0x0	**err_intr_stat_pd_wc_short_err_vc13**
2	0x0	**err_intr_stat_pd_crc_err_vc13**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc13**
0	0x0	**err_intr_stat_ppfsm_timeout_vc13**

NVCSI_STREAM_2_CORRECTABLE_ERR_INTR_STATUS_VC14_0

Offset: 0x80bd
Byte Offset: 0x202f4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc14**
4	0x0	**err_intr_stat_ph_single_crc_err_vc14**
3	0x0	**err_intr_stat_pd_wc_short_err_vc14**
2	0x0	**err_intr_stat_pd_crc_err_vc14**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc14**
0	0x0	**err_intr_stat_ppfsm_timeout_vc14**

NVCSI_STREAM_2_CORRECTABLE_ERR_INTR_STATUS_VC15_0

Offset: 0x80be
Byte Offset: 0x202f8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc15**
4	0x0	**err_intr_stat_ph_single_crc_err_vc15**
3	0x0	**err_intr_stat_pd_wc_short_err_vc15**
2	0x0	**err_intr_stat_pd_crc_err_vc15**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc15**
0	0x0	**err_intr_stat_ppfsm_timeout_vc15**

NVCSI_STREAM_2_UNCORRECTABLE_ERR_INTR_STATUS_0

Offset: 0x80bf
Byte Offset: 0x202fc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**err_intr_stat_novc**: Not VC related interrupt
15	0x0	**err_intr_stat_vc15**: VC15 event
14	0x0	**err_intr_stat_vc14**: VC14 event
13	0x0	**err_intr_stat_vc13**: VC13 event
12	0x0	**err_intr_stat_vc12**: VC12 event
11	0x0	**err_intr_stat_vc11**: VC11 event
10	0x0	**err_intr_stat_vc10**: VC10 event
9	0x0	**err_intr_stat_vc9**: VC9 event
8	0x0	**err_intr_stat_vc8**: VC8 event
7	0x0	**err_intr_stat_vc7**: VC7 event

Bit	Reset	Description
6	0x0	**err_intr_stat_vc6**: VC6 event
5	0x0	**err_intr_stat_vc5**: VC5 event
4	0x0	**err_intr_stat_vc4**: VC4 event
3	0x0	**err_intr_stat_vc3**: VC3 event
2	0x0	**err_intr_stat_vc2**: VC2 event
1	0x0	**err_intr_stat_vc1**: VC1 event
0	0x0	**err_intr_stat_vc0**: VC0 event

NVCSI_STREAM_2_UNCORRECTABLE_ERR_INTR_STATUS_NOVC_0

Offset: 0x80c0
Byte Offset: 0x20300
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**err_intr_stat_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**err_intr_stat_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_2_UNCORRECTABLE_ERR_INTR_STATUS_VC0_0

Offset: 0x80c1
Byte Offset: 0x20304
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc0**
4	0x0	**err_intr_stat_ph_single_crc_err_vc0**
3	0x0	**err_intr_stat_pd_wc_short_err_vc0**
2	0x0	**err_intr_stat_pd_crc_err_vc0**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc0**
0	0x0	**err_intr_stat_ppfsm_timeout_vc0**

NVCSI_STREAM_2_UNCORRECTABLE_ERR_INTR_STATUS_VC1_0

Offset: 0x80c2
Byte Offset: 0x20308
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc1**
4	0x0	**err_intr_stat_ph_single_crc_err_vc1**
3	0x0	**err_intr_stat_pd_wc_short_err_vc1**
2	0x0	**err_intr_stat_pd_crc_err_vc1**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc1**
0	0x0	**err_intr_stat_ppfsm_timeout_vc1**

NVCSI_STREAM_2_UNCORRECTABLE_ERR_INTR_STATUS_VC2_0

Offset: 0x80c3
Byte Offset: 0x2030c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc2**
4	0x0	**err_intr_stat_ph_single_crc_err_vc2**
3	0x0	**err_intr_stat_pd_wc_short_err_vc2**
2	0x0	**err_intr_stat_pd_crc_err_vc2**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc2**
0	0x0	**err_intr_stat_ppfsm_timeout_vc2**

NVCSI_STREAM_2_UNCORRECTABLE_ERR_INTR_STATUS_VC3_0

Offset: 0x80c4
Byte Offset: 0x20310
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc3**
4	0x0	**err_intr_stat_ph_single_crc_err_vc3**
3	0x0	**err_intr_stat_pd_wc_short_err_vc3**
2	0x0	**err_intr_stat_pd_crc_err_vc3**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc3**
0	0x0	**err_intr_stat_ppfsm_timeout_vc3**

NVCSI_STREAM_2_UNCORRECTABLE_ERR_INTR_STATUS_VC4_0

Offset: 0x80c5
Byte Offset: 0x20314
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc4**
4	0x0	**err_intr_stat_ph_single_crc_err_vc4**
3	0x0	**err_intr_stat_pd_wc_short_err_vc4**
2	0x0	**err_intr_stat_pd_crc_err_vc4**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc4**
0	0x0	**err_intr_stat_ppfsm_timeout_vc4**

NVCSI_STREAM_2_UNCORRECTABLE_ERR_INTR_STATUS_VC5_0

Offset: 0x80c6
Byte Offset: 0x20318
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc5**
4	0x0	**err_intr_stat_ph_single_crc_err_vc5**
3	0x0	**err_intr_stat_pd_wc_short_err_vc5**
2	0x0	**err_intr_stat_pd_crc_err_vc5**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc5**
0	0x0	**err_intr_stat_ppfsm_timeout_vc5**

NVCSI_STREAM_2_UNCORRECTABLE_ERR_INTR_STATUS_VC6_0

Offset: 0x80c7
Byte Offset: 0x2031c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc6**
4	0x0	**err_intr_stat_ph_single_crc_err_vc6**
3	0x0	**err_intr_stat_pd_wc_short_err_vc6**
2	0x0	**err_intr_stat_pd_crc_err_vc6**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc6**
0	0x0	**err_intr_stat_ppfsm_timeout_vc6**

NVCSI_STREAM_2_UNCORRECTABLE_ERR_INTR_STATUS_VC7_0

Offset: 0x80c8
Byte Offset: 0x20320
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc7**
4	0x0	**err_intr_stat_ph_single_crc_err_vc7**
3	0x0	**err_intr_stat_pd_wc_short_err_vc7**
2	0x0	**err_intr_stat_pd_crc_err_vc7**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc7**
0	0x0	**err_intr_stat_ppfsm_timeout_vc7**

NVCSI_STREAM_2_UNCORRECTABLE_ERR_INTR_STATUS_VC8_0

Offset: 0x80c9
Byte Offset: 0x20324
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc8**
4	0x0	**err_intr_stat_ph_single_crc_err_vc8**
3	0x0	**err_intr_stat_pd_wc_short_err_vc8**
2	0x0	**err_intr_stat_pd_crc_err_vc8**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc8**
0	0x0	**err_intr_stat_ppfsm_timeout_vc8**

NVCSI_STREAM_2_UNCORRECTABLE_ERR_INTR_STATUS_VC9_0

Offset: 0x80ca
Byte Offset: 0x20328
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc9**
4	0x0	**err_intr_stat_ph_single_crc_err_vc9**
3	0x0	**err_intr_stat_pd_wc_short_err_vc9**
2	0x0	**err_intr_stat_pd_crc_err_vc9**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc9**
0	0x0	**err_intr_stat_ppfsm_timeout_vc9**

NVCSI_STREAM_2_UNCORRECTABLE_ERR_INTR_STATUS_VC10_0

Offset: 0x80cb
Byte Offset: 0x2032c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc10**
4	0x0	**err_intr_stat_ph_single_crc_err_vc10**
3	0x0	**err_intr_stat_pd_wc_short_err_vc10**
2	0x0	**err_intr_stat_pd_crc_err_vc10**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc10**
0	0x0	**err_intr_stat_ppfsm_timeout_vc10**

NVCSI_STREAM_2_UNCORRECTABLE_ERR_INTR_STATUS_VC11_0

Offset: 0x80cc
Byte Offset: 0x20330
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc11**
4	0x0	**err_intr_stat_ph_single_crc_err_vc11**
3	0x0	**err_intr_stat_pd_wc_short_err_vc11**
2	0x0	**err_intr_stat_pd_crc_err_vc11**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc11**
0	0x0	**err_intr_stat_ppfsm_timeout_vc11**

NVCSI_STREAM_2_UNCORRECTABLE_ERR_INTR_STATUS_VC12_0

Offset: 0x80cd
Byte Offset: 0x20334
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc12**
4	0x0	**err_intr_stat_ph_single_crc_err_vc12**
3	0x0	**err_intr_stat_pd_wc_short_err_vc12**
2	0x0	**err_intr_stat_pd_crc_err_vc12**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc12**
0	0x0	**err_intr_stat_ppfsm_timeout_vc12**

NVCSI_STREAM_2_UNCORRECTABLE_ERR_INTR_STATUS_VC13_0

Offset: 0x80ce
Byte Offset: 0x20338
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc13**
4	0x0	**err_intr_stat_ph_single_crc_err_vc13**
3	0x0	**err_intr_stat_pd_wc_short_err_vc13**
2	0x0	**err_intr_stat_pd_crc_err_vc13**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc13**
0	0x0	**err_intr_stat_ppfsm_timeout_vc13**

NVCSI_STREAM_2_UNCORRECTABLE_ERR_INTR_STATUS_VC14_0

Offset: 0x80cf
Byte Offset: 0x2033c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc14**
4	0x0	**err_intr_stat_ph_single_crc_err_vc14**
3	0x0	**err_intr_stat_pd_wc_short_err_vc14**
2	0x0	**err_intr_stat_pd_crc_err_vc14**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc14**
0	0x0	**err_intr_stat_ppfsm_timeout_vc14**

NVCSI_STREAM_2_UNCORRECTABLE_ERR_INTR_STATUS_VC15_0

Offset: 0x80d0
Byte Offset: 0x20340
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc15**
4	0x0	**err_intr_stat_ph_single_crc_err_vc15**
3	0x0	**err_intr_stat_pd_wc_short_err_vc15**
2	0x0	**err_intr_stat_pd_crc_err_vc15**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc15**
0	0x0	**err_intr_stat_ppfsm_timeout_vc15**

NVCSI_STREAM_2_ERR_INTR_MASK_NOVC_0

Offset: 0x80d1
Byte Offset: 0x20344
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**err_intr_mask_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**err_intr_mask_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_2_ERR_INTR_MASK_VC0_0

Offset: 0x80d2
Byte Offset: 0x20348
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc0**
4	0x0	**err_intr_mask_ph_single_crc_err_vc0**
3	0x0	**err_intr_mask_pd_wc_short_err_vc0**
2	0x0	**err_intr_mask_pd_crc_err_vc0**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc0**
0	0x0	**err_intr_mask_ppfsm_timeout_vc0**

NVCSI_STREAM_2_ERR_INTR_MASK_VC1_0

Offset: 0x80d3
Byte Offset: 0x2034c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc1**

Bit	Reset	Description
4	0x0	**err_intr_mask_ph_single_crc_err_vc1**
3	0x0	**err_intr_mask_pd_wc_short_err_vc1**
2	0x0	**err_intr_mask_pd_crc_err_vc1**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc1**
0	0x0	**err_intr_mask_ppfsm_timeout_vc1**

NVCSI_STREAM_2_ERR_INTR_MASK_VC2_0

Offset: 0x80d4
Byte Offset: 0x20350
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc2**
4	0x0	**err_intr_mask_ph_single_crc_err_vc2**
3	0x0	**err_intr_mask_pd_wc_short_err_vc2**
2	0x0	**err_intr_mask_pd_crc_err_vc2**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc2**
0	0x0	**err_intr_mask_ppfsm_timeout_vc2**

NVCSI_STREAM_2_ERR_INTR_MASK_VC3_0

Offset: 0x80d5
Byte Offset: 0x20354
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc3**
4	0x0	**err_intr_mask_ph_single_crc_err_vc3**
3	0x0	**err_intr_mask_pd_wc_short_err_vc3**
2	0x0	**err_intr_mask_pd_crc_err_vc3**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc3**
0	0x0	**err_intr_mask_ppfsm_timeout_vc3**

NVCSI_STREAM_2_ERR_INTR_MASK_VC4_0

Offset: 0x80d6
Byte Offset: 0x20358
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc4**
4	0x0	**err_intr_mask_ph_single_crc_err_vc4**
3	0x0	**err_intr_mask_pd_wc_short_err_vc4**
2	0x0	**err_intr_mask_pd_crc_err_vc4**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc4**
0	0x0	**err_intr_mask_ppfsm_timeout_vc4**

NVCSI_STREAM_2_ERR_INTR_MASK_VC5_0

Offset: 0x80d7
Byte Offset: 0x2035c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc5**
4	0x0	**err_intr_mask_ph_single_crc_err_vc5**
3	0x0	**err_intr_mask_pd_wc_short_err_vc5**
2	0x0	**err_intr_mask_pd_crc_err_vc5**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc5**
0	0x0	**err_intr_mask_ppfsm_timeout_vc5**

NVCSI_STREAM_2_ERR_INTR_MASK_VC6_0

Offset: 0x80d8
Byte Offset: 0x20360
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc6**
4	0x0	**err_intr_mask_ph_single_crc_err_vc6**
3	0x0	**err_intr_mask_pd_wc_short_err_vc6**
2	0x0	**err_intr_mask_pd_crc_err_vc6**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc6**
0	0x0	**err_intr_mask_ppfsm_timeout_vc6**

NVCSI_STREAM_2_ERR_INTR_MASK_VC7_0

Offset: 0x80d9
Byte Offset: 0x20364
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc7**
4	0x0	**err_intr_mask_ph_single_crc_err_vc7**
3	0x0	**err_intr_mask_pd_wc_short_err_vc7**
2	0x0	**err_intr_mask_pd_crc_err_vc7**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc7**
0	0x0	**err_intr_mask_ppfsm_timeout_vc7**

NVCSI_STREAM_2_ERR_INTR_MASK_VC8_0

Offset: 0x80da
Byte Offset: 0x20368
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc8**
4	0x0	**err_intr_mask_ph_single_crc_err_vc8**
3	0x0	**err_intr_mask_pd_wc_short_err_vc8**
2	0x0	**err_intr_mask_pd_crc_err_vc8**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc8**
0	0x0	**err_intr_mask_ppfsm_timeout_vc8**

NVCSI_STREAM_2_ERR_INTR_MASK_VC9_0

Offset: 0x80db
Byte Offset: 0x2036c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc9**
4	0x0	**err_intr_mask_ph_single_crc_err_vc9**
3	0x0	**err_intr_mask_pd_wc_short_err_vc9**
2	0x0	**err_intr_mask_pd_crc_err_vc9**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc9**
0	0x0	**err_intr_mask_ppfsm_timeout_vc9**

NVCSI_STREAM_2_ERR_INTR_MASK_VC10_0

Offset: 0x80dc
Byte Offset: 0x20370
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc10**
4	0x0	**err_intr_mask_ph_single_crc_err_vc10**
3	0x0	**err_intr_mask_pd_wc_short_err_vc10**
2	0x0	**err_intr_mask_pd_crc_err_vc10**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc10**
0	0x0	**err_intr_mask_ppfsm_timeout_vc10**

NVCSI_STREAM_2_ERR_INTR_MASK_VC11_0

Offset: 0x80dd
Byte Offset: 0x20374
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc11**
4	0x0	**err_intr_mask_ph_single_crc_err_vc11**
3	0x0	**err_intr_mask_pd_wc_short_err_vc11**
2	0x0	**err_intr_mask_pd_crc_err_vc11**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc11**
0	0x0	**err_intr_mask_ppfsm_timeout_vc11**

NVCSI_STREAM_2_ERR_INTR_MASK_VC12_0

Offset: 0x80de
Byte Offset: 0x20378
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc12**
4	0x0	**err_intr_mask_ph_single_crc_err_vc12**
3	0x0	**err_intr_mask_pd_wc_short_err_vc12**
2	0x0	**err_intr_mask_pd_crc_err_vc12**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc12**
0	0x0	**err_intr_mask_ppfsm_timeout_vc12**

NVCSI_STREAM_2_ERR_INTR_MASK_VC13_0

Offset: 0x80df
Byte Offset: 0x2037c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc13**
4	0x0	**err_intr_mask_ph_single_crc_err_vc13**
3	0x0	**err_intr_mask_pd_wc_short_err_vc13**
2	0x0	**err_intr_mask_pd_crc_err_vc13**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc13**
0	0x0	**err_intr_mask_ppfsm_timeout_vc13**

NVCSI_STREAM_2_ERR_INTR_MASK_VC14_0

Offset: 0x80e0
Byte Offset: 0x20380
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc14**
4	0x0	**err_intr_mask_ph_single_crc_err_vc14**
3	0x0	**err_intr_mask_pd_wc_short_err_vc14**
2	0x0	**err_intr_mask_pd_crc_err_vc14**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc14**
0	0x0	**err_intr_mask_ppfsm_timeout_vc14**

NVCSI_STREAM_2_ERR_INTR_MASK_VC15_0

Offset: 0x80e1
Byte Offset: 0x20384
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc15**
4	0x0	**err_intr_mask_ph_single_crc_err_vc15**
3	0x0	**err_intr_mask_pd_wc_short_err_vc15**
2	0x0	**err_intr_mask_pd_crc_err_vc15**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc15**
0	0x0	**err_intr_mask_ppfsm_timeout_vc15**

NVCSI_STREAM_2_ERR_INTR_TYPE_NOVC_0

Offset: 0x80e2
Byte Offset: 0x20388
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**err_intr_type_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**err_intr_type_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_2_ERR_INTR_TYPE_VC0_0

Offset: 0x80e3
Byte Offset: 0x2038c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc0**
4	0x0	**err_intr_type_ph_single_crc_err_vc0**

Bit	Reset	Description
3	0x0	**err_intr_type_pd_wc_short_err_vc0**
2	0x0	**err_intr_type_pd_crc_err_vc0**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc0**
0	0x0	**err_intr_type_ppfsm_timeout_vc0**

NVCSI_STREAM_2_ERR_INTR_TYPE_VC1_0

Offset: 0x80e4
Byte Offset: 0x20390
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc1**
4	0x0	**err_intr_type_ph_single_crc_err_vc1**
3	0x0	**err_intr_type_pd_wc_short_err_vc1**
2	0x0	**err_intr_type_pd_crc_err_vc1**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc1**
0	0x0	**err_intr_type_ppfsm_timeout_vc1**

NVCSI_STREAM_2_ERR_INTR_TYPE_VC2_0

Offset: 0x80e5
Byte Offset: 0x20394
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc2**

Bit	Reset	Description
4	0x0	**err_intr_type_ph_single_crc_err_vc2**
3	0x0	**err_intr_type_pd_wc_short_err_vc2**
2	0x0	**err_intr_type_pd_crc_err_vc2**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc2**
0	0x0	**err_intr_type_ppfsm_timeout_vc2**

NVCSI_STREAM_2_ERR_INTR_TYPE_VC3_0

Offset: 0x80e6
Byte Offset: 0x20398
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc3**
4	0x0	**err_intr_type_ph_single_crc_err_vc3**
3	0x0	**err_intr_type_pd_wc_short_err_vc3**
2	0x0	**err_intr_type_pd_crc_err_vc3**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc3**
0	0x0	**err_intr_type_ppfsm_timeout_vc3**

NVCSI_STREAM_2_ERR_INTR_TYPE_VC4_0

Offset: 0x80e7
Byte Offset: 0x2039c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc4**
4	0x0	**err_intr_type_ph_single_crc_err_vc4**
3	0x0	**err_intr_type_pd_wc_short_err_vc4**
2	0x0	**err_intr_type_pd_crc_err_vc4**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc4**
0	0x0	**err_intr_type_ppfsm_timeout_vc4**

NVCSI_STREAM_2_ERR_INTR_TYPE_VC5_0

Offset: 0x80e8
Byte Offset: 0x203a0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc5**
4	0x0	**err_intr_type_ph_single_crc_err_vc5**
3	0x0	**err_intr_type_pd_wc_short_err_vc5**
2	0x0	**err_intr_type_pd_crc_err_vc5**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc5**
0	0x0	**err_intr_type_ppfsm_timeout_vc5**

NVCSI_STREAM_2_ERR_INTR_TYPE_VC6_0

Offset: 0x80e9
Byte Offset: 0x203a4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc6**
4	0x0	**err_intr_type_ph_single_crc_err_vc6**
3	0x0	**err_intr_type_pd_wc_short_err_vc6**
2	0x0	**err_intr_type_pd_crc_err_vc6**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc6**
0	0x0	**err_intr_type_ppfsm_timeout_vc6**

NVCSI_STREAM_2_ERR_INTR_TYPE_VC7_0

Offset: 0x80ea
Byte Offset: 0x203a8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc7**
4	0x0	**err_intr_type_ph_single_crc_err_vc7**
3	0x0	**err_intr_type_pd_wc_short_err_vc7**
2	0x0	**err_intr_type_pd_crc_err_vc7**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc7**
0	0x0	**err_intr_type_ppfsm_timeout_vc7**

NVCSI_STREAM_2_ERR_INTR_TYPE_VC8_0

Offset: 0x80eb
Byte Offset: 0x203ac
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc8**
4	0x0	**err_intr_type_ph_single_crc_err_vc8**
3	0x0	**err_intr_type_pd_wc_short_err_vc8**
2	0x0	**err_intr_type_pd_crc_err_vc8**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc8**
0	0x0	**err_intr_type_ppfsm_timeout_vc8**

NVCSI_STREAM_2_ERR_INTR_TYPE_VC9_0

Offset: 0x80ec
Byte Offset: 0x203b0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc9**
4	0x0	**err_intr_type_ph_single_crc_err_vc9**
3	0x0	**err_intr_type_pd_wc_short_err_vc9**
2	0x0	**err_intr_type_pd_crc_err_vc9**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc9**
0	0x0	**err_intr_type_ppfsm_timeout_vc9**

NVCSI_STREAM_2_ERR_INTR_TYPE_VC10_0

Offset: 0x80ed
Byte Offset: 0x203b4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc10**
4	0x0	**err_intr_type_ph_single_crc_err_vc10**
3	0x0	**err_intr_type_pd_wc_short_err_vc10**
2	0x0	**err_intr_type_pd_crc_err_vc10**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc10**
0	0x0	**err_intr_type_ppfsm_timeout_vc10**

NVCSI_STREAM_2_ERR_INTR_TYPE_VC11_0

Offset: 0x80ee
Byte Offset: 0x203b8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc11**
4	0x0	**err_intr_type_ph_single_crc_err_vc11**
3	0x0	**err_intr_type_pd_wc_short_err_vc11**
2	0x0	**err_intr_type_pd_crc_err_vc11**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc11**
0	0x0	**err_intr_type_ppfsm_timeout_vc11**

NVCSI_STREAM_2_ERR_INTR_TYPE_VC12_0

Offset: 0x80ef
Byte Offset: 0x203bc
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc12**
4	0x0	**err_intr_type_ph_single_crc_err_vc12**
3	0x0	**err_intr_type_pd_wc_short_err_vc12**
2	0x0	**err_intr_type_pd_crc_err_vc12**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc12**
0	0x0	**err_intr_type_ppfsm_timeout_vc12**

NVCSI_STREAM_2_ERR_INTR_TYPE_VC13_0

Offset: 0x80f0
Byte Offset: 0x203c0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc13**
4	0x0	**err_intr_type_ph_single_crc_err_vc13**
3	0x0	**err_intr_type_pd_wc_short_err_vc13**
2	0x0	**err_intr_type_pd_crc_err_vc13**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc13**
0	0x0	**err_intr_type_ppfsm_timeout_vc13**

NVCSI_STREAM_2_ERR_INTR_TYPE_VC14_0

Offset: 0x80f1
Byte Offset: 0x203c4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc14**
4	0x0	**err_intr_type_ph_single_crc_err_vc14**
3	0x0	**err_intr_type_pd_wc_short_err_vc14**
2	0x0	**err_intr_type_pd_crc_err_vc14**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc14**
0	0x0	**err_intr_type_ppfsm_timeout_vc14**

NVCSI_STREAM_2_ERR_INTR_TYPE_VC15_0

Offset: 0x80f2
Byte Offset: 0x203c8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc15**
4	0x0	**err_intr_type_ph_single_crc_err_vc15**
3	0x0	**err_intr_type_pd_wc_short_err_vc15**
2	0x0	**err_intr_type_pd_crc_err_vc15**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc15**
0	0x0	**err_intr_type_ppfsm_timeout_vc15**

NVCSI_STREAM_2_TPG_ENABLE_0

Offset: 0x80f3
Byte Offset: 0x203cc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	DISABLE	**TPG_ENABLE**: Enable the TPG path **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_2_TPG_VC_ENABLE_0

Offset: 0x80f4
Byte Offset: 0x203d0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15	DISABLE	**VC15_EN**: Enable bit for the VC15 **0** = DISABLE **1** = ENABLE
14	DISABLE	**VC14_EN**: Enable bit for the VC14 **0** = DISABLE **1** = ENABLE
13	DISABLE	**VC13_EN**: Enable bit for the VC13 **0** = DISABLE **1** = ENABLE
12	DISABLE	**VC12_EN**: Enable bit for the VC12 **0** = DISABLE **1** = ENABLE
11	DISABLE	**VC11_EN**: Enable bit for the VC11 **0** = DISABLE **1** = ENABLE
10	DISABLE	**VC10_EN**: Enable bit for the VC10 **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
9	DISABLE	**VC9_EN**: Enable bit for the VC9 **0** = DISABLE **1** = ENABLE
8	DISABLE	**VC8_EN**: Enable bit for the VC8 **0** = DISABLE **1** = ENABLE
7	DISABLE	**VC7_EN**: Enable bit for the VC7 **0** = DISABLE **1** = ENABLE
6	DISABLE	**VC6_EN**: Enable bit for the VC6 **0** = DISABLE **1** = ENABLE
5	DISABLE	**VC5_EN**: Enable bit for the VC5 **0** = DISABLE **1** = ENABLE
4	DISABLE	**VC4_EN**: Enable bit for the VC4 **0** = DISABLE **1** = ENABLE
3	DISABLE	**VC3_EN**: Enable bit for the VC3 **0** = DISABLE **1** = ENABLE
2	DISABLE	**VC2_EN**: Enable bit for the VC2 **0** = DISABLE **1** = ENABLE
1	DISABLE	**VC1_EN**: Enable bit for the VC1 **0** = DISABLE **1** = ENABLE
0	DISABLE	**VC0_EN**: Enable bit for the VC0 **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_2_LOAD_TPG_CFG_0

Offset: 0x80f5
Byte Offset: 0x203d4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**LOAD**: Load the shadow register

NVCSI_STREAM_2_TPG_CTRL_0

Offset: 0x80f6
Byte Offset: 0x203d8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000008a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1000,1010)

Bit	Reset	Description
7:4	0x8	**DATA_SPEED**: Control the data generation speed, valid range is 1~8.
3	ENABLE	**SKIP_LS_LE_PKT**: If the LS/LE packet need to generated. **0** = DISABLE **1** = ENABLE
2	DISABLE	**OVERRIDE_CRC**: Override the packet header CRC and payload CRC. **0** = DISABLE **1** = ENABLE
1	CORE	**DEST**: The TPG pattern is send to PP or send to CIL for TX. **0** = CIL **1** = CORE

Bit	Reset	Description
0	DPHY	**PHY_MODE**: CPHY or DPHY packet structure for TPG **0** = DPHY **1** = CPHY

NVCSI_STREAM_2_TPG_VBLANK_0

Offset: 0x80f7
Byte Offset: 0x203dc
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000800 (0b0000,0000,0000,0000,0000,1000,0000,0000)

Bit	Reset	Description
31:0	0x800	**VBLANK**: The vblank value, unit is nvcsi clock cycles.

NVCSI_STREAM_2_TPG_HBLANK_0

Offset: 0x80f8
Byte Offset: 0x203e0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000800 (0b0000,0000,0000,0000,0000,1000,0000,0000)

Bit	Reset	Description
31:0	0x800	**HBLANK**: The hblank value, unit is nvcsi clock cycles.

NVCSI_STREAM_2_TPG_STATUS_0

Offset: 0x80f9
Byte Offset: 0x203e4
Read/Write: RO
Parity Protection: N
Shadow: N

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0001ffff (0bxxxx,xxxx,xxxx,xxx1,1111,1111,1111,1111)

Bit	Reset	Description
16	IDLE	**VC15_STATUS**: Indicate the VC15 status **0** = BUSY **1** = IDLE
15	IDLE	**VC14_STATUS**: Indicate the VC14 status **0** = BUSY **1** = IDLE
14	IDLE	**VC13_STATUS**: Indicate the VC13 status **0** = BUSY **1** = IDLE
13	IDLE	**VC12_STATUS**: Indicate the VC12 status **0** = BUSY **1** = IDLE
12	IDLE	**VC11_STATUS**: Indicate the VC11 status **0** = BUSY **1** = IDLE
11	IDLE	**VC10_STATUS**: Indicate the VC10 status **0** = BUSY **1** = IDLE
10	IDLE	**VC9_STATUS**: Indicate the VC9 status **0** = BUSY **1** = IDLE
9	IDLE	**VC8_STATUS**: Indicate the VC8 status **0** = BUSY **1** = IDLE
8	IDLE	**VC7_STATUS**: Indicate the VC7 status **0** = BUSY **1** = IDLE

Bit	Reset	Description
7	IDLE	**VC6_STATUS**: Indicate the VC6 status **0** = BUSY **1** = IDLE
6	IDLE	**VC5_STATUS**: Indicate the VC5 status **0** = BUSY **1** = IDLE
5	IDLE	**VC4_STATUS**: Indicate the VC4 status **0** = BUSY **1** = IDLE
4	IDLE	**VC3_STATUS**: Indicate the VC3 status **0** = BUSY **1** = IDLE
3	IDLE	**VC2_STATUS**: Indicate the VC2 status **0** = BUSY **1** = IDLE
2	IDLE	**VC1_STATUS**: Indicate the VC1 status **0** = BUSY **1** = IDLE
1	IDLE	**VC0_STATUS**: Indicate the VC0 status **0** = BUSY **1** = IDLE
0	IDLE	**STATUS**: Indicate the TPG is in idle state, all packet has been send and all VC are disabled. **0** = BUSY **1** = IDLE

NVCSI_STREAM_2_TPG_PH_ECC_0

Offset: 0x80fa
Byte Offset: 0x203e8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:24	0x0	**LINE_ECC**: This field is for the long packet header ECC.
23:18	0x0	**EOL_ECC**: This field is for the EOL short packet ECC.
17:12	0x0	**SOL_ECC**: This field is for the SOL short packet ECC.
11:6	0x0	**EOF_ECC**: This field is for the EOF short packet ECC.
5:0	0x0	**SOF_ECC**: The TPG will not generate ECC for a packet. When using the TPG, SW should set the PP to skip the ecc check. To verify the ecc logic for safety BIST, SW can write a pre-calculated ECC for the TPG, when use with this mode, the TPG should generate a grescale pattern. This field is for the SOF short packet ECC.

NVCSI_STREAM_2_TPG_PF_CRC_0

Offset: 0x80fb
Byte Offset: 0x203ec
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PF_CRC**: This field is for the long packet payload CRC override.

NVCSI_STREAM_2_TPG_PH_SOF_CRC_0

Offset: 0x80fc
Byte Offset: 0x203f0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This field is for the CPHY SOF packet second packet header CRC override.
15:0	0x0	**CRC_0**: This field is for the CPHY SOF packet first packet header CRC override.

NVCSI_STREAM_2_TPG_PH_EOF_CRC_0

Offset: 0x80fd
Byte Offset: 0x203f4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This field is for the CPHY EOF packet second packet header CRC override.
15:0	0x0	**CRC_0**: This field is for the CPHY EOF packet first packet header CRC override.

NVCSI_STREAM_2_TPG_PH_SOL_CRC_0

Offset: 0x80fe
Byte Offset: 0x203f8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This field is for the CPHY SOL packet second packet header CRC override.
15:0	0x0	**CRC_0**: This field is for the CPHY SOL packet first packet header CRC override.

NVCSI_STREAM_2_TPG_PH_EOL_CRC_0

Offset: 0x80ff
Byte Offset: 0x203fc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This field is for the CPHY EOL packet second packet header CRC override.
15:0	0x0	**CRC_0**: This field is for the CPHY EOL packet first packet header CRC override.

NVCSI_STREAM_2_TPG_PH_LONG_PKT_CRC_0

Offset: 0x8100
Byte Offset: 0x20400
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This filed is for the CPHY long packet second packet header CRC override.
15:0	0x0	**CRC_0**: This filed is for the CPHY long packet first packet header CRC override.

NVCSI_STREAM_2_TPG_PKT_DELIMETER_0

Offset: 0x8101
Byte Offset: 0x20404
Read/Write: R/W
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0011)

Bit	Reset	Description
15:0	0x3	**NUM**: This field define the cycle number between two packet. Model the LP11 period between two packet, valid range is 1 to 65535.

NVCSI_STREAM_2_VC0_TPG_GAIN_CTRL_0

Offset: 0x8102
Byte Offset: 0x20408
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_2_VC1_TPG_GAIN_CTRL_0

Offset: 0x8103
Byte Offset: 0x2040c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_2_VC2_TPG_GAIN_CTRL_0

Offset: 0x8104
Byte Offset: 0x20410
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_2_VC3_TPG_GAIN_CTRL_0

Offset: 0x8105
Byte Offset: 0x20414
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_2_VC4_TPG_GAIN_CTRL_0

Offset: 0x8106
Byte Offset: 0x20418
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_2_VC5_TPG_GAIN_CTRL_0

Offset: 0x8107
Byte Offset: 0x2041c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_2_VC6_TPG_GAIN_CTRL_0

Offset: 0x8108
Byte Offset: 0x20420
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_2_VC7_TPG_GAIN_CTRL_0

Offset: 0x8109
Byte Offset: 0x20424
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_2_VC8_TPG_GAIN_CTRL_0

Offset: 0x810a
Byte Offset: 0x20428
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_2_VC9_TPG_GAIN_CTRL_0

Offset: 0x810b
Byte Offset: 0x2042c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_2_VC10_TPG_GAIN_CTRL_0

Offset: 0x810c
Byte Offset: 0x20430
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_2_VC11_TPG_GAIN_CTRL_0

Offset: 0x810d
Byte Offset: 0x20434
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_2_VC12_TPG_GAIN_CTRL_0

Offset: 0x810e
Byte Offset: 0x20438
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_2_VC13_TPG_GAIN_CTRL_0

Offset: 0x810f
Byte Offset: 0x2043c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_2_VC14_TPG_GAIN_CTRL_0

Offset: 0x8110
Byte Offset: 0x20440
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_2_VC15_TPG_GAIN_CTRL_0

Offset: 0x8111
Byte Offset: 0x20444
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_2_SPARE_0
Offset: 0x8112
Byte Offset: 0x20448
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**stream_reg**

7.2.1.3.4 NVCSI Stream3 Registers

NVCSI_STREAM_3_SW_RESET_CTRL_0

Offset: 0xa000
Byte Offset: 0x28000
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DISABLE	**cfg_swreset**: Reset the pixel parser. **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_3_SLCG_CTRL_0

Offset: 0xa001
Byte Offset: 0x28004
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DISABLE	**cfg_slcg_override**: Enable the SLCG override for the pixel parser. **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_3_PP_PHY_CTRL_0

Offset: 0xa002
Byte Offset: 0x28008
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DPHY	**cfg_phy_mode**: **0** = DPHY **1** = CPHY

NVCSI_STREAM_3_PP_EN_CTRL_0

Offset: 0xa003
Byte Offset: 0x2800c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	CIL	**cfg_src**: The pixel source of the pixel parser **0** = CIL **1** = TPG
0	DISABLE	**cfg_pp_en**: Pixel Parser streaming enable **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_3_VC0_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xa004
Byte Offset: 0x28010
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc0_dt_nooverride_0**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC0_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xa005
Byte Offset: 0x28014
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc0_dt_nooverride_1**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC0_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xa006
Byte Offset: 0x28018
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc0_dt_nooverride_2**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC0_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xa007
Byte Offset: 0x2801c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc0_dt_nooverride_3**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC0_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xa008
Byte Offset: 0x28020
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc0_dt_nooverride_4**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC0_DT_OVERRIDE_0

Offset: 0xa009
Byte Offset: 0x28024
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc0_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc0_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc0_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_3_VC1_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xa00a
Byte Offset: 0x28028
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc1_dt_nooverride_0**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC1_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xa00b
Byte Offset: 0x2802c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc1_dt_nooverride_1**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC1_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xa00c
Byte Offset: 0x28030
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc1_dt_nooverride_2**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC1_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xa00d
Byte Offset: 0x28034
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc1_dt_nooverride_3**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC1_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xa00e
Byte Offset: 0x28038
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc1_dt_nooverride_4**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC1_DT_OVERRIDE_0

Offset: 0xa00f
Byte Offset: 0x2803c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc1_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc1_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc1_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_3_VC2_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xa010
Byte Offset: 0x28040
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc2_dt_nooverride_0**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC2_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xa011
Byte Offset: 0x28044
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc2_dt_nooverride_1**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC2_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xa012
Byte Offset: 0x28048
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc2_dt_nooverride_2**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC2_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xa013
Byte Offset: 0x2804c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc2_dt_nooverride_3**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC2_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xa014
Byte Offset: 0x28050
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc2_dt_nooverride_4**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC2_DT_OVERRIDE_0

Offset: 0xa015
Byte Offset: 0x28054
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc2_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc2_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc2_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_3_VC3_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xa016
Byte Offset: 0x28058
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc3_dt_nooverride_0**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC3_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xa017
Byte Offset: 0x2805c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc3_dt_nooverride_1**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC3_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xa018
Byte Offset: 0x28060
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc3_dt_nooverride_2**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC3_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xa019
Byte Offset: 0x28064
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc3_dt_nooverride_3**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC3_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xa01a
Byte Offset: 0x28068
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc3_dt_nooverride_4**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC3_DT_OVERRIDE_0

Offset: 0xa01b
Byte Offset: 0x2806c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc3_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc3_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc3_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_3_VC4_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xa01c
Byte Offset: 0x28070
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc4_dt_nooverride_0**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC4_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xa01d
Byte Offset: 0x28074
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc4_dt_nooverride_1**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC4_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xa01e
Byte Offset: 0x28078
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc4_dt_nooverride_2**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC4_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xa01f
Byte Offset: 0x2807c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc4_dt_nooverride_3**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC4_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xa020
Byte Offset: 0x28080
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc4_dt_nooverride_4**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC4_DT_OVERRIDE_0

Offset: 0xa021
Byte Offset: 0x28084
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc4_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc4_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc4_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_3_VC5_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xa022
Byte Offset: 0x28088
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc5_dt_nooverride_0**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC5_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xa023
Byte Offset: 0x2808c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc5_dt_nooverride_1**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC5_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xa024
Byte Offset: 0x28090
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc5_dt_nooverride_2**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC5_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xa025
Byte Offset: 0x28094
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc5_dt_nooverride_3**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC5_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xa026
Byte Offset: 0x28098
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc5_dt_nooverride_4**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC5_DT_OVERRIDE_0

Offset: 0xa027
Byte Offset: 0x2809c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc5_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc5_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc5_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_3_VC6_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xa028
Byte Offset: 0x280a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc6_dt_nooverride_0**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC6_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xa029
Byte Offset: 0x280a4
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc6_dt_nooverride_1**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC6_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xa02a
Byte Offset: 0x280a8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc6_dt_nooverride_2**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC6_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xa02b
Byte Offset: 0x280ac
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc6_dt_nooverride_3**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC6_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xa02c
Byte Offset: 0x280b0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc6_dt_nooverride_4**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC6_DT_OVERRIDE_0

Offset: 0xa02d
Byte Offset: 0x280b4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc6_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc6_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc6_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_3_VC7_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xa02e
Byte Offset: 0x280b8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc7_dt_nooverride_0**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC7_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xa02f
Byte Offset: 0x280bc
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc7_dt_nooverride_1**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC7_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xa030
Byte Offset: 0x280c0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc7_dt_nooverride_2**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC7_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xa031
Byte Offset: 0x280c4
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc7_dt_nooverride_3**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC7_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xa032
Byte Offset: 0x280c8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc7_dt_nooverride_4**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC7_DT_OVERRIDE_0

Offset: 0xa033
Byte Offset: 0x280cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc7_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc7_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc7_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_3_VC8_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xa034
Byte Offset: 0x280d0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc8_dt_nooverride_0**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC8_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xa035
Byte Offset: 0x280d4
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc8_dt_nooverride_1**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC8_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xa036
Byte Offset: 0x280d8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc8_dt_nooverride_2**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC8_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xa037
Byte Offset: 0x280dc
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc8_dt_nooverride_3**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC8_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xa038
Byte Offset: 0x280e0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc8_dt_nooverride_4**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC8_DT_OVERRIDE_0

Offset: 0xa039
Byte Offset: 0x280e4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc8_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc8_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc8_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_3_VC9_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xa03a
Byte Offset: 0x280e8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc9_dt_nooverride_0**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC9_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xa03b
Byte Offset: 0x280ec
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc9_dt_nooverride_1**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC9_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xa03c
Byte Offset: 0x280f0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc9_dt_nooverride_2**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC9_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xa03d
Byte Offset: 0x280f4
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc9_dt_nooverride_3**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC9_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xa03e
Byte Offset: 0x280f8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc9_dt_nooverride_4**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC9_DT_OVERRIDE_0

Offset: 0xa03f
Byte Offset: 0x280fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc9_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc9_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc9_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_3_VC10_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xa040
Byte Offset: 0x28100
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc10_dt_nooverride_0**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC10_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xa041
Byte Offset: 0x28104
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc10_dt_nooverride_1**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC10_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xa042
Byte Offset: 0x28108
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc10_dt_nooverride_2**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC10_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xa043
Byte Offset: 0x2810c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc10_dt_nooverride_3**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC10_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xa044
Byte Offset: 0x28110
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc10_dt_nooverride_4**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC10_DT_OVERRIDE_0

Offset: 0xa045
Byte Offset: 0x28114
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc10_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc10_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc10_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_3_VC11_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xa046
Byte Offset: 0x28118
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc11_dt_nooverride_0**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC11_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xa047
Byte Offset: 0x2811c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc11_dt_nooverride_1**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC11_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xa048
Byte Offset: 0x28120
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc11_dt_nooverride_2**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC11_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xa049
Byte Offset: 0x28124
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc11_dt_nooverride_3**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC11_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xa04a
Byte Offset: 0x28128
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc11_dt_nooverride_4**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC11_DT_OVERRIDE_0

Offset: 0xa04b
Byte Offset: 0x2812c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc11_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc11_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc11_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_3_VC12_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xa04c
Byte Offset: 0x28130
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc12_dt_nooverride_0**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC12_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xa04d
Byte Offset: 0x28134
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc12_dt_nooverride_1**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC12_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xa04e
Byte Offset: 0x28138
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc12_dt_nooverride_2**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC12_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xa04f
Byte Offset: 0x2813c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc12_dt_nooverride_3**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC12_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xa050
Byte Offset: 0x28140
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc12_dt_nooverride_4**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC12_DT_OVERRIDE_0

Offset: 0xa051
Byte Offset: 0x28144
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc12_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc12_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc12_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_3_VC13_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xa052
Byte Offset: 0x28148
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc13_dt_nooverride_0**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC13_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xa053
Byte Offset: 0x2814c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc13_dt_nooverride_1**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC13_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xa054
Byte Offset: 0x28150
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc13_dt_nooverride_2**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC13_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xa055
Byte Offset: 0x28154
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc13_dt_nooverride_3**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC13_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xa056
Byte Offset: 0x28158
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc13_dt_nooverride_4**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC13_DT_OVERRIDE_0

Offset: 0xa057
Byte Offset: 0x2815c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc13_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc13_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc13_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_3_VC14_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xa058
Byte Offset: 0x28160
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc14_dt_nooverride_0**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC14_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xa059
Byte Offset: 0x28164
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc14_dt_nooverride_1**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC14_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xa05a
Byte Offset: 0x28168
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc14_dt_nooverride_2**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC14_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xa05b
Byte Offset: 0x2816c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc14_dt_nooverride_3**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC14_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xa05c
Byte Offset: 0x28170
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc14_dt_nooverride_4**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC14_DT_OVERRIDE_0

Offset: 0xa05d
Byte Offset: 0x28174
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc14_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc14_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc14_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_3_VC15_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xa05e
Byte Offset: 0x28178
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc15_dt_nooverride_0**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC15_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xa05f
Byte Offset: 0x2817c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc15_dt_nooverride_1**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC15_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xa060
Byte Offset: 0x28180
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc15_dt_nooverride_2**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC15_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xa061
Byte Offset: 0x28184
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc15_dt_nooverride_3**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC15_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xa062
Byte Offset: 0x28188
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc15_dt_nooverride_4**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_3_VC15_DT_OVERRIDE_0

Offset: 0xa063
Byte Offset: 0x2818c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc15_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc15_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc15_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_3_PPFSM_TIMEOUT_CTRL_0

Offset: 0xa064
Byte Offset: 0x28190
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x7fffffff (0b0111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31	DISABLE	**cfg_timeout_en**: Enable Timeout counter for the PP FSM **0** = DISABLE **1** = ENABLE
30:0	0x7fffffff	**cfg_timeout_period**: Timeout period

NVCSI_STREAM_3_PH_CHK_CTRL_0

Offset: 0xa065
Byte Offset: 0x28194
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	ENABLE	**cfg_ph_16_vc**: 16 VC support **0** = DISABLE **1** = ENABLE
1	ENABLE	**cfg_ph_crc_chk_en**: PH CRC check enable (only for CPHY case), when this bit is set to 0, the packet header will still be decode when the CRC check fail, but the error will be set. **0** = DISABLE **1** = ENABLE
0	ENABLE	**cfg_ph_ecc_chk_en**: PH ECC check enable (only for DPHY case), when this bit is set to 0, the packet header will still be decode when the ECC check fail, but the error will be set. **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_3_VC0_DPCM_CTRL_0

Offset: 0xa066
Byte Offset: 0x28198
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc0_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_3_VC1_DPCM_CTRL_0

Offset: 0xa067
Byte Offset: 0x2819c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc1_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_3_VC2_DPCM_CTRL_0

Offset: 0xa068
Byte Offset: 0x281a0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc2_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_3_VC3_DPCM_CTRL_0

Offset: 0xa069
Byte Offset: 0x281a4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc3_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_3_VC4_DPCM_CTRL_0

Offset: 0xa06a
Byte Offset: 0x281a8
Read/Write: R/W
Parity Protection: Y
Shadow: N

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc4_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_3_VC5_DPCM_CTRL_0

Offset: 0xa06b
Byte Offset: 0x281ac
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc5_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_3_VC6_DPCM_CTRL_0

Offset: 0xa06c
Byte Offset: 0x281b0
Read/Write: R/W
Parity Protection: Y

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc6_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_3_VC7_DPCM_CTRL_0

Offset: 0xa06d
Byte Offset: 0x281b4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc7_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_3_VC8_DPCM_CTRL_0

Offset: 0xa06e
Byte Offset: 0x281b8
Read/Write: R/W

Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc8_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_3_VC9_DPCM_CTRL_0

Offset: 0xa06f
Byte Offset: 0x281bc
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc9_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_3_VC10_DPCM_CTRL_0

Offset: 0xa070
Byte Offset: 0x281c0

Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc10_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_3_VC11_DPCM_CTRL_0

Offset: 0xa071
Byte Offset: 0x281c4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc11_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_3_VC12_DPCM_CTRL_0

Offset: 0xa072

Byte Offset: 0x281c8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc12_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_3_VC13_DPCM_CTRL_0

Offset: 0xa073
Byte Offset: 0x281cc
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc13_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_3_VC14_DPCM_CTRL_0

Offset: 0xa074
Byte Offset: 0x281d0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc14_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_3_VC15_DPCM_CTRL_0

Offset: 0xa075
Byte Offset: 0x281d4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc15_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_3_PF_CRC_0
Status register on packet data CRC
Offset: 0xa076
Byte Offset: 0x281d8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**calc_crc**: 16 bit CRC computed over current packet (has to match with the CRC in PF for a good packet)
15:0	0x0	**rx_crc**: 16 bit CRC from PF

NVCSI_STREAM_3_PH_WC_0
Status register on WC
Offset: 0xa077
Byte Offset: 0x281dc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**calc_wc**: 16 bit WC computed over current packet (has to match with the WC in PH for a good packet)
15:0	0x0	**rx_wc**: 16 bit WC from PH

NVCSI_STREAM_3_PH_DI_0
Status register on Data ID
Offset: 0xa078
Byte Offset: 0x281e0
Read/Write: RO
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9:6	0x0	**rx_vc**: 4 bit VC from PH
5:0	0x0	**rx_dt**: 6 bit DTYPE from PH

NVCSI_STREAM_3_ERROR_STATUS2VI_MASK_0

Offset: 0xa079
Byte Offset: 0x281e4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15	0x0	**cfg_err_status2vi_mask_vc15**: for VC15 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
14	0x0	**cfg_err_status2vi_mask_vc14**: for VC14 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
13	0x0	**cfg_err_status2vi_mask_vc13**: for VC13 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
12	0x0	**cfg_err_status2vi_mask_vc12**: for VC12 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
11	0x0	**cfg_err_status2vi_mask_vc11**: for VC11 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
10	0x0	**cfg_err_status2vi_mask_vc10**: for VC10 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
9	0x0	**cfg_err_status2vi_mask_vc9**: for VC9 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF

Bit	Reset	Description
8	0x0	**cfg_err_status2vi_mask_vc8**: for VC8 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
7	0x0	**cfg_err_status2vi_mask_vc7**: for VC7 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
6	0x0	**cfg_err_status2vi_mask_vc6**: for VC6 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
5	0x0	**cfg_err_status2vi_mask_vc5**: for VC5 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
4	0x0	**cfg_err_status2vi_mask_vc4**: for VC4 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
3	0x0	**cfg_err_status2vi_mask_vc3**: for VC3 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
2	0x0	**cfg_err_status2vi_mask_vc2**: for VC2 : 0 = err_status2vi_vc2 will be send to VI at EOF; 1 = No error will be send to VI at EOF
1	0x0	**cfg_err_status2vi_mask_vc1**: for VC1 : 0 = err_status2vi_vc1 will be send to VI at EOF; 1 = No error will be send to VI at EOF
0	0x0	**cfg_err_status2vi_mask_vc0**: for VC0 : 0 = err_status2vi_vc0 will be send to VI at EOF; 1 = No error will be send to VI at EOF

NVCSI_STREAM_3_ERROR_STATUS2VI_VC0_0

This is RO register for SW
Offset: 0xa07a
Byte Offset: 0x281e8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc0**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_3_ERROR_STATUS2VI_VC1_0

This is RO register for SW
Offset: 0xa07b
Byte Offset: 0x281ec
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc1**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_3_ERROR_STATUS2VI_VC2_0

This is RO register for SW
Offset: 0xa07c
Byte Offset: 0x281f0
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc2**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_3_ERROR_STATUS2VI_VC3_0

This is RO register for SW
Offset: 0xa07d
Byte Offset: 0x281f4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc3**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_3_ERROR_STATUS2VI_VC4_0

This is RO register for SW
Offset: 0xa07e
Byte Offset: 0x281f8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc4**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_3_ERROR_STATUS2VI_VC5_0

This is RO register for SW
Offset: 0xa07f
Byte Offset: 0x281fc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc5**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_3_ERROR_STATUS2VI_VC6_0

This is RO register for SW
Offset: 0xa080
Byte Offset: 0x28200
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc6**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_3_ERROR_STATUS2VI_VC7_0

This is RO register for SW
Offset: 0xa081
Byte Offset: 0x28204
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc7**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_3_ERROR_STATUS2VI_VC8_0

This is RO register for SW
Offset: 0xa082
Byte Offset: 0x28208
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc8**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_3_ERROR_STATUS2VI_VC9_0

This is RO register for SW
Offset: 0xa083
Byte Offset: 0x2820c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc9**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_3_ERROR_STATUS2VI_VC10_0

This is RO register for SW
Offset: 0xa084
Byte Offset: 0x28210
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc10**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_3_ERROR_STATUS2VI_VC11_0

This is RO register for SW
Offset: 0xa085
Byte Offset: 0x28214
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc11**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_3_ERROR_STATUS2VI_VC12_0

This is RO register for SW
Offset: 0xa086
Byte Offset: 0x28218
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc12**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_3_ERROR_STATUS2VI_VC13_0

This is RO register for SW
Offset: 0xa087
Byte Offset: 0x2821c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc13**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_3_ERROR_STATUS2VI_VC14_0

This is RO register for SW
Offset: 0xa088
Byte Offset: 0x28220
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc14**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_3_ERROR_STATUS2VI_VC15_0

This is RO register for SW
Offset: 0xa089
Byte Offset: 0x28224
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc15**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_3_INTR_STATUS_0

Offset: 0xa08a
Byte Offset: 0x28228
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**intr_stat_novc**: Not VC related interrupt
15	0x0	**intr_stat_vc15**: VC15 event

Bit	Reset	Description
14	0x0	**intr_stat_vc14**: VC14 event
13	0x0	**intr_stat_vc13**: VC13 event
12	0x0	**intr_stat_vc12**: VC12 event
11	0x0	**intr_stat_vc11**: VC11 event
10	0x0	**intr_stat_vc10**: VC10 event
9	0x0	**intr_stat_vc9**: VC9 event
8	0x0	**intr_stat_vc8**: VC8 event
7	0x0	**intr_stat_vc7**: VC7 event
6	0x0	**intr_stat_vc6**: VC6 event
5	0x0	**intr_stat_vc5**: VC5 event
4	0x0	**intr_stat_vc4**: VC4 event
3	0x0	**intr_stat_vc3**: VC3 event
2	0x0	**intr_stat_vc2**: VC2 event
1	0x0	**intr_stat_vc1**: VC1 event
0	0x0	**intr_stat_vc0**: VC0 event

NVCSI_STREAM_3_INTR_STATUS_NOVC_0

Offset: 0xa08b
Byte Offset: 0x2822c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**intr_stat_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**intr_stat_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_3_INTR_STATUS_VC0_0

Offset: 0xa08c
Byte Offset: 0x28230
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc0**
4	0x0	**intr_stat_ph_single_crc_err_vc0**
3	0x0	**intr_stat_pd_wc_short_err_vc0**
2	0x0	**intr_stat_pd_crc_err_vc0**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc0**
0	0x0	**intr_stat_ppfsm_timeout_vc0**

NVCSI_STREAM_3_INTR_STATUS_VC1_0

Offset: 0xa08d
Byte Offset: 0x28234
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc1**

Bit	Reset	Description
4	0x0	**intr_stat_ph_single_crc_err_vc1**
3	0x0	**intr_stat_pd_wc_short_err_vc1**
2	0x0	**intr_stat_pd_crc_err_vc1**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc1**
0	0x0	**intr_stat_ppfsm_timeout_vc1**

NVCSI_STREAM_3_INTR_STATUS_VC2_0

Offset: 0xa08e
Byte Offset: 0x28238
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc2**
4	0x0	**intr_stat_ph_single_crc_err_vc2**
3	0x0	**intr_stat_pd_wc_short_err_vc2**
2	0x0	**intr_stat_pd_crc_err_vc2**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc2**
0	0x0	**intr_stat_ppfsm_timeout_vc2**

NVCSI_STREAM_3_INTR_STATUS_VC3_0

Offset: 0xa08f
Byte Offset: 0x2823c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc3**
4	0x0	**intr_stat_ph_single_crc_err_vc3**
3	0x0	**intr_stat_pd_wc_short_err_vc3**
2	0x0	**intr_stat_pd_crc_err_vc3**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc3**
0	0x0	**intr_stat_ppfsm_timeout_vc3**

NVCSI_STREAM_3_INTR_STATUS_VC4_0

Offset: 0xa090
Byte Offset: 0x28240
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc4**
4	0x0	**intr_stat_ph_single_crc_err_vc4**
3	0x0	**intr_stat_pd_wc_short_err_vc4**
2	0x0	**intr_stat_pd_crc_err_vc4**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc4**
0	0x0	**intr_stat_ppfsm_timeout_vc4**

NVCSI_STREAM_3_INTR_STATUS_VC5_0

Offset: 0xa091
Byte Offset: 0x28244
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc5**
4	0x0	**intr_stat_ph_single_crc_err_vc5**
3	0x0	**intr_stat_pd_wc_short_err_vc5**
2	0x0	**intr_stat_pd_crc_err_vc5**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc5**
0	0x0	**intr_stat_ppfsm_timeout_vc5**

NVCSI_STREAM_3_INTR_STATUS_VC6_0

Offset: 0xa092
Byte Offset: 0x28248
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc6**
4	0x0	**intr_stat_ph_single_crc_err_vc6**
3	0x0	**intr_stat_pd_wc_short_err_vc6**
2	0x0	**intr_stat_pd_crc_err_vc6**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc6**
0	0x0	**intr_stat_ppfsm_timeout_vc6**

NVCSI_STREAM_3_INTR_STATUS_VC7_0

Offset: 0xa093
Byte Offset: 0x2824c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc7**
4	0x0	**intr_stat_ph_single_crc_err_vc7**
3	0x0	**intr_stat_pd_wc_short_err_vc7**
2	0x0	**intr_stat_pd_crc_err_vc7**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc7**
0	0x0	**intr_stat_ppfsm_timeout_vc7**

NVCSI_STREAM_3_INTR_STATUS_VC8_0

Offset: 0xa094
Byte Offset: 0x28250
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc8**
4	0x0	**intr_stat_ph_single_crc_err_vc8**
3	0x0	**intr_stat_pd_wc_short_err_vc8**
2	0x0	**intr_stat_pd_crc_err_vc8**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc8**
0	0x0	**intr_stat_ppfsm_timeout_vc8**

NVCSI_STREAM_3_INTR_STATUS_VC9_0

Offset: 0xa095
Byte Offset: 0x28254
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc9**
4	0x0	**intr_stat_ph_single_crc_err_vc9**
3	0x0	**intr_stat_pd_wc_short_err_vc9**
2	0x0	**intr_stat_pd_crc_err_vc9**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc9**
0	0x0	**intr_stat_ppfsm_timeout_vc9**

NVCSI_STREAM_3_INTR_STATUS_VC10_0

Offset: 0xa096
Byte Offset: 0x28258
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc10**
4	0x0	**intr_stat_ph_single_crc_err_vc10**
3	0x0	**intr_stat_pd_wc_short_err_vc10**
2	0x0	**intr_stat_pd_crc_err_vc10**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc10**
0	0x0	**intr_stat_ppfsm_timeout_vc10**

NVCSI_STREAM_3_INTR_STATUS_VC11_0

Offset: 0xa097
Byte Offset: 0x2825c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc11**
4	0x0	**intr_stat_ph_single_crc_err_vc11**
3	0x0	**intr_stat_pd_wc_short_err_vc11**
2	0x0	**intr_stat_pd_crc_err_vc11**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc11**
0	0x0	**intr_stat_ppfsm_timeout_vc11**

NVCSI_STREAM_3_INTR_STATUS_VC12_0

Offset: 0xa098
Byte Offset: 0x28260
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc12**
4	0x0	**intr_stat_ph_single_crc_err_vc12**
3	0x0	**intr_stat_pd_wc_short_err_vc12**
2	0x0	**intr_stat_pd_crc_err_vc12**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc12**
0	0x0	**intr_stat_ppfsm_timeout_vc12**

NVCSI_STREAM_3_INTR_STATUS_VC13_0

Offset: 0xa099
Byte Offset: 0x28264
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc13**
4	0x0	**intr_stat_ph_single_crc_err_vc13**
3	0x0	**intr_stat_pd_wc_short_err_vc13**
2	0x0	**intr_stat_pd_crc_err_vc13**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc13**
0	0x0	**intr_stat_ppfsm_timeout_vc13**

NVCSI_STREAM_3_INTR_STATUS_VC14_0

Offset: 0xa09a
Byte Offset: 0x28268
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc14**
4	0x0	**intr_stat_ph_single_crc_err_vc14**
3	0x0	**intr_stat_pd_wc_short_err_vc14**
2	0x0	**intr_stat_pd_crc_err_vc14**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc14**
0	0x0	**intr_stat_ppfsm_timeout_vc14**

NVCSI_STREAM_3_INTR_STATUS_VC15_0

Offset: 0xa09b
Byte Offset: 0x2826c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc15**
4	0x0	**intr_stat_ph_single_crc_err_vc15**
3	0x0	**intr_stat_pd_wc_short_err_vc15**
2	0x0	**intr_stat_pd_crc_err_vc15**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc15**
0	0x0	**intr_stat_ppfsm_timeout_vc15**

NVCSI_STREAM_3_INTR_MASK_NOVC_0

Offset: 0xa09c
Byte Offset: 0x28270
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**intr_mask_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**intr_mask_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_3_INTR_MASK_VC0_0

Offset: 0xa09d
Byte Offset: 0x28274
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc0**
4	0x0	**intr_mask_ph_single_crc_err_vc0**

Bit	Reset	Description
3	0x0	**intr_mask_pd_wc_short_err_vc0**
2	0x0	**intr_mask_pd_crc_err_vc0**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc0**
0	0x0	**intr_mask_ppfsm_timeout_vc0**

NVCSI_STREAM_3_INTR_MASK_VC1_0

Offset: 0xa09e
Byte Offset: 0x28278
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc1**
4	0x0	**intr_mask_ph_single_crc_err_vc1**
3	0x0	**intr_mask_pd_wc_short_err_vc1**
2	0x0	**intr_mask_pd_crc_err_vc1**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc1**
0	0x0	**intr_mask_ppfsm_timeout_vc1**

NVCSI_STREAM_3_INTR_MASK_VC2_0

Offset: 0xa09f
Byte Offset: 0x2827c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc2**

Bit	Reset	Description
4	0x0	**intr_mask_ph_single_crc_err_vc2**
3	0x0	**intr_mask_pd_wc_short_err_vc2**
2	0x0	**intr_mask_pd_crc_err_vc2**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc2**
0	0x0	**intr_mask_ppfsm_timeout_vc2**

NVCSI_STREAM_3_INTR_MASK_VC3_0

Offset: 0xa0a0
Byte Offset: 0x28280
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc3**
4	0x0	**intr_mask_ph_single_crc_err_vc3**
3	0x0	**intr_mask_pd_wc_short_err_vc3**
2	0x0	**intr_mask_pd_crc_err_vc3**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc3**
0	0x0	**intr_mask_ppfsm_timeout_vc3**

NVCSI_STREAM_3_INTR_MASK_VC4_0

Offset: 0xa0a1
Byte Offset: 0x28284
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc4**
4	0x0	**intr_mask_ph_single_crc_err_vc4**
3	0x0	**intr_mask_pd_wc_short_err_vc4**
2	0x0	**intr_mask_pd_crc_err_vc4**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc4**
0	0x0	**intr_mask_ppfsm_timeout_vc4**

NVCSI_STREAM_3_INTR_MASK_VC5_0

Offset: 0xa0a2
Byte Offset: 0x28288
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc5**
4	0x0	**intr_mask_ph_single_crc_err_vc5**
3	0x0	**intr_mask_pd_wc_short_err_vc5**
2	0x0	**intr_mask_pd_crc_err_vc5**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc5**
0	0x0	**intr_mask_ppfsm_timeout_vc5**

NVCSI_STREAM_3_INTR_MASK_VC6_0

Offset: 0xa0a3
Byte Offset: 0x2828c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc6**
4	0x0	**intr_mask_ph_single_crc_err_vc6**
3	0x0	**intr_mask_pd_wc_short_err_vc6**
2	0x0	**intr_mask_pd_crc_err_vc6**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc6**
0	0x0	**intr_mask_ppfsm_timeout_vc6**

NVCSI_STREAM_3_INTR_MASK_VC7_0

Offset: 0xa0a4
Byte Offset: 0x28290
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc7**
4	0x0	**intr_mask_ph_single_crc_err_vc7**
3	0x0	**intr_mask_pd_wc_short_err_vc7**
2	0x0	**intr_mask_pd_crc_err_vc7**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc7**
0	0x0	**intr_mask_ppfsm_timeout_vc7**

NVCSI_STREAM_3_INTR_MASK_VC8_0

Offset: 0xa0a5
Byte Offset: 0x28294
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc8**
4	0x0	**intr_mask_ph_single_crc_err_vc8**
3	0x0	**intr_mask_pd_wc_short_err_vc8**
2	0x0	**intr_mask_pd_crc_err_vc8**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc8**
0	0x0	**intr_mask_ppfsm_timeout_vc8**

NVCSI_STREAM_3_INTR_MASK_VC9_0

Offset: 0xa0a6
Byte Offset: 0x28298
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc9**
4	0x0	**intr_mask_ph_single_crc_err_vc9**
3	0x0	**intr_mask_pd_wc_short_err_vc9**
2	0x0	**intr_mask_pd_crc_err_vc9**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc9**
0	0x0	**intr_mask_ppfsm_timeout_vc9**

NVCSI_STREAM_3_INTR_MASK_VC10_0

Offset: 0xa0a7
Byte Offset: 0x2829c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc10**
4	0x0	**intr_mask_ph_single_crc_err_vc10**
3	0x0	**intr_mask_pd_wc_short_err_vc10**
2	0x0	**intr_mask_pd_crc_err_vc10**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc10**
0	0x0	**intr_mask_ppfsm_timeout_vc10**

NVCSI_STREAM_3_INTR_MASK_VC11_0

Offset: 0xa0a8
Byte Offset: 0x282a0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc11**
4	0x0	**intr_mask_ph_single_crc_err_vc11**
3	0x0	**intr_mask_pd_wc_short_err_vc11**
2	0x0	**intr_mask_pd_crc_err_vc11**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc11**
0	0x0	**intr_mask_ppfsm_timeout_vc11**

NVCSI_STREAM_3_INTR_MASK_VC12_0

Offset: 0xa0a9
Byte Offset: 0x282a4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc12**
4	0x0	**intr_mask_ph_single_crc_err_vc12**
3	0x0	**intr_mask_pd_wc_short_err_vc12**
2	0x0	**intr_mask_pd_crc_err_vc12**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc12**
0	0x0	**intr_mask_ppfsm_timeout_vc12**

NVCSI_STREAM_3_INTR_MASK_VC13_0

Offset: 0xa0aa
Byte Offset: 0x282a8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc13**
4	0x0	**intr_mask_ph_single_crc_err_vc13**
3	0x0	**intr_mask_pd_wc_short_err_vc13**
2	0x0	**intr_mask_pd_crc_err_vc13**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc13**
0	0x0	**intr_mask_ppfsm_timeout_vc13**

NVCSI_STREAM_3_INTR_MASK_VC14_0

Offset: 0xa0ab
Byte Offset: 0x282ac
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc14**
4	0x0	**intr_mask_ph_single_crc_err_vc14**
3	0x0	**intr_mask_pd_wc_short_err_vc14**
2	0x0	**intr_mask_pd_crc_err_vc14**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc14**
0	0x0	**intr_mask_ppfsm_timeout_vc14**

NVCSI_STREAM_3_INTR_MASK_VC15_0

Offset: 0xa0ac
Byte Offset: 0x282b0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc15**
4	0x0	**intr_mask_ph_single_crc_err_vc15**
3	0x0	**intr_mask_pd_wc_short_err_vc15**
2	0x0	**intr_mask_pd_crc_err_vc15**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc15**
0	0x0	**intr_mask_ppfsm_timeout_vc15**

NVCSI_STREAM_3_CORRECTABLE_ERR_INTR_STATUS_0

Offset: 0xa0ad
Byte Offset: 0x282b4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**err_intr_stat_novc**: Not VC related interrupt
15	0x0	**err_intr_stat_vc15**: VC15 event
14	0x0	**err_intr_stat_vc14**: VC14 event
13	0x0	**err_intr_stat_vc13**: VC13 event
12	0x0	**err_intr_stat_vc12**: VC12 event
11	0x0	**err_intr_stat_vc11**: VC11 event
10	0x0	**err_intr_stat_vc10**: VC10 event
9	0x0	**err_intr_stat_vc9**: VC9 event
8	0x0	**err_intr_stat_vc8**: VC8 event
7	0x0	**err_intr_stat_vc7**: VC7 event
6	0x0	**err_intr_stat_vc6**: VC6 event
5	0x0	**err_intr_stat_vc5**: VC5 event
4	0x0	**err_intr_stat_vc4**: VC4 event
3	0x0	**err_intr_stat_vc3**: VC3 event
2	0x0	**err_intr_stat_vc2**: VC2 event
1	0x0	**err_intr_stat_vc1**: VC1 event
0	0x0	**err_intr_stat_vc0**: VC0 event

NVCSI_STREAM_3_CORRECTABLE_ERR_INTR_STATUS_NOVC_0

Offset: 0xa0ae
Byte Offset: 0x282b8
Read/Write: R/W

Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**err_intr_stat_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**err_intr_stat_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_3_CORRECTABLE_ERR_INTR_STATUS_VC0_0

Offset: 0xa0af
Byte Offset: 0x282bc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc0**
4	0x0	**err_intr_stat_ph_single_crc_err_vc0**
3	0x0	**err_intr_stat_pd_wc_short_err_vc0**
2	0x0	**err_intr_stat_pd_crc_err_vc0**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc0**
0	0x0	**err_intr_stat_ppfsm_timeout_vc0**

NVCSI_STREAM_3_CORRECTABLE_ERR_INTR_STATUS_VC1_0

Offset: 0xa0b0
Byte Offset: 0x282c0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc1**
4	0x0	**err_intr_stat_ph_single_crc_err_vc1**
3	0x0	**err_intr_stat_pd_wc_short_err_vc1**
2	0x0	**err_intr_stat_pd_crc_err_vc1**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc1**
0	0x0	**err_intr_stat_ppfsm_timeout_vc1**

NVCSI_STREAM_3_CORRECTABLE_ERR_INTR_STATUS_VC2_0

Offset: 0xa0b1
Byte Offset: 0x282c4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc2**
4	0x0	**err_intr_stat_ph_single_crc_err_vc2**
3	0x0	**err_intr_stat_pd_wc_short_err_vc2**
2	0x0	**err_intr_stat_pd_crc_err_vc2**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc2**
0	0x0	**err_intr_stat_ppfsm_timeout_vc2**

NVCSI_STREAM_3_CORRECTABLE_ERR_INTR_STATUS_VC3_0

Offset: 0xa0b2
Byte Offset: 0x282c8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc3**
4	0x0	**err_intr_stat_ph_single_crc_err_vc3**
3	0x0	**err_intr_stat_pd_wc_short_err_vc3**
2	0x0	**err_intr_stat_pd_crc_err_vc3**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc3**
0	0x0	**err_intr_stat_ppfsm_timeout_vc3**

NVCSI_STREAM_3_CORRECTABLE_ERR_INTR_STATUS_VC4_0

Offset: 0xa0b3
Byte Offset: 0x282cc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc4**
4	0x0	**err_intr_stat_ph_single_crc_err_vc4**
3	0x0	**err_intr_stat_pd_wc_short_err_vc4**
2	0x0	**err_intr_stat_pd_crc_err_vc4**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc4**
0	0x0	**err_intr_stat_ppfsm_timeout_vc4**

NVCSI_STREAM_3_CORRECTABLE_ERR_INTR_STATUS_VC5_0

Offset: 0xa0b4
Byte Offset: 0x282d0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc5**
4	0x0	**err_intr_stat_ph_single_crc_err_vc5**
3	0x0	**err_intr_stat_pd_wc_short_err_vc5**
2	0x0	**err_intr_stat_pd_crc_err_vc5**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc5**
0	0x0	**err_intr_stat_ppfsm_timeout_vc5**

NVCSI_STREAM_3_CORRECTABLE_ERR_INTR_STATUS_VC6_0

Offset: 0xa0b5
Byte Offset: 0x282d4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc6**
4	0x0	**err_intr_stat_ph_single_crc_err_vc6**
3	0x0	**err_intr_stat_pd_wc_short_err_vc6**
2	0x0	**err_intr_stat_pd_crc_err_vc6**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc6**
0	0x0	**err_intr_stat_ppfsm_timeout_vc6**

NVCSI_STREAM_3_CORRECTABLE_ERR_INTR_STATUS_VC7_0

Offset: 0xa0b6
Byte Offset: 0x282d8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc7**
4	0x0	**err_intr_stat_ph_single_crc_err_vc7**
3	0x0	**err_intr_stat_pd_wc_short_err_vc7**
2	0x0	**err_intr_stat_pd_crc_err_vc7**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc7**
0	0x0	**err_intr_stat_ppfsm_timeout_vc7**

NVCSI_STREAM_3_CORRECTABLE_ERR_INTR_STATUS_VC8_0

Offset: 0xa0b7
Byte Offset: 0x282dc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc8**
4	0x0	**err_intr_stat_ph_single_crc_err_vc8**
3	0x0	**err_intr_stat_pd_wc_short_err_vc8**
2	0x0	**err_intr_stat_pd_crc_err_vc8**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc8**
0	0x0	**err_intr_stat_ppfsm_timeout_vc8**

NVCSI_STREAM_3_CORRECTABLE_ERR_INTR_STATUS_VC9_0

Offset: 0xa0b8
Byte Offset: 0x282e0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc9**
4	0x0	**err_intr_stat_ph_single_crc_err_vc9**
3	0x0	**err_intr_stat_pd_wc_short_err_vc9**
2	0x0	**err_intr_stat_pd_crc_err_vc9**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc9**
0	0x0	**err_intr_stat_ppfsm_timeout_vc9**

NVCSI_STREAM_3_CORRECTABLE_ERR_INTR_STATUS_VC10_0

Offset: 0xa0b9
Byte Offset: 0x282e4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc10**
4	0x0	**err_intr_stat_ph_single_crc_err_vc10**
3	0x0	**err_intr_stat_pd_wc_short_err_vc10**
2	0x0	**err_intr_stat_pd_crc_err_vc10**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc10**
0	0x0	**err_intr_stat_ppfsm_timeout_vc10**

NVCSI_STREAM_3_CORRECTABLE_ERR_INTR_STATUS_VC11_0

Offset: 0xa0ba
Byte Offset: 0x282e8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc11**
4	0x0	**err_intr_stat_ph_single_crc_err_vc11**
3	0x0	**err_intr_stat_pd_wc_short_err_vc11**
2	0x0	**err_intr_stat_pd_crc_err_vc11**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc11**
0	0x0	**err_intr_stat_ppfsm_timeout_vc11**

NVCSI_STREAM_3_CORRECTABLE_ERR_INTR_STATUS_VC12_0

Offset: 0xa0bb
Byte Offset: 0x282ec
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc12**
4	0x0	**err_intr_stat_ph_single_crc_err_vc12**
3	0x0	**err_intr_stat_pd_wc_short_err_vc12**
2	0x0	**err_intr_stat_pd_crc_err_vc12**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc12**
0	0x0	**err_intr_stat_ppfsm_timeout_vc12**

NVCSI_STREAM_3_CORRECTABLE_ERR_INTR_STATUS_VC13_0

Offset: 0xa0bc
Byte Offset: 0x282f0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc13**
4	0x0	**err_intr_stat_ph_single_crc_err_vc13**
3	0x0	**err_intr_stat_pd_wc_short_err_vc13**
2	0x0	**err_intr_stat_pd_crc_err_vc13**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc13**
0	0x0	**err_intr_stat_ppfsm_timeout_vc13**

NVCSI_STREAM_3_CORRECTABLE_ERR_INTR_STATUS_VC14_0

Offset: 0xa0bd
Byte Offset: 0x282f4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc14**
4	0x0	**err_intr_stat_ph_single_crc_err_vc14**
3	0x0	**err_intr_stat_pd_wc_short_err_vc14**
2	0x0	**err_intr_stat_pd_crc_err_vc14**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc14**
0	0x0	**err_intr_stat_ppfsm_timeout_vc14**

NVCSI_STREAM_3_CORRECTABLE_ERR_INTR_STATUS_VC15_0

Offset: 0xa0be
Byte Offset: 0x282f8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc15**
4	0x0	**err_intr_stat_ph_single_crc_err_vc15**
3	0x0	**err_intr_stat_pd_wc_short_err_vc15**
2	0x0	**err_intr_stat_pd_crc_err_vc15**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc15**
0	0x0	**err_intr_stat_ppfsm_timeout_vc15**

NVCSI_STREAM_3_UNCORRECTABLE_ERR_INTR_STATUS_0

Offset: 0xa0bf
Byte Offset: 0x282fc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**err_intr_stat_novc**: Not VC related interrupt
15	0x0	**err_intr_stat_vc15**: VC15 event
14	0x0	**err_intr_stat_vc14**: VC14 event
13	0x0	**err_intr_stat_vc13**: VC13 event
12	0x0	**err_intr_stat_vc12**: VC12 event
11	0x0	**err_intr_stat_vc11**: VC11 event
10	0x0	**err_intr_stat_vc10**: VC10 event
9	0x0	**err_intr_stat_vc9**: VC9 event
8	0x0	**err_intr_stat_vc8**: VC8 event
7	0x0	**err_intr_stat_vc7**: VC7 event

Bit	Reset	Description
6	0x0	**err_intr_stat_vc6**: VC6 event
5	0x0	**err_intr_stat_vc5**: VC5 event
4	0x0	**err_intr_stat_vc4**: VC4 event
3	0x0	**err_intr_stat_vc3**: VC3 event
2	0x0	**err_intr_stat_vc2**: VC2 event
1	0x0	**err_intr_stat_vc1**: VC1 event
0	0x0	**err_intr_stat_vc0**: VC0 event

NVCSI_STREAM_3_UNCORRECTABLE_ERR_INTR_STATUS_NOVC_0

Offset: 0xa0c0
Byte Offset: 0x28300
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**err_intr_stat_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**err_intr_stat_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_3_UNCORRECTABLE_ERR_INTR_STATUS_VC0_0

Offset: 0xa0c1
Byte Offset: 0x28304
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc0**
4	0x0	**err_intr_stat_ph_single_crc_err_vc0**
3	0x0	**err_intr_stat_pd_wc_short_err_vc0**
2	0x0	**err_intr_stat_pd_crc_err_vc0**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc0**
0	0x0	**err_intr_stat_ppfsm_timeout_vc0**

NVCSI_STREAM_3_UNCORRECTABLE_ERR_INTR_STATUS_VC1_0

Offset: 0xa0c2
Byte Offset: 0x28308
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc1**
4	0x0	**err_intr_stat_ph_single_crc_err_vc1**
3	0x0	**err_intr_stat_pd_wc_short_err_vc1**
2	0x0	**err_intr_stat_pd_crc_err_vc1**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc1**
0	0x0	**err_intr_stat_ppfsm_timeout_vc1**

NVCSI_STREAM_3_UNCORRECTABLE_ERR_INTR_STATUS_VC2_0

Offset: 0xa0c3
Byte Offset: 0x2830c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc2**
4	0x0	**err_intr_stat_ph_single_crc_err_vc2**
3	0x0	**err_intr_stat_pd_wc_short_err_vc2**
2	0x0	**err_intr_stat_pd_crc_err_vc2**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc2**
0	0x0	**err_intr_stat_ppfsm_timeout_vc2**

NVCSI_STREAM_3_UNCORRECTABLE_ERR_INTR_STATUS_VC3_0

Offset: 0xa0c4
Byte Offset: 0x28310
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc3**
4	0x0	**err_intr_stat_ph_single_crc_err_vc3**
3	0x0	**err_intr_stat_pd_wc_short_err_vc3**
2	0x0	**err_intr_stat_pd_crc_err_vc3**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc3**
0	0x0	**err_intr_stat_ppfsm_timeout_vc3**

NVCSI_STREAM_3_UNCORRECTABLE_ERR_INTR_STATUS_VC4_0

Offset: 0xa0c5
Byte Offset: 0x28314
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc4**
4	0x0	**err_intr_stat_ph_single_crc_err_vc4**
3	0x0	**err_intr_stat_pd_wc_short_err_vc4**
2	0x0	**err_intr_stat_pd_crc_err_vc4**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc4**
0	0x0	**err_intr_stat_ppfsm_timeout_vc4**

NVCSI_STREAM_3_UNCORRECTABLE_ERR_INTR_STATUS_VC5_0

Offset: 0xa0c6
Byte Offset: 0x28318
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc5**
4	0x0	**err_intr_stat_ph_single_crc_err_vc5**
3	0x0	**err_intr_stat_pd_wc_short_err_vc5**
2	0x0	**err_intr_stat_pd_crc_err_vc5**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc5**
0	0x0	**err_intr_stat_ppfsm_timeout_vc5**

NVCSI_STREAM_3_UNCORRECTABLE_ERR_INTR_STATUS_VC6_0

Offset: 0xa0c7
Byte Offset: 0x2831c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc6**
4	0x0	**err_intr_stat_ph_single_crc_err_vc6**
3	0x0	**err_intr_stat_pd_wc_short_err_vc6**
2	0x0	**err_intr_stat_pd_crc_err_vc6**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc6**
0	0x0	**err_intr_stat_ppfsm_timeout_vc6**

NVCSI_STREAM_3_UNCORRECTABLE_ERR_INTR_STATUS_VC7_0

Offset: 0xa0c8
Byte Offset: 0x28320
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc7**
4	0x0	**err_intr_stat_ph_single_crc_err_vc7**
3	0x0	**err_intr_stat_pd_wc_short_err_vc7**
2	0x0	**err_intr_stat_pd_crc_err_vc7**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc7**
0	0x0	**err_intr_stat_ppfsm_timeout_vc7**

NVCSI_STREAM_3_UNCORRECTABLE_ERR_INTR_STATUS_VC8_0

Offset: 0xa0c9
Byte Offset: 0x28324
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc8**
4	0x0	**err_intr_stat_ph_single_crc_err_vc8**
3	0x0	**err_intr_stat_pd_wc_short_err_vc8**
2	0x0	**err_intr_stat_pd_crc_err_vc8**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc8**
0	0x0	**err_intr_stat_ppfsm_timeout_vc8**

NVCSI_STREAM_3_UNCORRECTABLE_ERR_INTR_STATUS_VC9_0

Offset: 0xa0ca
Byte Offset: 0x28328
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc9**
4	0x0	**err_intr_stat_ph_single_crc_err_vc9**
3	0x0	**err_intr_stat_pd_wc_short_err_vc9**
2	0x0	**err_intr_stat_pd_crc_err_vc9**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc9**
0	0x0	**err_intr_stat_ppfsm_timeout_vc9**

NVCSI_STREAM_3_UNCORRECTABLE_ERR_INTR_STATUS_VC10_0

Offset: 0xa0cb
Byte Offset: 0x2832c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc10**
4	0x0	**err_intr_stat_ph_single_crc_err_vc10**
3	0x0	**err_intr_stat_pd_wc_short_err_vc10**
2	0x0	**err_intr_stat_pd_crc_err_vc10**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc10**
0	0x0	**err_intr_stat_ppfsm_timeout_vc10**

NVCSI_STREAM_3_UNCORRECTABLE_ERR_INTR_STATUS_VC11_0

Offset: 0xa0cc
Byte Offset: 0x28330
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc11**
4	0x0	**err_intr_stat_ph_single_crc_err_vc11**
3	0x0	**err_intr_stat_pd_wc_short_err_vc11**
2	0x0	**err_intr_stat_pd_crc_err_vc11**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc11**
0	0x0	**err_intr_stat_ppfsm_timeout_vc11**

NVCSI_STREAM_3_UNCORRECTABLE_ERR_INTR_STATUS_VC12_0

Offset: 0xa0cd
Byte Offset: 0x28334
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc12**
4	0x0	**err_intr_stat_ph_single_crc_err_vc12**
3	0x0	**err_intr_stat_pd_wc_short_err_vc12**
2	0x0	**err_intr_stat_pd_crc_err_vc12**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc12**
0	0x0	**err_intr_stat_ppfsm_timeout_vc12**

NVCSI_STREAM_3_UNCORRECTABLE_ERR_INTR_STATUS_VC13_0

Offset: 0xa0ce
Byte Offset: 0x28338
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc13**
4	0x0	**err_intr_stat_ph_single_crc_err_vc13**
3	0x0	**err_intr_stat_pd_wc_short_err_vc13**
2	0x0	**err_intr_stat_pd_crc_err_vc13**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc13**
0	0x0	**err_intr_stat_ppfsm_timeout_vc13**

NVCSI_STREAM_3_UNCORRECTABLE_ERR_INTR_STATUS_VC14_0

Offset: 0xa0cf
Byte Offset: 0x2833c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc14**
4	0x0	**err_intr_stat_ph_single_crc_err_vc14**
3	0x0	**err_intr_stat_pd_wc_short_err_vc14**
2	0x0	**err_intr_stat_pd_crc_err_vc14**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc14**
0	0x0	**err_intr_stat_ppfsm_timeout_vc14**

NVCSI_STREAM_3_UNCORRECTABLE_ERR_INTR_STATUS_VC15_0

Offset: 0xa0d0
Byte Offset: 0x28340
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc15**
4	0x0	**err_intr_stat_ph_single_crc_err_vc15**
3	0x0	**err_intr_stat_pd_wc_short_err_vc15**
2	0x0	**err_intr_stat_pd_crc_err_vc15**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc15**
0	0x0	**err_intr_stat_ppfsm_timeout_vc15**

NVCSI_STREAM_3_ERR_INTR_MASK_NOVC_0

Offset: 0xa0d1
Byte Offset: 0x28344
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**err_intr_mask_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**err_intr_mask_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_3_ERR_INTR_MASK_VC0_0

Offset: 0xa0d2
Byte Offset: 0x28348
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc0**
4	0x0	**err_intr_mask_ph_single_crc_err_vc0**
3	0x0	**err_intr_mask_pd_wc_short_err_vc0**
2	0x0	**err_intr_mask_pd_crc_err_vc0**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc0**
0	0x0	**err_intr_mask_ppfsm_timeout_vc0**

NVCSI_STREAM_3_ERR_INTR_MASK_VC1_0

Offset: 0xa0d3
Byte Offset: 0x2834c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc1**

Bit	Reset	Description
4	0x0	**err_intr_mask_ph_single_crc_err_vc1**
3	0x0	**err_intr_mask_pd_wc_short_err_vc1**
2	0x0	**err_intr_mask_pd_crc_err_vc1**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc1**
0	0x0	**err_intr_mask_ppfsm_timeout_vc1**

NVCSI_STREAM_3_ERR_INTR_MASK_VC2_0

Offset: 0xa0d4
Byte Offset: 0x28350
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc2**
4	0x0	**err_intr_mask_ph_single_crc_err_vc2**
3	0x0	**err_intr_mask_pd_wc_short_err_vc2**
2	0x0	**err_intr_mask_pd_crc_err_vc2**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc2**
0	0x0	**err_intr_mask_ppfsm_timeout_vc2**

NVCSI_STREAM_3_ERR_INTR_MASK_VC3_0

Offset: 0xa0d5
Byte Offset: 0x28354
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc3**
4	0x0	**err_intr_mask_ph_single_crc_err_vc3**
3	0x0	**err_intr_mask_pd_wc_short_err_vc3**
2	0x0	**err_intr_mask_pd_crc_err_vc3**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc3**
0	0x0	**err_intr_mask_ppfsm_timeout_vc3**

NVCSI_STREAM_3_ERR_INTR_MASK_VC4_0

Offset: 0xa0d6
Byte Offset: 0x28358
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc4**
4	0x0	**err_intr_mask_ph_single_crc_err_vc4**
3	0x0	**err_intr_mask_pd_wc_short_err_vc4**
2	0x0	**err_intr_mask_pd_crc_err_vc4**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc4**
0	0x0	**err_intr_mask_ppfsm_timeout_vc4**

NVCSI_STREAM_3_ERR_INTR_MASK_VC5_0

Offset: 0xa0d7
Byte Offset: 0x2835c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc5**
4	0x0	**err_intr_mask_ph_single_crc_err_vc5**
3	0x0	**err_intr_mask_pd_wc_short_err_vc5**
2	0x0	**err_intr_mask_pd_crc_err_vc5**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc5**
0	0x0	**err_intr_mask_ppfsm_timeout_vc5**

NVCSI_STREAM_3_ERR_INTR_MASK_VC6_0

Offset: 0xa0d8
Byte Offset: 0x28360
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc6**
4	0x0	**err_intr_mask_ph_single_crc_err_vc6**
3	0x0	**err_intr_mask_pd_wc_short_err_vc6**
2	0x0	**err_intr_mask_pd_crc_err_vc6**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc6**
0	0x0	**err_intr_mask_ppfsm_timeout_vc6**

NVCSI_STREAM_3_ERR_INTR_MASK_VC7_0

Offset: 0xa0d9
Byte Offset: 0x28364
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc7**
4	0x0	**err_intr_mask_ph_single_crc_err_vc7**
3	0x0	**err_intr_mask_pd_wc_short_err_vc7**
2	0x0	**err_intr_mask_pd_crc_err_vc7**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc7**
0	0x0	**err_intr_mask_ppfsm_timeout_vc7**

NVCSI_STREAM_3_ERR_INTR_MASK_VC8_0

Offset: 0xa0da
Byte Offset: 0x28368
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc8**
4	0x0	**err_intr_mask_ph_single_crc_err_vc8**
3	0x0	**err_intr_mask_pd_wc_short_err_vc8**
2	0x0	**err_intr_mask_pd_crc_err_vc8**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc8**
0	0x0	**err_intr_mask_ppfsm_timeout_vc8**

NVCSI_STREAM_3_ERR_INTR_MASK_VC9_0

Offset: 0xa0db
Byte Offset: 0x2836c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc9**
4	0x0	**err_intr_mask_ph_single_crc_err_vc9**
3	0x0	**err_intr_mask_pd_wc_short_err_vc9**
2	0x0	**err_intr_mask_pd_crc_err_vc9**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc9**
0	0x0	**err_intr_mask_ppfsm_timeout_vc9**

NVCSI_STREAM_3_ERR_INTR_MASK_VC10_0

Offset: 0xa0dc
Byte Offset: 0x28370
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc10**
4	0x0	**err_intr_mask_ph_single_crc_err_vc10**
3	0x0	**err_intr_mask_pd_wc_short_err_vc10**
2	0x0	**err_intr_mask_pd_crc_err_vc10**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc10**
0	0x0	**err_intr_mask_ppfsm_timeout_vc10**

NVCSI_STREAM_3_ERR_INTR_MASK_VC11_0

Offset: 0xa0dd
Byte Offset: 0x28374
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc11**
4	0x0	**err_intr_mask_ph_single_crc_err_vc11**
3	0x0	**err_intr_mask_pd_wc_short_err_vc11**
2	0x0	**err_intr_mask_pd_crc_err_vc11**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc11**
0	0x0	**err_intr_mask_ppfsm_timeout_vc11**

NVCSI_STREAM_3_ERR_INTR_MASK_VC12_0

Offset: 0xa0de
Byte Offset: 0x28378
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc12**
4	0x0	**err_intr_mask_ph_single_crc_err_vc12**
3	0x0	**err_intr_mask_pd_wc_short_err_vc12**
2	0x0	**err_intr_mask_pd_crc_err_vc12**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc12**
0	0x0	**err_intr_mask_ppfsm_timeout_vc12**

NVCSI_STREAM_3_ERR_INTR_MASK_VC13_0

Offset: 0xa0df
Byte Offset: 0x2837c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc13**
4	0x0	**err_intr_mask_ph_single_crc_err_vc13**
3	0x0	**err_intr_mask_pd_wc_short_err_vc13**
2	0x0	**err_intr_mask_pd_crc_err_vc13**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc13**
0	0x0	**err_intr_mask_ppfsm_timeout_vc13**

NVCSI_STREAM_3_ERR_INTR_MASK_VC14_0

Offset: 0xa0e0
Byte Offset: 0x28380
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc14**
4	0x0	**err_intr_mask_ph_single_crc_err_vc14**
3	0x0	**err_intr_mask_pd_wc_short_err_vc14**
2	0x0	**err_intr_mask_pd_crc_err_vc14**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc14**
0	0x0	**err_intr_mask_ppfsm_timeout_vc14**

NVCSI_STREAM_3_ERR_INTR_MASK_VC15_0

Offset: 0xa0e1
Byte Offset: 0x28384
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc15**
4	0x0	**err_intr_mask_ph_single_crc_err_vc15**
3	0x0	**err_intr_mask_pd_wc_short_err_vc15**
2	0x0	**err_intr_mask_pd_crc_err_vc15**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc15**
0	0x0	**err_intr_mask_ppfsm_timeout_vc15**

NVCSI_STREAM_3_ERR_INTR_TYPE_NOVC_0

Offset: 0xa0e2
Byte Offset: 0x28388
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**err_intr_type_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**err_intr_type_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_3_ERR_INTR_TYPE_VC0_0

Offset: 0xa0e3
Byte Offset: 0x2838c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc0**
4	0x0	**err_intr_type_ph_single_crc_err_vc0**

Bit	Reset	Description
3	0x0	**err_intr_type_pd_wc_short_err_vc0**
2	0x0	**err_intr_type_pd_crc_err_vc0**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc0**
0	0x0	**err_intr_type_ppfsm_timeout_vc0**

NVCSI_STREAM_3_ERR_INTR_TYPE_VC1_0

Offset: 0xa0e4
Byte Offset: 0x28390
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc1**
4	0x0	**err_intr_type_ph_single_crc_err_vc1**
3	0x0	**err_intr_type_pd_wc_short_err_vc1**
2	0x0	**err_intr_type_pd_crc_err_vc1**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc1**
0	0x0	**err_intr_type_ppfsm_timeout_vc1**

NVCSI_STREAM_3_ERR_INTR_TYPE_VC2_0

Offset: 0xa0e5
Byte Offset: 0x28394
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc2**

Bit	Reset	Description
4	0x0	**err_intr_type_ph_single_crc_err_vc2**
3	0x0	**err_intr_type_pd_wc_short_err_vc2**
2	0x0	**err_intr_type_pd_crc_err_vc2**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc2**
0	0x0	**err_intr_type_ppfsm_timeout_vc2**

NVCSI_STREAM_3_ERR_INTR_TYPE_VC3_0

Offset: 0xa0e6
Byte Offset: 0x28398
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc3**
4	0x0	**err_intr_type_ph_single_crc_err_vc3**
3	0x0	**err_intr_type_pd_wc_short_err_vc3**
2	0x0	**err_intr_type_pd_crc_err_vc3**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc3**
0	0x0	**err_intr_type_ppfsm_timeout_vc3**

NVCSI_STREAM_3_ERR_INTR_TYPE_VC4_0

Offset: 0xa0e7
Byte Offset: 0x2839c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc4**
4	0x0	**err_intr_type_ph_single_crc_err_vc4**
3	0x0	**err_intr_type_pd_wc_short_err_vc4**
2	0x0	**err_intr_type_pd_crc_err_vc4**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc4**
0	0x0	**err_intr_type_ppfsm_timeout_vc4**

NVCSI_STREAM_3_ERR_INTR_TYPE_VC5_0

Offset: 0xa0e8
Byte Offset: 0x283a0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc5**
4	0x0	**err_intr_type_ph_single_crc_err_vc5**
3	0x0	**err_intr_type_pd_wc_short_err_vc5**
2	0x0	**err_intr_type_pd_crc_err_vc5**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc5**
0	0x0	**err_intr_type_ppfsm_timeout_vc5**

NVCSI_STREAM_3_ERR_INTR_TYPE_VC6_0

Offset: 0xa0e9
Byte Offset: 0x283a4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc6**
4	0x0	**err_intr_type_ph_single_crc_err_vc6**
3	0x0	**err_intr_type_pd_wc_short_err_vc6**
2	0x0	**err_intr_type_pd_crc_err_vc6**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc6**
0	0x0	**err_intr_type_ppfsm_timeout_vc6**

NVCSI_STREAM_3_ERR_INTR_TYPE_VC7_0

Offset: 0xa0ea
Byte Offset: 0x283a8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc7**
4	0x0	**err_intr_type_ph_single_crc_err_vc7**
3	0x0	**err_intr_type_pd_wc_short_err_vc7**
2	0x0	**err_intr_type_pd_crc_err_vc7**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc7**
0	0x0	**err_intr_type_ppfsm_timeout_vc7**

NVCSI_STREAM_3_ERR_INTR_TYPE_VC8_0

Offset: 0xa0eb
Byte Offset: 0x283ac
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc8**
4	0x0	**err_intr_type_ph_single_crc_err_vc8**
3	0x0	**err_intr_type_pd_wc_short_err_vc8**
2	0x0	**err_intr_type_pd_crc_err_vc8**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc8**
0	0x0	**err_intr_type_ppfsm_timeout_vc8**

NVCSI_STREAM_3_ERR_INTR_TYPE_VC9_0

Offset: 0xa0ec
Byte Offset: 0x283b0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc9**
4	0x0	**err_intr_type_ph_single_crc_err_vc9**
3	0x0	**err_intr_type_pd_wc_short_err_vc9**
2	0x0	**err_intr_type_pd_crc_err_vc9**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc9**
0	0x0	**err_intr_type_ppfsm_timeout_vc9**

NVCSI_STREAM_3_ERR_INTR_TYPE_VC10_0

Offset: 0xa0ed
Byte Offset: 0x283b4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc10**
4	0x0	**err_intr_type_ph_single_crc_err_vc10**
3	0x0	**err_intr_type_pd_wc_short_err_vc10**
2	0x0	**err_intr_type_pd_crc_err_vc10**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc10**
0	0x0	**err_intr_type_ppfsm_timeout_vc10**

NVCSI_STREAM_3_ERR_INTR_TYPE_VC11_0

Offset: 0xa0ee
Byte Offset: 0x283b8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc11**
4	0x0	**err_intr_type_ph_single_crc_err_vc11**
3	0x0	**err_intr_type_pd_wc_short_err_vc11**
2	0x0	**err_intr_type_pd_crc_err_vc11**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc11**
0	0x0	**err_intr_type_ppfsm_timeout_vc11**

NVCSI_STREAM_3_ERR_INTR_TYPE_VC12_0

Offset: 0xa0ef
Byte Offset: 0x283bc
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc12**
4	0x0	**err_intr_type_ph_single_crc_err_vc12**
3	0x0	**err_intr_type_pd_wc_short_err_vc12**
2	0x0	**err_intr_type_pd_crc_err_vc12**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc12**
0	0x0	**err_intr_type_ppfsm_timeout_vc12**

NVCSI_STREAM_3_ERR_INTR_TYPE_VC13_0

Offset: 0xa0f0
Byte Offset: 0x283c0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc13**
4	0x0	**err_intr_type_ph_single_crc_err_vc13**
3	0x0	**err_intr_type_pd_wc_short_err_vc13**
2	0x0	**err_intr_type_pd_crc_err_vc13**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc13**
0	0x0	**err_intr_type_ppfsm_timeout_vc13**

NVCSI_STREAM_3_ERR_INTR_TYPE_VC14_0

Offset: 0xa0f1
Byte Offset: 0x283c4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc14**
4	0x0	**err_intr_type_ph_single_crc_err_vc14**
3	0x0	**err_intr_type_pd_wc_short_err_vc14**
2	0x0	**err_intr_type_pd_crc_err_vc14**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc14**
0	0x0	**err_intr_type_ppfsm_timeout_vc14**

NVCSI_STREAM_3_ERR_INTR_TYPE_VC15_0

Offset: 0xa0f2
Byte Offset: 0x283c8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc15**
4	0x0	**err_intr_type_ph_single_crc_err_vc15**
3	0x0	**err_intr_type_pd_wc_short_err_vc15**
2	0x0	**err_intr_type_pd_crc_err_vc15**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc15**
0	0x0	**err_intr_type_ppfsm_timeout_vc15**

NVCSI_STREAM_3_TPG_ENABLE_0

Offset: 0xa0f3
Byte Offset: 0x283cc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	DISABLE	**TPG_ENABLE**: Enable the TPG path **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_3_TPG_VC_ENABLE_0

Offset: 0xa0f4
Byte Offset: 0x283d0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15	DISABLE	**VC15_EN**: Enable bit for the VC15 **0** = DISABLE **1** = ENABLE
14	DISABLE	**VC14_EN**: Enable bit for the VC14 **0** = DISABLE **1** = ENABLE
13	DISABLE	**VC13_EN**: Enable bit for the VC13 **0** = DISABLE **1** = ENABLE
12	DISABLE	**VC12_EN**: Enable bit for the VC12 **0** = DISABLE **1** = ENABLE
11	DISABLE	**VC11_EN**: Enable bit for the VC11 **0** = DISABLE **1** = ENABLE
10	DISABLE	**VC10_EN**: Enable bit for the VC10 **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
9	DISABLE	**VC9_EN**: Enable bit for the VC9 **0** = DISABLE **1** = ENABLE
8	DISABLE	**VC8_EN**: Enable bit for the VC8 **0** = DISABLE **1** = ENABLE
7	DISABLE	**VC7_EN**: Enable bit for the VC7 **0** = DISABLE **1** = ENABLE
6	DISABLE	**VC6_EN**: Enable bit for the VC6 **0** = DISABLE **1** = ENABLE
5	DISABLE	**VC5_EN**: Enable bit for the VC5 **0** = DISABLE **1** = ENABLE
4	DISABLE	**VC4_EN**: Enable bit for the VC4 **0** = DISABLE **1** = ENABLE
3	DISABLE	**VC3_EN**: Enable bit for the VC3 **0** = DISABLE **1** = ENABLE
2	DISABLE	**VC2_EN**: Enable bit for the VC2 **0** = DISABLE **1** = ENABLE
1	DISABLE	**VC1_EN**: Enable bit for the VC1 **0** = DISABLE **1** = ENABLE
0	DISABLE	**VC0_EN**: Enable bit for the VC0 **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_3_LOAD_TPG_CFG_0

Offset: 0xa0f5
Byte Offset: 0x283d4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**LOAD**: Load the shadow register

NVCSI_STREAM_3_TPG_CTRL_0

Offset: 0xa0f6
Byte Offset: 0x283d8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000008a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1000,1010)

Bit	Reset	Description
7:4	0x8	**DATA_SPEED**: Control the data generation speed, valid range is 1~8.
3	ENABLE	**SKIP_LS_LE_PKT**: If the LS/LE packet need to generated. **0** = DISABLE **1** = ENABLE
2	DISABLE	**OVERRIDE_CRC**: Override the packet header CRC and payload CRC. **0** = DISABLE **1** = ENABLE
1	CORE	**DEST**: The TPG pattern is send to PP or send to CIL for TX. **0** = CIL **1** = CORE

Bit	Reset	Description
0	DPHY	**PHY_MODE**: CPHY or DPHY packet structure for TPG **0** = DPHY **1** = CPHY

NVCSI_STREAM_3_TPG_VBLANK_0
Offset: 0xa0f7
Byte Offset: 0x283dc
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000800 (0b0000,0000,0000,0000,0000,1000,0000,0000)

Bit	Reset	Description
31:0	0x800	**VBLANK**: The vblank value, unit is nvcsi clock cycles.

NVCSI_STREAM_3_TPG_HBLANK_0
Offset: 0xa0f8
Byte Offset: 0x283e0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000800 (0b0000,0000,0000,0000,0000,1000,0000,0000)

Bit	Reset	Description
31:0	0x800	**HBLANK**: The hblank value, unit is nvcsi clock cycles.

NVCSI_STREAM_3_TPG_STATUS_0
Offset: 0xa0f9
Byte Offset: 0x283e4
Read/Write: RO
Parity Protection: N
Shadow: N

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0001ffff (0bxxxx,xxxx,xxxx,xxx1,1111,1111,1111,1111)

Bit	Reset	Description
16	IDLE	**VC15_STATUS**: Indicate the VC15 status **0** = BUSY **1** = IDLE
15	IDLE	**VC14_STATUS**: Indicate the VC14 status **0** = BUSY **1** = IDLE
14	IDLE	**VC13_STATUS**: Indicate the VC13 status **0** = BUSY **1** = IDLE
13	IDLE	**VC12_STATUS**: Indicate the VC12 status **0** = BUSY **1** = IDLE
12	IDLE	**VC11_STATUS**: Indicate the VC11 status **0** = BUSY **1** = IDLE
11	IDLE	**VC10_STATUS**: Indicate the VC10 status **0** = BUSY **1** = IDLE
10	IDLE	**VC9_STATUS**: Indicate the VC9 status **0** = BUSY **1** = IDLE
9	IDLE	**VC8_STATUS**: Indicate the VC8 status **0** = BUSY **1** = IDLE
8	IDLE	**VC7_STATUS**: Indicate the VC7 status **0** = BUSY **1** = IDLE

Bit	Reset	Description
7	IDLE	**VC6_STATUS**: Indicate the VC6 status **0** = BUSY **1** = IDLE
6	IDLE	**VC5_STATUS**: Indicate the VC5 status **0** = BUSY **1** = IDLE
5	IDLE	**VC4_STATUS**: Indicate the VC4 status **0** = BUSY **1** = IDLE
4	IDLE	**VC3_STATUS**: Indicate the VC3 status **0** = BUSY **1** = IDLE
3	IDLE	**VC2_STATUS**: Indicate the VC2 status **0** = BUSY **1** = IDLE
2	IDLE	**VC1_STATUS**: Indicate the VC1 status **0** = BUSY **1** = IDLE
1	IDLE	**VC0_STATUS**: Indicate the VC0 status **0** = BUSY **1** = IDLE
0	IDLE	**STATUS**: Indicate the TPG is in idle state, all packet has been send and all VC are disabled. **0** = BUSY **1** = IDLE

NVCSI_STREAM_3_TPG_PH_ECC_0

Offset: 0xa0fa
Byte Offset: 0x283e8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:24	0x0	**LINE_ECC**: This field is for the long packet header ECC.
23:18	0x0	**EOL_ECC**: This field is for the EOL short packet ECC.
17:12	0x0	**SOL_ECC**: This field is for the SOL short packet ECC.
11:6	0x0	**EOF_ECC**: This field is for the EOF short packet ECC.
5:0	0x0	**SOF_ECC**: The TPG will not generate ECC for a packet. When using the TPG, SW should set the PP to skip the ecc check. To verify the ecc logic for safety BIST, SW can write a pre-calculated ECC for the TPG, when use with this mode, the TPG should generate a grescale pattern. This field is for the SOF short packet ECC.

NVCSI_STREAM_3_TPG_PF_CRC_0

Offset: 0xa0fb
Byte Offset: 0x283ec
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PF_CRC**: This field is for the long packet payload CRC override.

NVCSI_STREAM_3_TPG_PH_SOF_CRC_0

Offset: 0xa0fc
Byte Offset: 0x283f0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This field is for the CPHY SOF packet second packet header CRC override.
15:0	0x0	**CRC_0**: This field is for the CPHY SOF packet first packet header CRC override.

NVCSI_STREAM_3_TPG_PH_EOF_CRC_0

Offset: 0xa0fd
Byte Offset: 0x283f4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This field is for the CPHY EOF packet second packet header CRC override.
15:0	0x0	**CRC_0**: This field is for the CPHY EOF packet first packet header CRC override.

NVCSI_STREAM_3_TPG_PH_SOL_CRC_0

Offset: 0xa0fe
Byte Offset: 0x283f8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This field is for the CPHY SOL packet second packet header CRC override.
15:0	0x0	**CRC_0**: This field is for the CPHY SOL packet first packet header CRC override.

NVCSI_STREAM_3_TPG_PH_EOL_CRC_0

Offset: 0xa0ff
Byte Offset: 0x283fc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This field is for the CPHY EOL packet second packet header CRC override.
15:0	0x0	**CRC_0**: This field is for the CPHY EOL packet first packet header CRC override.

NVCSI_STREAM_3_TPG_PH_LONG_PKT_CRC_0

Offset: 0xa100
Byte Offset: 0x28400
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This filed is for the CPHY long packet second packet header CRC override.
15:0	0x0	**CRC_0**: This filed is for the CPHY long packet first packet header CRC override.

NVCSI_STREAM_3_TPG_PKT_DELIMETER_0

Offset: 0xa101
Byte Offset: 0x28404
Read/Write: R/W
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0011)

Bit	Reset	Description
15:0	0x3	**NUM**: This field define the cycle number between two packet. Model the LP11 period between two packet, valid range is 1 to 65535.

NVCSI_STREAM_3_VC0_TPG_GAIN_CTRL_0

Offset: 0xa102
Byte Offset: 0x28408
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_3_VC1_TPG_GAIN_CTRL_0

Offset: 0xa103
Byte Offset: 0x2840c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_3_VC2_TPG_GAIN_CTRL_0

Offset: 0xa104
Byte Offset: 0x28410
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_3_VC3_TPG_GAIN_CTRL_0

Offset: 0xa105
Byte Offset: 0x28414
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_3_VC4_TPG_GAIN_CTRL_0

Offset: 0xa106
Byte Offset: 0x28418
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_3_VC5_TPG_GAIN_CTRL_0

Offset: 0xa107
Byte Offset: 0x2841c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_3_VC6_TPG_GAIN_CTRL_0

Offset: 0xa108
Byte Offset: 0x28420
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_3_VC7_TPG_GAIN_CTRL_0

Offset: 0xa109
Byte Offset: 0x28424
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_3_VC8_TPG_GAIN_CTRL_0

Offset: 0xa10a
Byte Offset: 0x28428
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_3_VC9_TPG_GAIN_CTRL_0

Offset: 0xa10b
Byte Offset: 0x2842c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_3_VC10_TPG_GAIN_CTRL_0

Offset: 0xa10c
Byte Offset: 0x28430
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_3_VC11_TPG_GAIN_CTRL_0

Offset: 0xa10d
Byte Offset: 0x28434
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_3_VC12_TPG_GAIN_CTRL_0

Offset: 0xa10e
Byte Offset: 0x28438
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_3_VC13_TPG_GAIN_CTRL_0

Offset: 0xa10f
Byte Offset: 0x2843c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_3_VC14_TPG_GAIN_CTRL_0

Offset: 0xa110
Byte Offset: 0x28440
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_3_VC15_TPG_GAIN_CTRL_0

Offset: 0xa111
Byte Offset: 0x28444
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_3_SPARE_0
Offset: 0xa112
Byte Offset: 0x28448
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**stream_reg**

7.2.1.3.5 NVCSI Stream4 Registers

NVCSI_STREAM_4_SW_RESET_CTRL_0

Offset: 0xc000
Byte Offset: 0x30000
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DISABLE	**cfg_swreset**: Reset the pixel parser. **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_4_SLCG_CTRL_0

Offset: 0xc001
Byte Offset: 0x30004
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DISABLE	**cfg_slcg_override**: Enable the SLCG override for the pixel parser. **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_4_PP_PHY_CTRL_0

Offset: 0xc002
Byte Offset: 0x30008
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DPHY	**cfg_phy_mode**: **0** = DPHY **1** = CPHY

NVCSI_STREAM_4_PP_EN_CTRL_0

Offset: 0xc003
Byte Offset: 0x3000c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	CIL	**cfg_src**: The pixel source of the pixel parser **0** = CIL **1** = TPG
0	DISABLE	**cfg_pp_en**: Pixel Parser streaming enable **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_4_VC0_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xc004
Byte Offset: 0x30010
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc0_dt_nooverride_0**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC0_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xc005
Byte Offset: 0x30014
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc0_dt_nooverride_1**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC0_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xc006
Byte Offset: 0x30018
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc0_dt_nooverride_2**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC0_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xc007
Byte Offset: 0x3001c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc0_dt_nooverride_3**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC0_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xc008
Byte Offset: 0x30020
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc0_dt_nooverride_4**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC0_DT_OVERRIDE_0

Offset: 0xc009
Byte Offset: 0x30024
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc0_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc0_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc0_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_4_VC1_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xc00a
Byte Offset: 0x30028
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc1_dt_nooverride_0**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC1_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xc00b
Byte Offset: 0x3002c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc1_dt_nooverride_1**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC1_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xc00c
Byte Offset: 0x30030
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc1_dt_nooverride_2**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC1_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xc00d
Byte Offset: 0x30034
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc1_dt_nooverride_3**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC1_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xc00e
Byte Offset: 0x30038
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc1_dt_nooverride_4**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC1_DT_OVERRIDE_0

Offset: 0xc00f
Byte Offset: 0x3003c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc1_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc1_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc1_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_4_VC2_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xc010
Byte Offset: 0x30040
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc2_dt_nooverride_0**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC2_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xc011
Byte Offset: 0x30044
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc2_dt_nooverride_1**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC2_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xc012
Byte Offset: 0x30048
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc2_dt_nooverride_2**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC2_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xc013
Byte Offset: 0x3004c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc2_dt_nooverride_3**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC2_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xc014
Byte Offset: 0x30050
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc2_dt_nooverride_4**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC2_DT_OVERRIDE_0

Offset: 0xc015
Byte Offset: 0x30054
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc2_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc2_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc2_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_4_VC3_DT_NOOVERRIDE_CTRL_0_0

VC3

Offset: 0xc016
Byte Offset: 0x30058
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc3_dt_nooverride_0**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC3_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xc017
Byte Offset: 0x3005c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc3_dt_nooverride_1**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC3_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xc018
Byte Offset: 0x30060
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc3_dt_nooverride_2**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC3_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xc019
Byte Offset: 0x30064
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc3_dt_nooverride_3**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC3_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xc01a
Byte Offset: 0x30068
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc3_dt_nooverride_4**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC3_DT_OVERRIDE_0

Offset: 0xc01b
Byte Offset: 0x3006c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc3_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc3_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc3_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_4_VC4_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xc01c
Byte Offset: 0x30070
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc4_dt_nooverride_0**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC4_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xc01d
Byte Offset: 0x30074
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc4_dt_nooverride_1**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC4_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xc01e
Byte Offset: 0x30078
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc4_dt_nooverride_2**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC4_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xc01f
Byte Offset: 0x3007c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc4_dt_nooverride_3**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC4_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xc020
Byte Offset: 0x30080
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc4_dt_nooverride_4**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC4_DT_OVERRIDE_0

Offset: 0xc021
Byte Offset: 0x30084
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc4_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc4_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc4_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_4_VC5_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xc022
Byte Offset: 0x30088
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc5_dt_nooverride_0**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC5_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xc023
Byte Offset: 0x3008c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc5_dt_nooverride_1**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC5_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xc024
Byte Offset: 0x30090
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc5_dt_nooverride_2**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC5_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xc025
Byte Offset: 0x30094
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc5_dt_nooverride_3**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC5_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xc026
Byte Offset: 0x30098
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc5_dt_nooverride_4**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC5_DT_OVERRIDE_0

Offset: 0xc027
Byte Offset: 0x3009c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc5_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc5_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc5_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_4_VC6_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xc028
Byte Offset: 0x300a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc6_dt_nooverride_0**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC6_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xc029
Byte Offset: 0x300a4
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc6_dt_nooverride_1**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC6_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xc02a
Byte Offset: 0x300a8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc6_dt_nooverride_2**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC6_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xc02b
Byte Offset: 0x300ac
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc6_dt_nooverride_3**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC6_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xc02c
Byte Offset: 0x300b0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc6_dt_nooverride_4**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC6_DT_OVERRIDE_0

Offset: 0xc02d
Byte Offset: 0x300b4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc6_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc6_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc6_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_4_VC7_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xc02e
Byte Offset: 0x300b8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc7_dt_nooverride_0**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC7_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xc02f
Byte Offset: 0x300bc
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc7_dt_nooverride_1**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC7_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xc030
Byte Offset: 0x300c0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc7_dt_nooverride_2**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC7_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xc031
Byte Offset: 0x300c4
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc7_dt_nooverride_3**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC7_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xc032
Byte Offset: 0x300c8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc7_dt_nooverride_4**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC7_DT_OVERRIDE_0

Offset: 0xc033
Byte Offset: 0x300cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc7_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc7_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc7_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_4_VC8_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xc034
Byte Offset: 0x300d0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc8_dt_nooverride_0**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC8_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xc035
Byte Offset: 0x300d4
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc8_dt_nooverride_1**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC8_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xc036
Byte Offset: 0x300d8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc8_dt_nooverride_2**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC8_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xc037
Byte Offset: 0x300dc
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc8_dt_nooverride_3**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC8_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xc038
Byte Offset: 0x300e0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc8_dt_nooverride_4**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC8_DT_OVERRIDE_0

Offset: 0xc039
Byte Offset: 0x300e4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc8_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc8_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc8_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_4_VC9_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xc03a
Byte Offset: 0x300e8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc9_dt_nooverride_0**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC9_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xc03b
Byte Offset: 0x300ec
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc9_dt_nooverride_1**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC9_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xc03c
Byte Offset: 0x300f0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc9_dt_nooverride_2**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC9_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xc03d
Byte Offset: 0x300f4
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc9_dt_nooverride_3**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC9_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xc03e
Byte Offset: 0x300f8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc9_dt_nooverride_4**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC9_DT_OVERRIDE_0

Offset: 0xc03f
Byte Offset: 0x300fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc9_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc9_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc9_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_4_VC10_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xc040
Byte Offset: 0x30100
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc10_dt_nooverride_0**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC10_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xc041
Byte Offset: 0x30104
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc10_dt_nooverride_1**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC10_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xc042
Byte Offset: 0x30108
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc10_dt_nooverride_2**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC10_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xc043
Byte Offset: 0x3010c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc10_dt_nooverride_3**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC10_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xc044
Byte Offset: 0x30110
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc10_dt_nooverride_4**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC10_DT_OVERRIDE_0

Offset: 0xc045
Byte Offset: 0x30114
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc10_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc10_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc10_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_4_VC11_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xc046
Byte Offset: 0x30118
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc11_dt_nooverride_0**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC11_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xc047
Byte Offset: 0x3011c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc11_dt_nooverride_1**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC11_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xc048
Byte Offset: 0x30120
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc11_dt_nooverride_2**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC11_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xc049
Byte Offset: 0x30124
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc11_dt_nooverride_3**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC11_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xc04a
Byte Offset: 0x30128
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc11_dt_nooverride_4**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC11_DT_OVERRIDE_0

Offset: 0xc04b
Byte Offset: 0x3012c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc11_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc11_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc11_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_4_VC12_DT_NOOVERRIDE_CTRL_0_0

VC12

Offset: 0xc04c
Byte Offset: 0x30130
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc12_dt_nooverride_0**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC12_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xc04d
Byte Offset: 0x30134
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc12_dt_nooverride_1**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC12_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xc04e
Byte Offset: 0x30138
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc12_dt_nooverride_2**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC12_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xc04f
Byte Offset: 0x3013c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc12_dt_nooverride_3**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC12_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xc050
Byte Offset: 0x30140
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc12_dt_nooverride_4**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC12_DT_OVERRIDE_0

Offset: 0xc051
Byte Offset: 0x30144
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc12_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc12_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc12_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_4_VC13_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xc052
Byte Offset: 0x30148
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc13_dt_nooverride_0**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC13_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xc053
Byte Offset: 0x3014c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc13_dt_nooverride_1**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC13_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xc054
Byte Offset: 0x30150
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc13_dt_nooverride_2**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC13_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xc055
Byte Offset: 0x30154
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc13_dt_nooverride_3**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC13_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xc056
Byte Offset: 0x30158
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc13_dt_nooverride_4**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC13_DT_OVERRIDE_0

Offset: 0xc057
Byte Offset: 0x3015c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc13_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc13_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc13_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_4_VC14_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xc058
Byte Offset: 0x30160
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc14_dt_nooverride_0**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC14_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xc059
Byte Offset: 0x30164
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc14_dt_nooverride_1**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC14_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xc05a
Byte Offset: 0x30168
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc14_dt_nooverride_2**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC14_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xc05b
Byte Offset: 0x3016c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc14_dt_nooverride_3**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC14_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xc05c
Byte Offset: 0x30170
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc14_dt_nooverride_4**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC14_DT_OVERRIDE_0

Offset: 0xc05d
Byte Offset: 0x30174
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc14_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc14_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc14_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_4_VC15_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xc05e
Byte Offset: 0x30178
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc15_dt_nooverride_0**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC15_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xc05f
Byte Offset: 0x3017c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc15_dt_nooverride_1**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC15_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xc060
Byte Offset: 0x30180
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc15_dt_nooverride_2**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC15_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xc061
Byte Offset: 0x30184
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc15_dt_nooverride_3**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC15_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xc062
Byte Offset: 0x30188
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc15_dt_nooverride_4**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_4_VC15_DT_OVERRIDE_0

Offset: 0xc063
Byte Offset: 0x3018c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc15_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc15_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc15_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_4_PPFSM_TIMEOUT_CTRL_0

Offset: 0xc064
Byte Offset: 0x30190
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x7fffffff (0b0111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31	DISABLE	**cfg_timeout_en**: Enable Timeout counter for the PP FSM **0** = DISABLE **1** = ENABLE
30:0	0x7fffffff	**cfg_timeout_period**: Timeout period

NVCSI_STREAM_4_PH_CHK_CTRL_0

Offset: 0xc065
Byte Offset: 0x30194
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	ENABLE	**cfg_ph_16_vc**: 16 VC support **0** = DISABLE **1** = ENABLE
1	ENABLE	**cfg_ph_crc_chk_en**: PH CRC check enable (only for CPHY case), when this bit is set to 0, the packet header will still be decode when the CRC check fail, but the error will be set. **0** = DISABLE **1** = ENABLE
0	ENABLE	**cfg_ph_ecc_chk_en**: PH ECC check enable (only for DPHY case), when this bit is set to 0, the packet header will still be decode when the ECC check fail, but the error will be set. **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_4_VC0_DPCM_CTRL_0

Offset: 0xc066
Byte Offset: 0x30198
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc0_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_4_VC1_DPCM_CTRL_0

Offset: 0xc067
Byte Offset: 0x3019c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc1_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_4_VC2_DPCM_CTRL_0

Offset: 0xc068
Byte Offset: 0x301a0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc2_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_4_VC3_DPCM_CTRL_0

Offset: 0xc069
Byte Offset: 0x301a4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc3_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_4_VC4_DPCM_CTRL_0

Offset: 0xc06a
Byte Offset: 0x301a8
Read/Write: R/W
Parity Protection: Y
Shadow: N

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc4_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_4_VC5_DPCM_CTRL_0

Offset: 0xc06b
Byte Offset: 0x301ac
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc5_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_4_VC6_DPCM_CTRL_0

Offset: 0xc06c
Byte Offset: 0x301b0
Read/Write: R/W
Parity Protection: Y

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc6_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_4_VC7_DPCM_CTRL_0

Offset: 0xc06d
Byte Offset: 0x301b4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc7_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_4_VC8_DPCM_CTRL_0

Offset: 0xc06e
Byte Offset: 0x301b8
Read/Write: R/W

Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc8_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_4_VC9_DPCM_CTRL_0

Offset: 0xc06f
Byte Offset: 0x301bc
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc9_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_4_VC10_DPCM_CTRL_0

Offset: 0xc070
Byte Offset: 0x301c0

Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc10_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_4_VC11_DPCM_CTRL_0

Offset: 0xc071
Byte Offset: 0x301c4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc11_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_4_VC12_DPCM_CTRL_0

Offset: 0xc072

Byte Offset: 0x301c8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc12_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_4_VC13_DPCM_CTRL_0

Offset: 0xc073
Byte Offset: 0x301cc
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc13_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_4_VC14_DPCM_CTRL_0

Offset: 0xc074
Byte Offset: 0x301d0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc14_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_4_VC15_DPCM_CTRL_0

Offset: 0xc075
Byte Offset: 0x301d4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc15_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_4_PF_CRC_0

Status register on packet data CRC
Offset: 0xc076
Byte Offset: 0x301d8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**calc_crc**: 16 bit CRC computed over current packet (has to match with the CRC in PF for a good packet)
15:0	0x0	**rx_crc**: 16 bit CRC from PF

NVCSI_STREAM_4_PH_WC_0

Status register on WC
Offset: 0xc077
Byte Offset: 0x301dc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**calc_wc**: 16 bit WC computed over current packet (has to match with the WC in PH for a good packet)
15:0	0x0	**rx_wc**: 16 bit WC from PH

NVCSI_STREAM_4_PH_DI_0

Status register on Data ID
Offset: 0xc078
Byte Offset: 0x301e0
Read/Write: RO
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9:6	0x0	**rx_vc**: 4 bit VC from PH
5:0	0x0	**rx_dt**: 6 bit DTYPE from PH

NVCSI_STREAM_4_ERROR_STATUS2VI_MASK_0

Offset: 0xc079
Byte Offset: 0x301e4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15	0x0	**cfg_err_status2vi_mask_vc15**: for VC15 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
14	0x0	**cfg_err_status2vi_mask_vc14**: for VC14 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
13	0x0	**cfg_err_status2vi_mask_vc13**: for VC13 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
12	0x0	**cfg_err_status2vi_mask_vc12**: for VC12 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
11	0x0	**cfg_err_status2vi_mask_vc11**: for VC11 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
10	0x0	**cfg_err_status2vi_mask_vc10**: for VC10 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
9	0x0	**cfg_err_status2vi_mask_vc9**: for VC9 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF

Bit	Reset	Description
8	0x0	**cfg_err_status2vi_mask_vc8**: for VC8 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
7	0x0	**cfg_err_status2vi_mask_vc7**: for VC7 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
6	0x0	**cfg_err_status2vi_mask_vc6**: for VC6 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
5	0x0	**cfg_err_status2vi_mask_vc5**: for VC5 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
4	0x0	**cfg_err_status2vi_mask_vc4**: for VC4 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
3	0x0	**cfg_err_status2vi_mask_vc3**: for VC3 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
2	0x0	**cfg_err_status2vi_mask_vc2**: for VC2 : 0 = err_status2vi_vc2 will be send to VI at EOF; 1 = No error will be send to VI at EOF
1	0x0	**cfg_err_status2vi_mask_vc1**: for VC1 : 0 = err_status2vi_vc1 will be send to VI at EOF; 1 = No error will be send to VI at EOF
0	0x0	**cfg_err_status2vi_mask_vc0**: for VC0 : 0 = err_status2vi_vc0 will be send to VI at EOF; 1 = No error will be send to VI at EOF

NVCSI_STREAM_4_ERROR_STATUS2VI_VC0_0

This is RO register for SW
Offset: 0xc07a
Byte Offset: 0x301e8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc0**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_4_ERROR_STATUS2VI_VC1_0

This is RO register for SW
Offset: 0xc07b
Byte Offset: 0x301ec
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc1**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_4_ERROR_STATUS2VI_VC2_0

This is RO register for SW
Offset: 0xc07c
Byte Offset: 0x301f0
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc2**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_4_ERROR_STATUS2VI_VC3_0

This is RO register for SW
Offset: 0xc07d
Byte Offset: 0x301f4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc3**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_4_ERROR_STATUS2VI_VC4_0

This is RO register for SW
Offset: 0xc07e
Byte Offset: 0x301f8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc4**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_4_ERROR_STATUS2VI_VC5_0

This is RO register for SW
Offset: 0xc07f
Byte Offset: 0x301fc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc5**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_4_ERROR_STATUS2VI_VC6_0

This is RO register for SW
Offset: 0xc080
Byte Offset: 0x30200
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc6**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_4_ERROR_STATUS2VI_VC7_0

This is RO register for SW
Offset: 0xc081
Byte Offset: 0x30204
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc7**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_4_ERROR_STATUS2VI_VC8_0

This is RO register for SW
Offset: 0xc082
Byte Offset: 0x30208
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc8**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_4_ERROR_STATUS2VI_VC9_0

This is RO register for SW
Offset: 0xc083
Byte Offset: 0x3020c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc9**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_4_ERROR_STATUS2VI_VC10_0

This is RO register for SW
Offset: 0xc084
Byte Offset: 0x30210
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc10**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_4_ERROR_STATUS2VI_VC11_0

This is RO register for SW
Offset: 0xc085
Byte Offset: 0x30214
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc11**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_4_ERROR_STATUS2VI_VC12_0

This is RO register for SW
Offset: 0xc086
Byte Offset: 0x30218
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc12**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_4_ERROR_STATUS2VI_VC13_0

This is RO register for SW
Offset: 0xc087
Byte Offset: 0x3021c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc13**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_4_ERROR_STATUS2VI_VC14_0

This is RO register for SW
Offset: 0xc088
Byte Offset: 0x30220
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc14**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_4_ERROR_STATUS2VI_VC15_0

This is RO register for SW
Offset: 0xc089
Byte Offset: 0x30224
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc15**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_4_INTR_STATUS_0

Offset: 0xc08a
Byte Offset: 0x30228
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**intr_stat_novc**: Not VC related interrupt
15	0x0	**intr_stat_vc15**: VC15 event

Bit	Reset	Description
14	0x0	**intr_stat_vc14**: VC14 event
13	0x0	**intr_stat_vc13**: VC13 event
12	0x0	**intr_stat_vc12**: VC12 event
11	0x0	**intr_stat_vc11**: VC11 event
10	0x0	**intr_stat_vc10**: VC10 event
9	0x0	**intr_stat_vc9**: VC9 event
8	0x0	**intr_stat_vc8**: VC8 event
7	0x0	**intr_stat_vc7**: VC7 event
6	0x0	**intr_stat_vc6**: VC6 event
5	0x0	**intr_stat_vc5**: VC5 event
4	0x0	**intr_stat_vc4**: VC4 event
3	0x0	**intr_stat_vc3**: VC3 event
2	0x0	**intr_stat_vc2**: VC2 event
1	0x0	**intr_stat_vc1**: VC1 event
0	0x0	**intr_stat_vc0**: VC0 event

NVCSI_STREAM_4_INTR_STATUS_NOVC_0

Offset: 0xc08b
Byte Offset: 0x3022c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**intr_stat_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**intr_stat_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_4_INTR_STATUS_VC0_0

Offset: 0xc08c
Byte Offset: 0x30230
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc0**
4	0x0	**intr_stat_ph_single_crc_err_vc0**
3	0x0	**intr_stat_pd_wc_short_err_vc0**
2	0x0	**intr_stat_pd_crc_err_vc0**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc0**
0	0x0	**intr_stat_ppfsm_timeout_vc0**

NVCSI_STREAM_4_INTR_STATUS_VC1_0

Offset: 0xc08d
Byte Offset: 0x30234
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc1**

Bit	Reset	Description
4	0x0	**intr_stat_ph_single_crc_err_vc1**
3	0x0	**intr_stat_pd_wc_short_err_vc1**
2	0x0	**intr_stat_pd_crc_err_vc1**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc1**
0	0x0	**intr_stat_ppfsm_timeout_vc1**

NVCSI_STREAM_4_INTR_STATUS_VC2_0

Offset: 0xc08e
Byte Offset: 0x30238
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc2**
4	0x0	**intr_stat_ph_single_crc_err_vc2**
3	0x0	**intr_stat_pd_wc_short_err_vc2**
2	0x0	**intr_stat_pd_crc_err_vc2**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc2**
0	0x0	**intr_stat_ppfsm_timeout_vc2**

NVCSI_STREAM_4_INTR_STATUS_VC3_0

Offset: 0xc08f
Byte Offset: 0x3023c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc3**
4	0x0	**intr_stat_ph_single_crc_err_vc3**
3	0x0	**intr_stat_pd_wc_short_err_vc3**
2	0x0	**intr_stat_pd_crc_err_vc3**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc3**
0	0x0	**intr_stat_ppfsm_timeout_vc3**

NVCSI_STREAM_4_INTR_STATUS_VC4_0

Offset: 0xc090
Byte Offset: 0x30240
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc4**
4	0x0	**intr_stat_ph_single_crc_err_vc4**
3	0x0	**intr_stat_pd_wc_short_err_vc4**
2	0x0	**intr_stat_pd_crc_err_vc4**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc4**
0	0x0	**intr_stat_ppfsm_timeout_vc4**

NVCSI_STREAM_4_INTR_STATUS_VC5_0

Offset: 0xc091
Byte Offset: 0x30244
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc5**
4	0x0	**intr_stat_ph_single_crc_err_vc5**
3	0x0	**intr_stat_pd_wc_short_err_vc5**
2	0x0	**intr_stat_pd_crc_err_vc5**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc5**
0	0x0	**intr_stat_ppfsm_timeout_vc5**

NVCSI_STREAM_4_INTR_STATUS_VC6_0

Offset: 0xc092
Byte Offset: 0x30248
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc6**
4	0x0	**intr_stat_ph_single_crc_err_vc6**
3	0x0	**intr_stat_pd_wc_short_err_vc6**
2	0x0	**intr_stat_pd_crc_err_vc6**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc6**
0	0x0	**intr_stat_ppfsm_timeout_vc6**

NVCSI_STREAM_4_INTR_STATUS_VC7_0

Offset: 0xc093
Byte Offset: 0x3024c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc7**
4	0x0	**intr_stat_ph_single_crc_err_vc7**
3	0x0	**intr_stat_pd_wc_short_err_vc7**
2	0x0	**intr_stat_pd_crc_err_vc7**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc7**
0	0x0	**intr_stat_ppfsm_timeout_vc7**

NVCSI_STREAM_4_INTR_STATUS_VC8_0

Offset: 0xc094
Byte Offset: 0x30250
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc8**
4	0x0	**intr_stat_ph_single_crc_err_vc8**
3	0x0	**intr_stat_pd_wc_short_err_vc8**
2	0x0	**intr_stat_pd_crc_err_vc8**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc8**
0	0x0	**intr_stat_ppfsm_timeout_vc8**

NVCSI_STREAM_4_INTR_STATUS_VC9_0

Offset: 0xc095
Byte Offset: 0x30254
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc9**
4	0x0	**intr_stat_ph_single_crc_err_vc9**
3	0x0	**intr_stat_pd_wc_short_err_vc9**
2	0x0	**intr_stat_pd_crc_err_vc9**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc9**
0	0x0	**intr_stat_ppfsm_timeout_vc9**

NVCSI_STREAM_4_INTR_STATUS_VC10_0

Offset: 0xc096
Byte Offset: 0x30258
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc10**
4	0x0	**intr_stat_ph_single_crc_err_vc10**
3	0x0	**intr_stat_pd_wc_short_err_vc10**
2	0x0	**intr_stat_pd_crc_err_vc10**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc10**
0	0x0	**intr_stat_ppfsm_timeout_vc10**

NVCSI_STREAM_4_INTR_STATUS_VC11_0

Offset: 0xc097
Byte Offset: 0x3025c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc11**
4	0x0	**intr_stat_ph_single_crc_err_vc11**
3	0x0	**intr_stat_pd_wc_short_err_vc11**
2	0x0	**intr_stat_pd_crc_err_vc11**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc11**
0	0x0	**intr_stat_ppfsm_timeout_vc11**

NVCSI_STREAM_4_INTR_STATUS_VC12_0

Offset: 0xc098
Byte Offset: 0x30260
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc12**
4	0x0	**intr_stat_ph_single_crc_err_vc12**
3	0x0	**intr_stat_pd_wc_short_err_vc12**
2	0x0	**intr_stat_pd_crc_err_vc12**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc12**
0	0x0	**intr_stat_ppfsm_timeout_vc12**

NVCSI_STREAM_4_INTR_STATUS_VC13_0

Offset: 0xc099
Byte Offset: 0x30264
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc13**
4	0x0	**intr_stat_ph_single_crc_err_vc13**
3	0x0	**intr_stat_pd_wc_short_err_vc13**
2	0x0	**intr_stat_pd_crc_err_vc13**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc13**
0	0x0	**intr_stat_ppfsm_timeout_vc13**

NVCSI_STREAM_4_INTR_STATUS_VC14_0

Offset: 0xc09a
Byte Offset: 0x30268
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc14**
4	0x0	**intr_stat_ph_single_crc_err_vc14**
3	0x0	**intr_stat_pd_wc_short_err_vc14**
2	0x0	**intr_stat_pd_crc_err_vc14**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc14**
0	0x0	**intr_stat_ppfsm_timeout_vc14**

NVCSI_STREAM_4_INTR_STATUS_VC15_0

Offset: 0xc09b
Byte Offset: 0x3026c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc15**
4	0x0	**intr_stat_ph_single_crc_err_vc15**
3	0x0	**intr_stat_pd_wc_short_err_vc15**
2	0x0	**intr_stat_pd_crc_err_vc15**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc15**
0	0x0	**intr_stat_ppfsm_timeout_vc15**

NVCSI_STREAM_4_INTR_MASK_NOVC_0

Offset: 0xc09c
Byte Offset: 0x30270
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**intr_mask_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**intr_mask_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_4_INTR_MASK_VC0_0

Offset: 0xc09d
Byte Offset: 0x30274
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc0**
4	0x0	**intr_mask_ph_single_crc_err_vc0**

Bit	Reset	Description
3	0x0	**intr_mask_pd_wc_short_err_vc0**
2	0x0	**intr_mask_pd_crc_err_vc0**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc0**
0	0x0	**intr_mask_ppfsm_timeout_vc0**

NVCSI_STREAM_4_INTR_MASK_VC1_0

Offset: 0xc09e
Byte Offset: 0x30278
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc1**
4	0x0	**intr_mask_ph_single_crc_err_vc1**
3	0x0	**intr_mask_pd_wc_short_err_vc1**
2	0x0	**intr_mask_pd_crc_err_vc1**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc1**
0	0x0	**intr_mask_ppfsm_timeout_vc1**

NVCSI_STREAM_4_INTR_MASK_VC2_0

Offset: 0xc09f
Byte Offset: 0x3027c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc2**

Bit	Reset	Description
4	0x0	**intr_mask_ph_single_crc_err_vc2**
3	0x0	**intr_mask_pd_wc_short_err_vc2**
2	0x0	**intr_mask_pd_crc_err_vc2**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc2**
0	0x0	**intr_mask_ppfsm_timeout_vc2**

NVCSI_STREAM_4_INTR_MASK_VC3_0

Offset: 0xc0a0
Byte Offset: 0x30280
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc3**
4	0x0	**intr_mask_ph_single_crc_err_vc3**
3	0x0	**intr_mask_pd_wc_short_err_vc3**
2	0x0	**intr_mask_pd_crc_err_vc3**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc3**
0	0x0	**intr_mask_ppfsm_timeout_vc3**

NVCSI_STREAM_4_INTR_MASK_VC4_0

Offset: 0xc0a1
Byte Offset: 0x30284
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc4**
4	0x0	**intr_mask_ph_single_crc_err_vc4**
3	0x0	**intr_mask_pd_wc_short_err_vc4**
2	0x0	**intr_mask_pd_crc_err_vc4**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc4**
0	0x0	**intr_mask_ppfsm_timeout_vc4**

NVCSI_STREAM_4_INTR_MASK_VC5_0

Offset: 0xc0a2
Byte Offset: 0x30288
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc5**
4	0x0	**intr_mask_ph_single_crc_err_vc5**
3	0x0	**intr_mask_pd_wc_short_err_vc5**
2	0x0	**intr_mask_pd_crc_err_vc5**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc5**
0	0x0	**intr_mask_ppfsm_timeout_vc5**

NVCSI_STREAM_4_INTR_MASK_VC6_0

Offset: 0xc0a3
Byte Offset: 0x3028c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc6**
4	0x0	**intr_mask_ph_single_crc_err_vc6**
3	0x0	**intr_mask_pd_wc_short_err_vc6**
2	0x0	**intr_mask_pd_crc_err_vc6**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc6**
0	0x0	**intr_mask_ppfsm_timeout_vc6**

NVCSI_STREAM_4_INTR_MASK_VC7_0

Offset: 0xc0a4
Byte Offset: 0x30290
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc7**
4	0x0	**intr_mask_ph_single_crc_err_vc7**
3	0x0	**intr_mask_pd_wc_short_err_vc7**
2	0x0	**intr_mask_pd_crc_err_vc7**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc7**
0	0x0	**intr_mask_ppfsm_timeout_vc7**

NVCSI_STREAM_4_INTR_MASK_VC8_0

Offset: 0xc0a5
Byte Offset: 0x30294
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc8**
4	0x0	**intr_mask_ph_single_crc_err_vc8**
3	0x0	**intr_mask_pd_wc_short_err_vc8**
2	0x0	**intr_mask_pd_crc_err_vc8**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc8**
0	0x0	**intr_mask_ppfsm_timeout_vc8**

NVCSI_STREAM_4_INTR_MASK_VC9_0

Offset: 0xc0a6
Byte Offset: 0x30298
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc9**
4	0x0	**intr_mask_ph_single_crc_err_vc9**
3	0x0	**intr_mask_pd_wc_short_err_vc9**
2	0x0	**intr_mask_pd_crc_err_vc9**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc9**
0	0x0	**intr_mask_ppfsm_timeout_vc9**

NVCSI_STREAM_4_INTR_MASK_VC10_0

Offset: 0xc0a7
Byte Offset: 0x3029c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc10**
4	0x0	**intr_mask_ph_single_crc_err_vc10**
3	0x0	**intr_mask_pd_wc_short_err_vc10**
2	0x0	**intr_mask_pd_crc_err_vc10**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc10**
0	0x0	**intr_mask_ppfsm_timeout_vc10**

NVCSI_STREAM_4_INTR_MASK_VC11_0

Offset: 0xc0a8
Byte Offset: 0x302a0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc11**
4	0x0	**intr_mask_ph_single_crc_err_vc11**
3	0x0	**intr_mask_pd_wc_short_err_vc11**
2	0x0	**intr_mask_pd_crc_err_vc11**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc11**
0	0x0	**intr_mask_ppfsm_timeout_vc11**

NVCSI_STREAM_4_INTR_MASK_VC12_0

Offset: 0xc0a9
Byte Offset: 0x302a4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc12**
4	0x0	**intr_mask_ph_single_crc_err_vc12**
3	0x0	**intr_mask_pd_wc_short_err_vc12**
2	0x0	**intr_mask_pd_crc_err_vc12**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc12**
0	0x0	**intr_mask_ppfsm_timeout_vc12**

NVCSI_STREAM_4_INTR_MASK_VC13_0

Offset: 0xc0aa
Byte Offset: 0x302a8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc13**
4	0x0	**intr_mask_ph_single_crc_err_vc13**
3	0x0	**intr_mask_pd_wc_short_err_vc13**
2	0x0	**intr_mask_pd_crc_err_vc13**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc13**
0	0x0	**intr_mask_ppfsm_timeout_vc13**

NVCSI_STREAM_4_INTR_MASK_VC14_0

Offset: 0xc0ab
Byte Offset: 0x302ac
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc14**
4	0x0	**intr_mask_ph_single_crc_err_vc14**
3	0x0	**intr_mask_pd_wc_short_err_vc14**
2	0x0	**intr_mask_pd_crc_err_vc14**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc14**
0	0x0	**intr_mask_ppfsm_timeout_vc14**

NVCSI_STREAM_4_INTR_MASK_VC15_0

Offset: 0xc0ac
Byte Offset: 0x302b0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc15**
4	0x0	**intr_mask_ph_single_crc_err_vc15**
3	0x0	**intr_mask_pd_wc_short_err_vc15**
2	0x0	**intr_mask_pd_crc_err_vc15**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc15**
0	0x0	**intr_mask_ppfsm_timeout_vc15**

NVCSI_STREAM_4_CORRECTABLE_ERR_INTR_STATUS_0

Offset: 0xc0ad
Byte Offset: 0x302b4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**err_intr_stat_novc**: Not VC related interrupt
15	0x0	**err_intr_stat_vc15**: VC15 event
14	0x0	**err_intr_stat_vc14**: VC14 event
13	0x0	**err_intr_stat_vc13**: VC13 event
12	0x0	**err_intr_stat_vc12**: VC12 event
11	0x0	**err_intr_stat_vc11**: VC11 event
10	0x0	**err_intr_stat_vc10**: VC10 event
9	0x0	**err_intr_stat_vc9**: VC9 event
8	0x0	**err_intr_stat_vc8**: VC8 event
7	0x0	**err_intr_stat_vc7**: VC7 event
6	0x0	**err_intr_stat_vc6**: VC6 event
5	0x0	**err_intr_stat_vc5**: VC5 event
4	0x0	**err_intr_stat_vc4**: VC4 event
3	0x0	**err_intr_stat_vc3**: VC3 event
2	0x0	**err_intr_stat_vc2**: VC2 event
1	0x0	**err_intr_stat_vc1**: VC1 event
0	0x0	**err_intr_stat_vc0**: VC0 event

NVCSI_STREAM_4_CORRECTABLE_ERR_INTR_STATUS_NOVC_0

Offset: 0xc0ae
Byte Offset: 0x302b8
Read/Write: R/W

Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**err_intr_stat_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**err_intr_stat_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_4_CORRECTABLE_ERR_INTR_STATUS_VC0_0

Offset: 0xc0af
Byte Offset: 0x302bc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc0**
4	0x0	**err_intr_stat_ph_single_crc_err_vc0**
3	0x0	**err_intr_stat_pd_wc_short_err_vc0**
2	0x0	**err_intr_stat_pd_crc_err_vc0**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc0**
0	0x0	**err_intr_stat_ppfsm_timeout_vc0**

NVCSI_STREAM_4_CORRECTABLE_ERR_INTR_STATUS_VC1_0

Offset: 0xc0b0
Byte Offset: 0x302c0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc1**
4	0x0	**err_intr_stat_ph_single_crc_err_vc1**
3	0x0	**err_intr_stat_pd_wc_short_err_vc1**
2	0x0	**err_intr_stat_pd_crc_err_vc1**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc1**
0	0x0	**err_intr_stat_ppfsm_timeout_vc1**

NVCSI_STREAM_4_CORRECTABLE_ERR_INTR_STATUS_VC2_0

Offset: 0xc0b1
Byte Offset: 0x302c4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc2**
4	0x0	**err_intr_stat_ph_single_crc_err_vc2**
3	0x0	**err_intr_stat_pd_wc_short_err_vc2**
2	0x0	**err_intr_stat_pd_crc_err_vc2**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc2**
0	0x0	**err_intr_stat_ppfsm_timeout_vc2**

NVCSI_STREAM_4_CORRECTABLE_ERR_INTR_STATUS_VC3_0

Offset: 0xc0b2
Byte Offset: 0x302c8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc3**
4	0x0	**err_intr_stat_ph_single_crc_err_vc3**
3	0x0	**err_intr_stat_pd_wc_short_err_vc3**
2	0x0	**err_intr_stat_pd_crc_err_vc3**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc3**
0	0x0	**err_intr_stat_ppfsm_timeout_vc3**

NVCSI_STREAM_4_CORRECTABLE_ERR_INTR_STATUS_VC4_0

Offset: 0xc0b3
Byte Offset: 0x302cc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc4**
4	0x0	**err_intr_stat_ph_single_crc_err_vc4**
3	0x0	**err_intr_stat_pd_wc_short_err_vc4**
2	0x0	**err_intr_stat_pd_crc_err_vc4**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc4**
0	0x0	**err_intr_stat_ppfsm_timeout_vc4**

NVCSI_STREAM_4_CORRECTABLE_ERR_INTR_STATUS_VC5_0

Offset: 0xc0b4
Byte Offset: 0x302d0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc5**
4	0x0	**err_intr_stat_ph_single_crc_err_vc5**
3	0x0	**err_intr_stat_pd_wc_short_err_vc5**
2	0x0	**err_intr_stat_pd_crc_err_vc5**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc5**
0	0x0	**err_intr_stat_ppfsm_timeout_vc5**

NVCSI_STREAM_4_CORRECTABLE_ERR_INTR_STATUS_VC6_0

Offset: 0xc0b5
Byte Offset: 0x302d4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc6**
4	0x0	**err_intr_stat_ph_single_crc_err_vc6**
3	0x0	**err_intr_stat_pd_wc_short_err_vc6**
2	0x0	**err_intr_stat_pd_crc_err_vc6**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc6**
0	0x0	**err_intr_stat_ppfsm_timeout_vc6**

NVCSI_STREAM_4_CORRECTABLE_ERR_INTR_STATUS_VC7_0

Offset: 0xc0b6
Byte Offset: 0x302d8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc7**
4	0x0	**err_intr_stat_ph_single_crc_err_vc7**
3	0x0	**err_intr_stat_pd_wc_short_err_vc7**
2	0x0	**err_intr_stat_pd_crc_err_vc7**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc7**
0	0x0	**err_intr_stat_ppfsm_timeout_vc7**

NVCSI_STREAM_4_CORRECTABLE_ERR_INTR_STATUS_VC8_0

Offset: 0xc0b7
Byte Offset: 0x302dc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc8**
4	0x0	**err_intr_stat_ph_single_crc_err_vc8**
3	0x0	**err_intr_stat_pd_wc_short_err_vc8**
2	0x0	**err_intr_stat_pd_crc_err_vc8**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc8**
0	0x0	**err_intr_stat_ppfsm_timeout_vc8**

NVCSI_STREAM_4_CORRECTABLE_ERR_INTR_STATUS_VC9_0

Offset: 0xc0b8
Byte Offset: 0x302e0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc9**
4	0x0	**err_intr_stat_ph_single_crc_err_vc9**
3	0x0	**err_intr_stat_pd_wc_short_err_vc9**
2	0x0	**err_intr_stat_pd_crc_err_vc9**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc9**
0	0x0	**err_intr_stat_ppfsm_timeout_vc9**

NVCSI_STREAM_4_CORRECTABLE_ERR_INTR_STATUS_VC10_0

Offset: 0xc0b9
Byte Offset: 0x302e4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc10**
4	0x0	**err_intr_stat_ph_single_crc_err_vc10**
3	0x0	**err_intr_stat_pd_wc_short_err_vc10**
2	0x0	**err_intr_stat_pd_crc_err_vc10**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc10**
0	0x0	**err_intr_stat_ppfsm_timeout_vc10**

NVCSI_STREAM_4_CORRECTABLE_ERR_INTR_STATUS_VC11_0

Offset: 0xc0ba
Byte Offset: 0x302e8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc11**
4	0x0	**err_intr_stat_ph_single_crc_err_vc11**
3	0x0	**err_intr_stat_pd_wc_short_err_vc11**
2	0x0	**err_intr_stat_pd_crc_err_vc11**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc11**
0	0x0	**err_intr_stat_ppfsm_timeout_vc11**

NVCSI_STREAM_4_CORRECTABLE_ERR_INTR_STATUS_VC12_0

Offset: 0xc0bb
Byte Offset: 0x302ec
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc12**
4	0x0	**err_intr_stat_ph_single_crc_err_vc12**
3	0x0	**err_intr_stat_pd_wc_short_err_vc12**
2	0x0	**err_intr_stat_pd_crc_err_vc12**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc12**
0	0x0	**err_intr_stat_ppfsm_timeout_vc12**

NVCSI_STREAM_4_CORRECTABLE_ERR_INTR_STATUS_VC13_0

Offset: 0xc0bc
Byte Offset: 0x302f0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc13**
4	0x0	**err_intr_stat_ph_single_crc_err_vc13**
3	0x0	**err_intr_stat_pd_wc_short_err_vc13**
2	0x0	**err_intr_stat_pd_crc_err_vc13**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc13**
0	0x0	**err_intr_stat_ppfsm_timeout_vc13**

NVCSI_STREAM_4_CORRECTABLE_ERR_INTR_STATUS_VC14_0

Offset: 0xc0bd
Byte Offset: 0x302f4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc14**
4	0x0	**err_intr_stat_ph_single_crc_err_vc14**
3	0x0	**err_intr_stat_pd_wc_short_err_vc14**
2	0x0	**err_intr_stat_pd_crc_err_vc14**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc14**
0	0x0	**err_intr_stat_ppfsm_timeout_vc14**

NVCSI_STREAM_4_CORRECTABLE_ERR_INTR_STATUS_VC15_0

Offset: 0xc0be
Byte Offset: 0x302f8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc15**
4	0x0	**err_intr_stat_ph_single_crc_err_vc15**
3	0x0	**err_intr_stat_pd_wc_short_err_vc15**
2	0x0	**err_intr_stat_pd_crc_err_vc15**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc15**
0	0x0	**err_intr_stat_ppfsm_timeout_vc15**

NVCSI_STREAM_4_UNCORRECTABLE_ERR_INTR_STATUS_0

Offset: 0xc0bf
Byte Offset: 0x302fc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**err_intr_stat_novc**: Not VC related interrupt
15	0x0	**err_intr_stat_vc15**: VC15 event
14	0x0	**err_intr_stat_vc14**: VC14 event
13	0x0	**err_intr_stat_vc13**: VC13 event
12	0x0	**err_intr_stat_vc12**: VC12 event
11	0x0	**err_intr_stat_vc11**: VC11 event
10	0x0	**err_intr_stat_vc10**: VC10 event
9	0x0	**err_intr_stat_vc9**: VC9 event
8	0x0	**err_intr_stat_vc8**: VC8 event
7	0x0	**err_intr_stat_vc7**: VC7 event

Bit	Reset	Description
6	0x0	**err_intr_stat_vc6**: VC6 event
5	0x0	**err_intr_stat_vc5**: VC5 event
4	0x0	**err_intr_stat_vc4**: VC4 event
3	0x0	**err_intr_stat_vc3**: VC3 event
2	0x0	**err_intr_stat_vc2**: VC2 event
1	0x0	**err_intr_stat_vc1**: VC1 event
0	0x0	**err_intr_stat_vc0**: VC0 event

NVCSI_STREAM_4_UNCORRECTABLE_ERR_INTR_STATUS_NOVC_0

Offset: 0xc0c0
Byte Offset: 0x30300
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**err_intr_stat_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**err_intr_stat_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_4_UNCORRECTABLE_ERR_INTR_STATUS_VC0_0

Offset: 0xc0c1
Byte Offset: 0x30304
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc0**
4	0x0	**err_intr_stat_ph_single_crc_err_vc0**
3	0x0	**err_intr_stat_pd_wc_short_err_vc0**
2	0x0	**err_intr_stat_pd_crc_err_vc0**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc0**
0	0x0	**err_intr_stat_ppfsm_timeout_vc0**

NVCSI_STREAM_4_UNCORRECTABLE_ERR_INTR_STATUS_VC1_0

Offset: 0xc0c2
Byte Offset: 0x30308
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc1**
4	0x0	**err_intr_stat_ph_single_crc_err_vc1**
3	0x0	**err_intr_stat_pd_wc_short_err_vc1**
2	0x0	**err_intr_stat_pd_crc_err_vc1**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc1**
0	0x0	**err_intr_stat_ppfsm_timeout_vc1**

NVCSI_STREAM_4_UNCORRECTABLE_ERR_INTR_STATUS_VC2_0

Offset: 0xc0c3
Byte Offset: 0x3030c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc2**
4	0x0	**err_intr_stat_ph_single_crc_err_vc2**
3	0x0	**err_intr_stat_pd_wc_short_err_vc2**
2	0x0	**err_intr_stat_pd_crc_err_vc2**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc2**
0	0x0	**err_intr_stat_ppfsm_timeout_vc2**

NVCSI_STREAM_4_UNCORRECTABLE_ERR_INTR_STATUS_VC3_0

Offset: 0xc0c4
Byte Offset: 0x30310
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc3**
4	0x0	**err_intr_stat_ph_single_crc_err_vc3**
3	0x0	**err_intr_stat_pd_wc_short_err_vc3**
2	0x0	**err_intr_stat_pd_crc_err_vc3**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc3**
0	0x0	**err_intr_stat_ppfsm_timeout_vc3**

NVCSI_STREAM_4_UNCORRECTABLE_ERR_INTR_STATUS_VC4_0

Offset: 0xc0c5
Byte Offset: 0x30314
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc4**
4	0x0	**err_intr_stat_ph_single_crc_err_vc4**
3	0x0	**err_intr_stat_pd_wc_short_err_vc4**
2	0x0	**err_intr_stat_pd_crc_err_vc4**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc4**
0	0x0	**err_intr_stat_ppfsm_timeout_vc4**

NVCSI_STREAM_4_UNCORRECTABLE_ERR_INTR_STATUS_VC5_0

Offset: 0xc0c6
Byte Offset: 0x30318
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc5**
4	0x0	**err_intr_stat_ph_single_crc_err_vc5**
3	0x0	**err_intr_stat_pd_wc_short_err_vc5**
2	0x0	**err_intr_stat_pd_crc_err_vc5**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc5**
0	0x0	**err_intr_stat_ppfsm_timeout_vc5**

NVCSI_STREAM_4_UNCORRECTABLE_ERR_INTR_STATUS_VC6_0

Offset: 0xc0c7
Byte Offset: 0x3031c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc6**
4	0x0	**err_intr_stat_ph_single_crc_err_vc6**
3	0x0	**err_intr_stat_pd_wc_short_err_vc6**
2	0x0	**err_intr_stat_pd_crc_err_vc6**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc6**
0	0x0	**err_intr_stat_ppfsm_timeout_vc6**

NVCSI_STREAM_4_UNCORRECTABLE_ERR_INTR_STATUS_VC7_0

Offset: 0xc0c8
Byte Offset: 0x30320
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc7**
4	0x0	**err_intr_stat_ph_single_crc_err_vc7**
3	0x0	**err_intr_stat_pd_wc_short_err_vc7**
2	0x0	**err_intr_stat_pd_crc_err_vc7**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc7**
0	0x0	**err_intr_stat_ppfsm_timeout_vc7**

NVCSI_STREAM_4_UNCORRECTABLE_ERR_INTR_STATUS_VC8_0

Offset: 0xc0c9
Byte Offset: 0x30324
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc8**
4	0x0	**err_intr_stat_ph_single_crc_err_vc8**
3	0x0	**err_intr_stat_pd_wc_short_err_vc8**
2	0x0	**err_intr_stat_pd_crc_err_vc8**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc8**
0	0x0	**err_intr_stat_ppfsm_timeout_vc8**

NVCSI_STREAM_4_UNCORRECTABLE_ERR_INTR_STATUS_VC9_0

Offset: 0xc0ca
Byte Offset: 0x30328
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc9**
4	0x0	**err_intr_stat_ph_single_crc_err_vc9**
3	0x0	**err_intr_stat_pd_wc_short_err_vc9**
2	0x0	**err_intr_stat_pd_crc_err_vc9**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc9**
0	0x0	**err_intr_stat_ppfsm_timeout_vc9**

NVCSI_STREAM_4_UNCORRECTABLE_ERR_INTR_STATUS_VC10_0

Offset: 0xc0cb
Byte Offset: 0x3032c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc10**
4	0x0	**err_intr_stat_ph_single_crc_err_vc10**
3	0x0	**err_intr_stat_pd_wc_short_err_vc10**
2	0x0	**err_intr_stat_pd_crc_err_vc10**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc10**
0	0x0	**err_intr_stat_ppfsm_timeout_vc10**

NVCSI_STREAM_4_UNCORRECTABLE_ERR_INTR_STATUS_VC11_0

Offset: 0xc0cc
Byte Offset: 0x30330
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc11**
4	0x0	**err_intr_stat_ph_single_crc_err_vc11**
3	0x0	**err_intr_stat_pd_wc_short_err_vc11**
2	0x0	**err_intr_stat_pd_crc_err_vc11**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc11**
0	0x0	**err_intr_stat_ppfsm_timeout_vc11**

NVCSI_STREAM_4_UNCORRECTABLE_ERR_INTR_STATUS_VC12_0

Offset: 0xc0cd
Byte Offset: 0x30334
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc12**
4	0x0	**err_intr_stat_ph_single_crc_err_vc12**
3	0x0	**err_intr_stat_pd_wc_short_err_vc12**
2	0x0	**err_intr_stat_pd_crc_err_vc12**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc12**
0	0x0	**err_intr_stat_ppfsm_timeout_vc12**

NVCSI_STREAM_4_UNCORRECTABLE_ERR_INTR_STATUS_VC13_0

Offset: 0xc0ce
Byte Offset: 0x30338
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc13**
4	0x0	**err_intr_stat_ph_single_crc_err_vc13**
3	0x0	**err_intr_stat_pd_wc_short_err_vc13**
2	0x0	**err_intr_stat_pd_crc_err_vc13**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc13**
0	0x0	**err_intr_stat_ppfsm_timeout_vc13**

NVCSI_STREAM_4_UNCORRECTABLE_ERR_INTR_STATUS_VC14_0

Offset: 0xc0cf
Byte Offset: 0x3033c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc14**
4	0x0	**err_intr_stat_ph_single_crc_err_vc14**
3	0x0	**err_intr_stat_pd_wc_short_err_vc14**
2	0x0	**err_intr_stat_pd_crc_err_vc14**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc14**
0	0x0	**err_intr_stat_ppfsm_timeout_vc14**

NVCSI_STREAM_4_UNCORRECTABLE_ERR_INTR_STATUS_VC15_0

Offset: 0xc0d0
Byte Offset: 0x30340
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc15**
4	0x0	**err_intr_stat_ph_single_crc_err_vc15**
3	0x0	**err_intr_stat_pd_wc_short_err_vc15**
2	0x0	**err_intr_stat_pd_crc_err_vc15**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc15**
0	0x0	**err_intr_stat_ppfsm_timeout_vc15**

NVCSI_STREAM_4_ERR_INTR_MASK_NOVC_0

Offset: 0xc0d1
Byte Offset: 0x30344
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**err_intr_mask_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**err_intr_mask_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_4_ERR_INTR_MASK_VC0_0

Offset: 0xc0d2
Byte Offset: 0x30348
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc0**
4	0x0	**err_intr_mask_ph_single_crc_err_vc0**
3	0x0	**err_intr_mask_pd_wc_short_err_vc0**
2	0x0	**err_intr_mask_pd_crc_err_vc0**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc0**
0	0x0	**err_intr_mask_ppfsm_timeout_vc0**

NVCSI_STREAM_4_ERR_INTR_MASK_VC1_0

Offset: 0xc0d3
Byte Offset: 0x3034c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc1**

Bit	Reset	Description
4	0x0	**err_intr_mask_ph_single_crc_err_vc1**
3	0x0	**err_intr_mask_pd_wc_short_err_vc1**
2	0x0	**err_intr_mask_pd_crc_err_vc1**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc1**
0	0x0	**err_intr_mask_ppfsm_timeout_vc1**

NVCSI_STREAM_4_ERR_INTR_MASK_VC2_0

Offset: 0xc0d4
Byte Offset: 0x30350
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc2**
4	0x0	**err_intr_mask_ph_single_crc_err_vc2**
3	0x0	**err_intr_mask_pd_wc_short_err_vc2**
2	0x0	**err_intr_mask_pd_crc_err_vc2**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc2**
0	0x0	**err_intr_mask_ppfsm_timeout_vc2**

NVCSI_STREAM_4_ERR_INTR_MASK_VC3_0

Offset: 0xc0d5
Byte Offset: 0x30354
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc3**
4	0x0	**err_intr_mask_ph_single_crc_err_vc3**
3	0x0	**err_intr_mask_pd_wc_short_err_vc3**
2	0x0	**err_intr_mask_pd_crc_err_vc3**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc3**
0	0x0	**err_intr_mask_ppfsm_timeout_vc3**

NVCSI_STREAM_4_ERR_INTR_MASK_VC4_0

Offset: 0xc0d6
Byte Offset: 0x30358
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc4**
4	0x0	**err_intr_mask_ph_single_crc_err_vc4**
3	0x0	**err_intr_mask_pd_wc_short_err_vc4**
2	0x0	**err_intr_mask_pd_crc_err_vc4**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc4**
0	0x0	**err_intr_mask_ppfsm_timeout_vc4**

NVCSI_STREAM_4_ERR_INTR_MASK_VC5_0

Offset: 0xc0d7
Byte Offset: 0x3035c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc5**
4	0x0	**err_intr_mask_ph_single_crc_err_vc5**
3	0x0	**err_intr_mask_pd_wc_short_err_vc5**
2	0x0	**err_intr_mask_pd_crc_err_vc5**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc5**
0	0x0	**err_intr_mask_ppfsm_timeout_vc5**

NVCSI_STREAM_4_ERR_INTR_MASK_VC6_0

Offset: 0xc0d8
Byte Offset: 0x30360
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc6**
4	0x0	**err_intr_mask_ph_single_crc_err_vc6**
3	0x0	**err_intr_mask_pd_wc_short_err_vc6**
2	0x0	**err_intr_mask_pd_crc_err_vc6**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc6**
0	0x0	**err_intr_mask_ppfsm_timeout_vc6**

NVCSI_STREAM_4_ERR_INTR_MASK_VC7_0

Offset: 0xc0d9
Byte Offset: 0x30364
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc7**
4	0x0	**err_intr_mask_ph_single_crc_err_vc7**
3	0x0	**err_intr_mask_pd_wc_short_err_vc7**
2	0x0	**err_intr_mask_pd_crc_err_vc7**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc7**
0	0x0	**err_intr_mask_ppfsm_timeout_vc7**

NVCSI_STREAM_4_ERR_INTR_MASK_VC8_0

Offset: 0xc0da
Byte Offset: 0x30368
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc8**
4	0x0	**err_intr_mask_ph_single_crc_err_vc8**
3	0x0	**err_intr_mask_pd_wc_short_err_vc8**
2	0x0	**err_intr_mask_pd_crc_err_vc8**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc8**
0	0x0	**err_intr_mask_ppfsm_timeout_vc8**

NVCSI_STREAM_4_ERR_INTR_MASK_VC9_0

Offset: 0xc0db
Byte Offset: 0x3036c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc9**
4	0x0	**err_intr_mask_ph_single_crc_err_vc9**
3	0x0	**err_intr_mask_pd_wc_short_err_vc9**
2	0x0	**err_intr_mask_pd_crc_err_vc9**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc9**
0	0x0	**err_intr_mask_ppfsm_timeout_vc9**

NVCSI_STREAM_4_ERR_INTR_MASK_VC10_0

Offset: 0xc0dc
Byte Offset: 0x30370
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc10**
4	0x0	**err_intr_mask_ph_single_crc_err_vc10**
3	0x0	**err_intr_mask_pd_wc_short_err_vc10**
2	0x0	**err_intr_mask_pd_crc_err_vc10**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc10**
0	0x0	**err_intr_mask_ppfsm_timeout_vc10**

NVCSI_STREAM_4_ERR_INTR_MASK_VC11_0

Offset: 0xc0dd
Byte Offset: 0x30374
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc11**
4	0x0	**err_intr_mask_ph_single_crc_err_vc11**
3	0x0	**err_intr_mask_pd_wc_short_err_vc11**
2	0x0	**err_intr_mask_pd_crc_err_vc11**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc11**
0	0x0	**err_intr_mask_ppfsm_timeout_vc11**

NVCSI_STREAM_4_ERR_INTR_MASK_VC12_0

Offset: 0xc0de
Byte Offset: 0x30378
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc12**
4	0x0	**err_intr_mask_ph_single_crc_err_vc12**
3	0x0	**err_intr_mask_pd_wc_short_err_vc12**
2	0x0	**err_intr_mask_pd_crc_err_vc12**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc12**
0	0x0	**err_intr_mask_ppfsm_timeout_vc12**

NVCSI_STREAM_4_ERR_INTR_MASK_VC13_0

Offset: 0xc0df
Byte Offset: 0x3037c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc13**
4	0x0	**err_intr_mask_ph_single_crc_err_vc13**
3	0x0	**err_intr_mask_pd_wc_short_err_vc13**
2	0x0	**err_intr_mask_pd_crc_err_vc13**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc13**
0	0x0	**err_intr_mask_ppfsm_timeout_vc13**

NVCSI_STREAM_4_ERR_INTR_MASK_VC14_0

Offset: 0xc0e0
Byte Offset: 0x30380
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc14**
4	0x0	**err_intr_mask_ph_single_crc_err_vc14**
3	0x0	**err_intr_mask_pd_wc_short_err_vc14**
2	0x0	**err_intr_mask_pd_crc_err_vc14**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc14**
0	0x0	**err_intr_mask_ppfsm_timeout_vc14**

NVCSI_STREAM_4_ERR_INTR_MASK_VC15_0

Offset: 0xc0e1
Byte Offset: 0x30384
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc15**
4	0x0	**err_intr_mask_ph_single_crc_err_vc15**
3	0x0	**err_intr_mask_pd_wc_short_err_vc15**
2	0x0	**err_intr_mask_pd_crc_err_vc15**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc15**
0	0x0	**err_intr_mask_ppfsm_timeout_vc15**

NVCSI_STREAM_4_ERR_INTR_TYPE_NOVC_0

Offset: 0xc0e2
Byte Offset: 0x30388
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**err_intr_type_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**err_intr_type_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_4_ERR_INTR_TYPE_VC0_0

Offset: 0xc0e3
Byte Offset: 0x3038c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc0**
4	0x0	**err_intr_type_ph_single_crc_err_vc0**

Bit	Reset	Description
3	0x0	**err_intr_type_pd_wc_short_err_vc0**
2	0x0	**err_intr_type_pd_crc_err_vc0**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc0**
0	0x0	**err_intr_type_ppfsm_timeout_vc0**

NVCSI_STREAM_4_ERR_INTR_TYPE_VC1_0

Offset: 0xc0e4
Byte Offset: 0x30390
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc1**
4	0x0	**err_intr_type_ph_single_crc_err_vc1**
3	0x0	**err_intr_type_pd_wc_short_err_vc1**
2	0x0	**err_intr_type_pd_crc_err_vc1**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc1**
0	0x0	**err_intr_type_ppfsm_timeout_vc1**

NVCSI_STREAM_4_ERR_INTR_TYPE_VC2_0

Offset: 0xc0e5
Byte Offset: 0x30394
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc2**

Bit	Reset	Description
4	0x0	**err_intr_type_ph_single_crc_err_vc2**
3	0x0	**err_intr_type_pd_wc_short_err_vc2**
2	0x0	**err_intr_type_pd_crc_err_vc2**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc2**
0	0x0	**err_intr_type_ppfsm_timeout_vc2**

NVCSI_STREAM_4_ERR_INTR_TYPE_VC3_0

Offset: 0xc0e6
Byte Offset: 0x30398
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc3**
4	0x0	**err_intr_type_ph_single_crc_err_vc3**
3	0x0	**err_intr_type_pd_wc_short_err_vc3**
2	0x0	**err_intr_type_pd_crc_err_vc3**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc3**
0	0x0	**err_intr_type_ppfsm_timeout_vc3**

NVCSI_STREAM_4_ERR_INTR_TYPE_VC4_0

Offset: 0xc0e7
Byte Offset: 0x3039c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc4**
4	0x0	**err_intr_type_ph_single_crc_err_vc4**
3	0x0	**err_intr_type_pd_wc_short_err_vc4**
2	0x0	**err_intr_type_pd_crc_err_vc4**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc4**
0	0x0	**err_intr_type_ppfsm_timeout_vc4**

NVCSI_STREAM_4_ERR_INTR_TYPE_VC5_0

Offset: 0xc0e8
Byte Offset: 0x303a0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc5**
4	0x0	**err_intr_type_ph_single_crc_err_vc5**
3	0x0	**err_intr_type_pd_wc_short_err_vc5**
2	0x0	**err_intr_type_pd_crc_err_vc5**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc5**
0	0x0	**err_intr_type_ppfsm_timeout_vc5**

NVCSI_STREAM_4_ERR_INTR_TYPE_VC6_0

Offset: 0xc0e9
Byte Offset: 0x303a4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc6**
4	0x0	**err_intr_type_ph_single_crc_err_vc6**
3	0x0	**err_intr_type_pd_wc_short_err_vc6**
2	0x0	**err_intr_type_pd_crc_err_vc6**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc6**
0	0x0	**err_intr_type_ppfsm_timeout_vc6**

NVCSI_STREAM_4_ERR_INTR_TYPE_VC7_0

Offset: 0xc0ea
Byte Offset: 0x303a8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc7**
4	0x0	**err_intr_type_ph_single_crc_err_vc7**
3	0x0	**err_intr_type_pd_wc_short_err_vc7**
2	0x0	**err_intr_type_pd_crc_err_vc7**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc7**
0	0x0	**err_intr_type_ppfsm_timeout_vc7**

NVCSI_STREAM_4_ERR_INTR_TYPE_VC8_0

Offset: 0xc0eb
Byte Offset: 0x303ac
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc8**
4	0x0	**err_intr_type_ph_single_crc_err_vc8**
3	0x0	**err_intr_type_pd_wc_short_err_vc8**
2	0x0	**err_intr_type_pd_crc_err_vc8**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc8**
0	0x0	**err_intr_type_ppfsm_timeout_vc8**

NVCSI_STREAM_4_ERR_INTR_TYPE_VC9_0

Offset: 0xc0ec
Byte Offset: 0x303b0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc9**
4	0x0	**err_intr_type_ph_single_crc_err_vc9**
3	0x0	**err_intr_type_pd_wc_short_err_vc9**
2	0x0	**err_intr_type_pd_crc_err_vc9**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc9**
0	0x0	**err_intr_type_ppfsm_timeout_vc9**

NVCSI_STREAM_4_ERR_INTR_TYPE_VC10_0

Offset: 0xc0ed
Byte Offset: 0x303b4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc10**
4	0x0	**err_intr_type_ph_single_crc_err_vc10**
3	0x0	**err_intr_type_pd_wc_short_err_vc10**
2	0x0	**err_intr_type_pd_crc_err_vc10**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc10**
0	0x0	**err_intr_type_ppfsm_timeout_vc10**

NVCSI_STREAM_4_ERR_INTR_TYPE_VC11_0

Offset: 0xc0ee
Byte Offset: 0x303b8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc11**
4	0x0	**err_intr_type_ph_single_crc_err_vc11**
3	0x0	**err_intr_type_pd_wc_short_err_vc11**
2	0x0	**err_intr_type_pd_crc_err_vc11**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc11**
0	0x0	**err_intr_type_ppfsm_timeout_vc11**

NVCSI_STREAM_4_ERR_INTR_TYPE_VC12_0

Offset: 0xc0ef
Byte Offset: 0x303bc
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc12**
4	0x0	**err_intr_type_ph_single_crc_err_vc12**
3	0x0	**err_intr_type_pd_wc_short_err_vc12**
2	0x0	**err_intr_type_pd_crc_err_vc12**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc12**
0	0x0	**err_intr_type_ppfsm_timeout_vc12**

NVCSI_STREAM_4_ERR_INTR_TYPE_VC13_0

Offset: 0xc0f0
Byte Offset: 0x303c0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc13**
4	0x0	**err_intr_type_ph_single_crc_err_vc13**
3	0x0	**err_intr_type_pd_wc_short_err_vc13**
2	0x0	**err_intr_type_pd_crc_err_vc13**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc13**
0	0x0	**err_intr_type_ppfsm_timeout_vc13**

NVCSI_STREAM_4_ERR_INTR_TYPE_VC14_0

Offset: 0xc0f1
Byte Offset: 0x303c4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc14**
4	0x0	**err_intr_type_ph_single_crc_err_vc14**
3	0x0	**err_intr_type_pd_wc_short_err_vc14**
2	0x0	**err_intr_type_pd_crc_err_vc14**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc14**
0	0x0	**err_intr_type_ppfsm_timeout_vc14**

NVCSI_STREAM_4_ERR_INTR_TYPE_VC15_0

Offset: 0xc0f2
Byte Offset: 0x303c8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc15**
4	0x0	**err_intr_type_ph_single_crc_err_vc15**
3	0x0	**err_intr_type_pd_wc_short_err_vc15**
2	0x0	**err_intr_type_pd_crc_err_vc15**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc15**
0	0x0	**err_intr_type_ppfsm_timeout_vc15**

NVCSI_STREAM_4_TPG_ENABLE_0

Offset: 0xc0f3
Byte Offset: 0x303cc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	DISABLE	**TPG_ENABLE**: Enable the TPG path **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_4_TPG_VC_ENABLE_0

Offset: 0xc0f4
Byte Offset: 0x303d0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15	DISABLE	**VC15_EN**: Enable bit for the VC15 **0** = DISABLE **1** = ENABLE
14	DISABLE	**VC14_EN**: Enable bit for the VC14 **0** = DISABLE **1** = ENABLE
13	DISABLE	**VC13_EN**: Enable bit for the VC13 **0** = DISABLE **1** = ENABLE
12	DISABLE	**VC12_EN**: Enable bit for the VC12 **0** = DISABLE **1** = ENABLE
11	DISABLE	**VC11_EN**: Enable bit for the VC11 **0** = DISABLE **1** = ENABLE
10	DISABLE	**VC10_EN**: Enable bit for the VC10 **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
9	DISABLE	**VC9_EN**: Enable bit for the VC9 **0** = DISABLE **1** = ENABLE
8	DISABLE	**VC8_EN**: Enable bit for the VC8 **0** = DISABLE **1** = ENABLE
7	DISABLE	**VC7_EN**: Enable bit for the VC7 **0** = DISABLE **1** = ENABLE
6	DISABLE	**VC6_EN**: Enable bit for the VC6 **0** = DISABLE **1** = ENABLE
5	DISABLE	**VC5_EN**: Enable bit for the VC5 **0** = DISABLE **1** = ENABLE
4	DISABLE	**VC4_EN**: Enable bit for the VC4 **0** = DISABLE **1** = ENABLE
3	DISABLE	**VC3_EN**: Enable bit for the VC3 **0** = DISABLE **1** = ENABLE
2	DISABLE	**VC2_EN**: Enable bit for the VC2 **0** = DISABLE **1** = ENABLE
1	DISABLE	**VC1_EN**: Enable bit for the VC1 **0** = DISABLE **1** = ENABLE
0	DISABLE	**VC0_EN**: Enable bit for the VC0 **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_4_LOAD_TPG_CFG_0

Offset: 0xc0f5
Byte Offset: 0x303d4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**LOAD**: Load the shadow register

NVCSI_STREAM_4_TPG_CTRL_0

Offset: 0xc0f6
Byte Offset: 0x303d8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000008a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1000,1010)

Bit	Reset	Description
7:4	0x8	**DATA_SPEED**: Control the data generation speed, valid range is 1~8.
3	ENABLE	**SKIP_LS_LE_PKT**: If the LS/LE packet need to generated. **0** = DISABLE **1** = ENABLE
2	DISABLE	**OVERRIDE_CRC**: Override the packet header CRC and payload CRC. **0** = DISABLE **1** = ENABLE
1	CORE	**DEST**: The TPG pattern is send to PP or send to CIL for TX. **0** = CIL **1** = CORE

Bit	Reset	Description
0	DPHY	**PHY_MODE**: CPHY or DPHY packet structure for TPG **0** = DPHY **1** = CPHY

NVCSI_STREAM_4_TPG_VBLANK_0
Offset: 0xc0f7
Byte Offset: 0x303dc
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000800 (0b0000,0000,0000,0000,0000,1000,0000,0000)

Bit	Reset	Description
31:0	0x800	**VBLANK**: The vblank value, unit is nvcsi clock cycles.

NVCSI_STREAM_4_TPG_HBLANK_0
Offset: 0xc0f8
Byte Offset: 0x303e0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000800 (0b0000,0000,0000,0000,0000,1000,0000,0000)

Bit	Reset	Description
31:0	0x800	**HBLANK**: The hblank value, unit is nvcsi clock cycles.

NVCSI_STREAM_4_TPG_STATUS_0
Offset: 0xc0f9
Byte Offset: 0x303e4
Read/Write: RO
Parity Protection: N
Shadow: N

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0001ffff (0bxxxx,xxxx,xxxx,xxx1,1111,1111,1111,1111)

Bit	Reset	Description
16	IDLE	**VC15_STATUS**: Indicate the VC15 status **0** = BUSY **1** = IDLE
15	IDLE	**VC14_STATUS**: Indicate the VC14 status **0** = BUSY **1** = IDLE
14	IDLE	**VC13_STATUS**: Indicate the VC13 status **0** = BUSY **1** = IDLE
13	IDLE	**VC12_STATUS**: Indicate the VC12 status **0** = BUSY **1** = IDLE
12	IDLE	**VC11_STATUS**: Indicate the VC11 status **0** = BUSY **1** = IDLE
11	IDLE	**VC10_STATUS**: Indicate the VC10 status **0** = BUSY **1** = IDLE
10	IDLE	**VC9_STATUS**: Indicate the VC9 status **0** = BUSY **1** = IDLE
9	IDLE	**VC8_STATUS**: Indicate the VC8 status **0** = BUSY **1** = IDLE
8	IDLE	**VC7_STATUS**: Indicate the VC7 status **0** = BUSY **1** = IDLE

Bit	Reset	Description
7	IDLE	**VC6_STATUS**: Indicate the VC6 status **0** = BUSY **1** = IDLE
6	IDLE	**VC5_STATUS**: Indicate the VC5 status **0** = BUSY **1** = IDLE
5	IDLE	**VC4_STATUS**: Indicate the VC4 status **0** = BUSY **1** = IDLE
4	IDLE	**VC3_STATUS**: Indicate the VC3 status **0** = BUSY **1** = IDLE
3	IDLE	**VC2_STATUS**: Indicate the VC2 status **0** = BUSY **1** = IDLE
2	IDLE	**VC1_STATUS**: Indicate the VC1 status **0** = BUSY **1** = IDLE
1	IDLE	**VC0_STATUS**: Indicate the VC0 status **0** = BUSY **1** = IDLE
0	IDLE	**STATUS**: Indicate the TPG is in idle state, all packet has been send and all VC are disabled. **0** = BUSY **1** = IDLE

NVCSI_STREAM_4_TPG_PH_ECC_0

Offset: 0xc0fa
Byte Offset: 0x303e8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:24	0x0	**LINE_ECC**: This field is for the long packet header ECC.
23:18	0x0	**EOL_ECC**: This field is for the EOL short packet ECC.
17:12	0x0	**SOL_ECC**: This field is for the SOL short packet ECC.
11:6	0x0	**EOF_ECC**: This field is for the EOF short packet ECC.
5:0	0x0	**SOF_ECC**: The TPG will not generate ECC for a packet. When using the TPG, SW should set the PP to skip the ecc check. To verify the ecc logic for safety BIST, SW can write a pre-calculated ECC for the TPG, when use with this mode, the TPG should generate a grescale pattern. This field is for the SOF short packet ECC.

NVCSI_STREAM_4_TPG_PF_CRC_0

Offset: 0xc0fb
Byte Offset: 0x303ec
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PF_CRC**: This field is for the long packet payload CRC override.

NVCSI_STREAM_4_TPG_PH_SOF_CRC_0

Offset: 0xc0fc
Byte Offset: 0x303f0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This field is for the CPHY SOF packet second packet header CRC override.
15:0	0x0	**CRC_0**: This field is for the CPHY SOF packet first packet header CRC override.

NVCSI_STREAM_4_TPG_PH_EOF_CRC_0

Offset: 0xc0fd
Byte Offset: 0x303f4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This field is for the CPHY EOF packet second packet header CRC override.
15:0	0x0	**CRC_0**: This field is for the CPHY EOF packet first packet header CRC override.

NVCSI_STREAM_4_TPG_PH_SOL_CRC_0

Offset: 0xc0fe
Byte Offset: 0x303f8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This field is for the CPHY SOL packet second packet header CRC override.
15:0	0x0	**CRC_0**: This field is for the CPHY SOL packet first packet header CRC override.

NVCSI_STREAM_4_TPG_PH_EOL_CRC_0

Offset: 0xc0ff
Byte Offset: 0x303fc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This field is for the CPHY EOL packet second packet header CRC override.
15:0	0x0	**CRC_0**: This field is for the CPHY EOL packet first packet header CRC override.

NVCSI_STREAM_4_TPG_PH_LONG_PKT_CRC_0

Offset: 0xc100
Byte Offset: 0x30400
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This filed is for the CPHY long packet second packet header CRC override.
15:0	0x0	**CRC_0**: This filed is for the CPHY long packet first packet header CRC override.

NVCSI_STREAM_4_TPG_PKT_DELIMETER_0

Offset: 0xc101
Byte Offset: 0x30404
Read/Write: R/W
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0011)

Bit	Reset	Description
15:0	0x3	**NUM**: This field define the cycle number between two packet. Model the LP11 period between two packet, valid range is 1 to 65535.

NVCSI_STREAM_4_VC0_TPG_GAIN_CTRL_0

Offset: 0xc102
Byte Offset: 0x30408
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_4_VC1_TPG_GAIN_CTRL_0

Offset: 0xc103
Byte Offset: 0x3040c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_4_VC2_TPG_GAIN_CTRL_0

Offset: 0xc104
Byte Offset: 0x30410
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_4_VC3_TPG_GAIN_CTRL_0

Offset: 0xc105
Byte Offset: 0x30414
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_4_VC4_TPG_GAIN_CTRL_0

Offset: 0xc106
Byte Offset: 0x30418
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_4_VC5_TPG_GAIN_CTRL_0

Offset: 0xc107
Byte Offset: 0x3041c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_4_VC6_TPG_GAIN_CTRL_0

Offset: 0xc108
Byte Offset: 0x30420
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_4_VC7_TPG_GAIN_CTRL_0

Offset: 0xc109
Byte Offset: 0x30424
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_4_VC8_TPG_GAIN_CTRL_0

Offset: 0xc10a
Byte Offset: 0x30428
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_4_VC9_TPG_GAIN_CTRL_0

Offset: 0xc10b
Byte Offset: 0x3042c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_4_VC10_TPG_GAIN_CTRL_0

Offset: 0xc10c
Byte Offset: 0x30430
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_4_VC11_TPG_GAIN_CTRL_0

Offset: 0xc10d
Byte Offset: 0x30434
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_4_VC12_TPG_GAIN_CTRL_0

Offset: 0xc10e
Byte Offset: 0x30438
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_4_VC13_TPG_GAIN_CTRL_0

Offset: 0xc10f
Byte Offset: 0x3043c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_4_VC14_TPG_GAIN_CTRL_0

Offset: 0xc110
Byte Offset: 0x30440
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_4_VC15_TPG_GAIN_CTRL_0

Offset: 0xc111
Byte Offset: 0x30444
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_4_SPARE_0
Offset: 0xc112
Byte Offset: 0x30448
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**stream_reg**

7.2.1.3.6 NVCSI Stream5 Registers

NVCSI_STREAM_5_SW_RESET_CTRL_0

Offset: 0xe000
Byte Offset: 0x38000
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DISABLE	**cfg_swreset**: Reset the pixel parser. **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_5_SLCG_CTRL_0

Offset: 0xe001
Byte Offset: 0x38004
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DISABLE	**cfg_slcg_override**: Enable the SLCG override for the pixel parser. **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_5_PP_PHY_CTRL_0

Offset: 0xe002
Byte Offset: 0x38008
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DPHY	**cfg_phy_mode**: **0** = DPHY **1** = CPHY

NVCSI_STREAM_5_PP_EN_CTRL_0

Offset: 0xe003
Byte Offset: 0x3800c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	CIL	**cfg_src**: The pixel source of the pixel parser **0** = CIL **1** = TPG
0	DISABLE	**cfg_pp_en**: Pixel Parser streaming enable **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_5_VC0_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xe004
Byte Offset: 0x38010
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc0_dt_nooverride_0**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC0_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xe005
Byte Offset: 0x38014
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc0_dt_nooverride_1**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC0_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xe006
Byte Offset: 0x38018
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc0_dt_nooverride_2**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC0_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xe007
Byte Offset: 0x3801c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc0_dt_nooverride_3**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC0_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xe008
Byte Offset: 0x38020
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc0_dt_nooverride_4**: When incoming PH_DTYPE in VC0 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC0_DT_OVERRIDE_0

Offset: 0xe009
Byte Offset: 0x38024
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc0_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc0_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc0_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_5_VC1_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xe00a
Byte Offset: 0x38028
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc1_dt_nooverride_0**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC1_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xe00b
Byte Offset: 0x3802c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc1_dt_nooverride_1**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC1_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xe00c
Byte Offset: 0x38030
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc1_dt_nooverride_2**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC1_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xe00d
Byte Offset: 0x38034
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc1_dt_nooverride_3**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC1_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xe00e
Byte Offset: 0x38038
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc1_dt_nooverride_4**: When incoming PH_DTYPE in VC1 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC1_DT_OVERRIDE_0

Offset: 0xe00f
Byte Offset: 0x3803c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc1_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc1_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc1_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_5_VC2_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xe010
Byte Offset: 0x38040
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc2_dt_nooverride_0**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC2_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xe011
Byte Offset: 0x38044
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc2_dt_nooverride_1**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC2_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xe012
Byte Offset: 0x38048
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc2_dt_nooverride_2**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC2_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xe013
Byte Offset: 0x3804c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc2_dt_nooverride_3**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC2_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xe014
Byte Offset: 0x38050
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc2_dt_nooverride_4**: When incoming PH_DTYPE in VC2 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC2_DT_OVERRIDE_0

Offset: 0xe015
Byte Offset: 0x38054
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc2_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc2_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc2_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_5_VC3_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xe016
Byte Offset: 0x38058
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc3_dt_nooverride_0**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC3_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xe017
Byte Offset: 0x3805c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc3_dt_nooverride_1**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC3_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xe018
Byte Offset: 0x38060
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc3_dt_nooverride_2**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC3_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xe019
Byte Offset: 0x38064
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc3_dt_nooverride_3**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC3_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xe01a
Byte Offset: 0x38068
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc3_dt_nooverride_4**: When incoming PH_DTYPE in VC3 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC3_DT_OVERRIDE_0

Offset: 0xe01b
Byte Offset: 0x3806c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc3_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc3_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc3_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_5_VC4_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xe01c
Byte Offset: 0x38070
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc4_dt_nooverride_0**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC4_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xe01d
Byte Offset: 0x38074
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc4_dt_nooverride_1**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC4_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xe01e
Byte Offset: 0x38078
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc4_dt_nooverride_2**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC4_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xe01f
Byte Offset: 0x3807c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc4_dt_nooverride_3**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC4_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xe020
Byte Offset: 0x38080
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc4_dt_nooverride_4**: When incoming PH_DTYPE in VC4 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC4_DT_OVERRIDE_0

Offset: 0xe021
Byte Offset: 0x38084
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc4_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc4_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc4_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_5_VC5_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xe022
Byte Offset: 0x38088
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc5_dt_nooverride_0**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC5_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xe023
Byte Offset: 0x3808c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc5_dt_nooverride_1**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC5_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xe024
Byte Offset: 0x38090
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc5_dt_nooverride_2**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC5_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xe025
Byte Offset: 0x38094
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc5_dt_nooverride_3**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC5_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xe026
Byte Offset: 0x38098
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc5_dt_nooverride_4**: When incoming PH_DTYPE in VC5 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC5_DT_OVERRIDE_0

Offset: 0xe027
Byte Offset: 0x3809c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc5_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc5_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc5_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_5_VC6_DT_NOOVERRIDE_CTRL_0_0

VC6

Offset: 0xe028
Byte Offset: 0x380a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc6_dt_nooverride_0**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC6_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xe029
Byte Offset: 0x380a4
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc6_dt_nooverride_1**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC6_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xe02a
Byte Offset: 0x380a8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc6_dt_nooverride_2**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC6_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xe02b
Byte Offset: 0x380ac
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc6_dt_nooverride_3**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC6_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xe02c
Byte Offset: 0x380b0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc6_dt_nooverride_4**: When incoming PH_DTYPE in VC6 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC6_DT_OVERRIDE_0

Offset: 0xe02d
Byte Offset: 0x380b4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc6_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc6_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc6_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_5_VC7_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xe02e
Byte Offset: 0x380b8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc7_dt_nooverride_0**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC7_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xe02f
Byte Offset: 0x380bc
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc7_dt_nooverride_1**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC7_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xe030
Byte Offset: 0x380c0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc7_dt_nooverride_2**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC7_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xe031
Byte Offset: 0x380c4
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc7_dt_nooverride_3**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC7_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xe032
Byte Offset: 0x380c8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc7_dt_nooverride_4**: When incoming PH_DTYPE in VC7 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC7_DT_OVERRIDE_0

Offset: 0xe033
Byte Offset: 0x380cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc7_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc7_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc7_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_5_VC8_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xe034
Byte Offset: 0x380d0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc8_dt_nooverride_0**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC8_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xe035
Byte Offset: 0x380d4
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc8_dt_nooverride_1**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC8_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xe036
Byte Offset: 0x380d8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc8_dt_nooverride_2**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC8_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xe037
Byte Offset: 0x380dc
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc8_dt_nooverride_3**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC8_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xe038
Byte Offset: 0x380e0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc8_dt_nooverride_4**: When incoming PH_DTYPE in VC8 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC8_DT_OVERRIDE_0

Offset: 0xe039
Byte Offset: 0x380e4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc8_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc8_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc8_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_5_VC9_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xe03a
Byte Offset: 0x380e8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc9_dt_nooverride_0**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC9_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xe03b
Byte Offset: 0x380ec
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc9_dt_nooverride_1**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC9_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xe03c
Byte Offset: 0x380f0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc9_dt_nooverride_2**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC9_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xe03d
Byte Offset: 0x380f4
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc9_dt_nooverride_3**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC9_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xe03e
Byte Offset: 0x380f8
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc9_dt_nooverride_4**: When incoming PH_DTYPE in VC9 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC9_DT_OVERRIDE_0

Offset: 0xe03f
Byte Offset: 0x380fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc9_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc9_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc9_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_5_VC10_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xe040
Byte Offset: 0x38100
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc10_dt_nooverride_0**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC10_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xe041
Byte Offset: 0x38104
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc10_dt_nooverride_1**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC10_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xe042
Byte Offset: 0x38108
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc10_dt_nooverride_2**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC10_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xe043
Byte Offset: 0x3810c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc10_dt_nooverride_3**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC10_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xe044
Byte Offset: 0x38110
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc10_dt_nooverride_4**: When incoming PH_DTYPE in VC10 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC10_DT_OVERRIDE_0

Offset: 0xe045
Byte Offset: 0x38114
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc10_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc10_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc10_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_5_VC11_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xe046
Byte Offset: 0x38118
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc11_dt_nooverride_0**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC11_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xe047
Byte Offset: 0x3811c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc11_dt_nooverride_1**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC11_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xe048
Byte Offset: 0x38120
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc11_dt_nooverride_2**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC11_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xe049
Byte Offset: 0x38124
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc11_dt_nooverride_3**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC11_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xe04a
Byte Offset: 0x38128
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc11_dt_nooverride_4**: When incoming PH_DTYPE in VC11 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC11_DT_OVERRIDE_0

Offset: 0xe04b
Byte Offset: 0x3812c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc11_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc11_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc11_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_5_VC12_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xe04c
Byte Offset: 0x38130
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc12_dt_nooverride_0**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC12_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xe04d
Byte Offset: 0x38134
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc12_dt_nooverride_1**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC12_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xe04e
Byte Offset: 0x38138
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc12_dt_nooverride_2**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC12_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xe04f
Byte Offset: 0x3813c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc12_dt_nooverride_3**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC12_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xe050
Byte Offset: 0x38140
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc12_dt_nooverride_4**: When incoming PH_DTYPE in VC12 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC12_DT_OVERRIDE_0

Offset: 0xe051
Byte Offset: 0x38144
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc12_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc12_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc12_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_5_VC13_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xe052
Byte Offset: 0x38148
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc13_dt_nooverride_0**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC13_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xe053
Byte Offset: 0x3814c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc13_dt_nooverride_1**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC13_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xe054
Byte Offset: 0x38150
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc13_dt_nooverride_2**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC13_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xe055
Byte Offset: 0x38154
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc13_dt_nooverride_3**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC13_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xe056
Byte Offset: 0x38158
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc13_dt_nooverride_4**: When incoming PH_DTYPE in VC13 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC13_DT_OVERRIDE_0

Offset: 0xe057
Byte Offset: 0x3815c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc13_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc13_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc13_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_5_VC14_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xe058
Byte Offset: 0x38160
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc14_dt_nooverride_0**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC14_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xe059
Byte Offset: 0x38164
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc14_dt_nooverride_1**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC14_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xe05a
Byte Offset: 0x38168
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc14_dt_nooverride_2**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC14_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xe05b
Byte Offset: 0x3816c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc14_dt_nooverride_3**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC14_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xe05c
Byte Offset: 0x38170
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc14_dt_nooverride_4**: When incoming PH_DTYPE in VC14 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC14_DT_OVERRIDE_0

Offset: 0xe05d
Byte Offset: 0x38174
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc14_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc14_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc14_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_5_VC15_DT_NOOVERRIDE_CTRL_0_0

Offset: 0xe05e
Byte Offset: 0x38178
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:0	0x12	**cfg_vc15_dt_nooverride_0**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC15_DT_NOOVERRIDE_CTRL_1_0

Offset: 0xe05f
Byte Offset: 0x3817c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000028 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1000)

Bit	Reset	Description
5:0	0x28	**cfg_vc15_dt_nooverride_1**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC15_DT_NOOVERRIDE_CTRL_2_0

Offset: 0xe060
Byte Offset: 0x38180
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000029 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1001)

Bit	Reset	Description
5:0	0x29	**cfg_vc15_dt_nooverride_2**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC15_DT_NOOVERRIDE_CTRL_3_0

Offset: 0xe061
Byte Offset: 0x38184
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
5:0	0x2a	**cfg_vc15_dt_nooverride_3**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC15_DT_NOOVERRIDE_CTRL_4_0

Offset: 0xe062
Byte Offset: 0x38188
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002b (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1011)

Bit	Reset	Description
5:0	0x2b	**cfg_vc15_dt_nooverride_4**: When incoming PH_DTYPE in VC15 match with this programmed value, then DT override is disabled

NVCSI_STREAM_5_VC15_DT_OVERRIDE_0

Offset: 0xe063
Byte Offset: 0x3818c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000002a (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,1010)

Bit	Reset	Description
31	DISABLE	**cfg_vc15_dt_override_en**: Override enable for image DT. **0** = DISABLE **1** = ENABLE
5:0	0x2a	**cfg_vc15_dt_override**: This value is used by Color Parser to arrgange the pixels to VI,when OVERRIDE_DT_EN[31] = '1'. If cfg_vc15_dt_override_en = 0, then Color Parser use the DT form PH

NVCSI_STREAM_5_PPFSM_TIMEOUT_CTRL_0

Offset: 0xe064
Byte Offset: 0x38190
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x7fffffff (0b0111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31	DISABLE	**cfg_timeout_en**: Enable Timeout counter for the PP FSM **0** = DISABLE **1** = ENABLE
30:0	0x7fffffff	**cfg_timeout_period**: Timeout period

NVCSI_STREAM_5_PH_CHK_CTRL_0

Offset: 0xe065
Byte Offset: 0x38194
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	ENABLE	**cfg_ph_16_vc**: 16 VC support **0** = DISABLE **1** = ENABLE
1	ENABLE	**cfg_ph_crc_chk_en**: PH CRC check enable (only for CPHY case), when this bit is set to 0, the packet header will still be decode when the CRC check fail, but the error will be set. **0** = DISABLE **1** = ENABLE
0	ENABLE	**cfg_ph_ecc_chk_en**: PH ECC check enable (only for DPHY case), when this bit is set to 0, the packet header will still be decode when the ECC check fail, but the error will be set. **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_5_VC0_DPCM_CTRL_0

Offset: 0xe066
Byte Offset: 0x38198
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc0_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_5_VC1_DPCM_CTRL_0

Offset: 0xe067
Byte Offset: 0x3819c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc1_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_5_VC2_DPCM_CTRL_0

Offset: 0xe068
Byte Offset: 0x381a0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc2_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_5_VC3_DPCM_CTRL_0

Offset: 0xe069
Byte Offset: 0x381a4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc3_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_5_VC4_DPCM_CTRL_0

Offset: 0xe06a
Byte Offset: 0x381a8
Read/Write: R/W
Parity Protection: Y
Shadow: N

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc4_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_5_VC5_DPCM_CTRL_0

Offset: 0xe06b
Byte Offset: 0x381ac
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc5_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_5_VC6_DPCM_CTRL_0

Offset: 0xe06c
Byte Offset: 0x381b0
Read/Write: R/W
Parity Protection: Y

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc6_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_5_VC7_DPCM_CTRL_0

Offset: 0xe06d
Byte Offset: 0x381b4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc7_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_5_VC8_DPCM_CTRL_0

Offset: 0xe06e
Byte Offset: 0x381b8
Read/Write: R/W

Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc8_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_5_VC9_DPCM_CTRL_0

Offset: 0xe06f
Byte Offset: 0x381bc
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc9_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_5_VC10_DPCM_CTRL_0

Offset: 0xe070
Byte Offset: 0x381c0

Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc10_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_5_VC11_DPCM_CTRL_0

Offset: 0xe071
Byte Offset: 0x381c4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc11_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_5_VC12_DPCM_CTRL_0

Offset: 0xe072

Byte Offset: 0x381c8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc12_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_5_VC13_DPCM_CTRL_0

Offset: 0xe073
Byte Offset: 0x381cc
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc13_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_5_VC14_DPCM_CTRL_0

Offset: 0xe074
Byte Offset: 0x381d0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc14_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_5_VC15_DPCM_CTRL_0

Offset: 0xe075
Byte Offset: 0x381d4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	BYPASS	**cfg_vc15_dpcm_compression_ratio**: DPCM type **0** = BYPASS **1** = dpcm10_8_10 **2** = dpcm10_7_10 **3** = dpcm10_6_10 **4** = dpcm12_8_12 **5** = dpcm12_7_12 **6** = dpcm12_6_12 **7** = dpcm14_10_14 **8** = dpcm14_8_14 **9** = dpcm12_10_12

NVCSI_STREAM_5_PF_CRC_0

Status register on packet data CRC
Offset: 0xe076
Byte Offset: 0x381d8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**calc_crc**: 16 bit CRC computed over current packet (has to match with the CRC in PF for a good packet)
15:0	0x0	**rx_crc**: 16 bit CRC from PF

NVCSI_STREAM_5_PH_WC_0

Status register on WC
Offset: 0xe077
Byte Offset: 0x381dc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**calc_wc**: 16 bit WC computed over current packet (has to match with the WC in PH for a good packet)
15:0	0x0	**rx_wc**: 16 bit WC from PH

NVCSI_STREAM_5_PH_DI_0

Status register on Data ID
Offset: 0xe078
Byte Offset: 0x381e0
Read/Write: RO
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9:6	0x0	**rx_vc**: 4 bit VC from PH
5:0	0x0	**rx_dt**: 6 bit DTYPE from PH

NVCSI_STREAM_5_ERROR_STATUS2VI_MASK_0

Offset: 0xe079
Byte Offset: 0x381e4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15	0x0	**cfg_err_status2vi_mask_vc15**: for VC15 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
14	0x0	**cfg_err_status2vi_mask_vc14**: for VC14 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
13	0x0	**cfg_err_status2vi_mask_vc13**: for VC13 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
12	0x0	**cfg_err_status2vi_mask_vc12**: for VC12 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
11	0x0	**cfg_err_status2vi_mask_vc11**: for VC11 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
10	0x0	**cfg_err_status2vi_mask_vc10**: for VC10 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
9	0x0	**cfg_err_status2vi_mask_vc9**: for VC9 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF

Bit	Reset	Description
8	0x0	**cfg_err_status2vi_mask_vc8**: for VC8 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
7	0x0	**cfg_err_status2vi_mask_vc7**: for VC7 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
6	0x0	**cfg_err_status2vi_mask_vc6**: for VC6 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
5	0x0	**cfg_err_status2vi_mask_vc5**: for VC5 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
4	0x0	**cfg_err_status2vi_mask_vc4**: for VC4 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
3	0x0	**cfg_err_status2vi_mask_vc3**: for VC3 : 0 = err_status2vi_vc3 will be send to VI at EOF; 1 = No error will be send to VI at EOF
2	0x0	**cfg_err_status2vi_mask_vc2**: for VC2 : 0 = err_status2vi_vc2 will be send to VI at EOF; 1 = No error will be send to VI at EOF
1	0x0	**cfg_err_status2vi_mask_vc1**: for VC1 : 0 = err_status2vi_vc1 will be send to VI at EOF; 1 = No error will be send to VI at EOF
0	0x0	**cfg_err_status2vi_mask_vc0**: for VC0 : 0 = err_status2vi_vc0 will be send to VI at EOF; 1 = No error will be send to VI at EOF

NVCSI_STREAM_5_ERROR_STATUS2VI_VC0_0

This is RO register for SW
Offset: 0xe07a
Byte Offset: 0x381e8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc0**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_5_ERROR_STATUS2VI_VC1_0

This is RO register for SW
Offset: 0xe07b
Byte Offset: 0x381ec
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc1**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_5_ERROR_STATUS2VI_VC2_0

This is RO register for SW
Offset: 0xe07c
Byte Offset: 0x381f0
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc2**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_5_ERROR_STATUS2VI_VC3_0

This is RO register for SW
Offset: 0xe07d
Byte Offset: 0x381f4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc3**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_5_ERROR_STATUS2VI_VC4_0

This is RO register for SW
Offset: 0xe07e
Byte Offset: 0x381f8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc4**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_5_ERROR_STATUS2VI_VC5_0

This is RO register for SW
Offset: 0xe07f
Byte Offset: 0x381fc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc5**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_5_ERROR_STATUS2VI_VC6_0

This is RO register for SW
Offset: 0xe080
Byte Offset: 0x38200
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc6**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_5_ERROR_STATUS2VI_VC7_0

This is RO register for SW
Offset: 0xe081
Byte Offset: 0x38204
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc7**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_5_ERROR_STATUS2VI_VC8_0

This is RO register for SW
Offset: 0xe082
Byte Offset: 0x38208
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc8**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_5_ERROR_STATUS2VI_VC9_0

This is RO register for SW
Offset: 0xe083
Byte Offset: 0x3820c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc9**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_5_ERROR_STATUS2VI_VC10_0

This is RO register for SW
Offset: 0xe084
Byte Offset: 0x38210
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc10**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_5_ERROR_STATUS2VI_VC11_0

This is RO register for SW
Offset: 0xe085
Byte Offset: 0x38214
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc11**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_5_ERROR_STATUS2VI_VC12_0

This is RO register for SW
Offset: 0xe086
Byte Offset: 0x38218
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc12**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_5_ERROR_STATUS2VI_VC13_0

This is RO register for SW
Offset: 0xe087
Byte Offset: 0x3821c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc13**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_5_ERROR_STATUS2VI_VC14_0

This is RO register for SW
Offset: 0xe088
Byte Offset: 0x38220
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc14**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_5_ERROR_STATUS2VI_VC15_0

This is RO register for SW
Offset: 0xe089
Byte Offset: 0x38224
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**err_status2vi_vc15**: [0] : PP fsm timeout error [1] : PH ECC single bit error [2] : Packet Payload CRC error [3] : Packet Payload is less than WC in PH [4] : PH one CRC error [5] : Embedded line CRC error

NVCSI_STREAM_5_INTR_STATUS_0

Offset: 0xe08a
Byte Offset: 0x38228
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**intr_stat_novc**: Not VC related interrupt
15	0x0	**intr_stat_vc15**: VC15 event

Bit	Reset	Description
14	0x0	**intr_stat_vc14**: VC14 event
13	0x0	**intr_stat_vc13**: VC13 event
12	0x0	**intr_stat_vc12**: VC12 event
11	0x0	**intr_stat_vc11**: VC11 event
10	0x0	**intr_stat_vc10**: VC10 event
9	0x0	**intr_stat_vc9**: VC9 event
8	0x0	**intr_stat_vc8**: VC8 event
7	0x0	**intr_stat_vc7**: VC7 event
6	0x0	**intr_stat_vc6**: VC6 event
5	0x0	**intr_stat_vc5**: VC5 event
4	0x0	**intr_stat_vc4**: VC4 event
3	0x0	**intr_stat_vc3**: VC3 event
2	0x0	**intr_stat_vc2**: VC2 event
1	0x0	**intr_stat_vc1**: VC1 event
0	0x0	**intr_stat_vc0**: VC0 event

NVCSI_STREAM_5_INTR_STATUS_NOVC_0

Offset: 0xe08b
Byte Offset: 0x3822c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**intr_stat_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**intr_stat_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_5_INTR_STATUS_VC0_0

Offset: 0xe08c
Byte Offset: 0x38230
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc0**
4	0x0	**intr_stat_ph_single_crc_err_vc0**
3	0x0	**intr_stat_pd_wc_short_err_vc0**
2	0x0	**intr_stat_pd_crc_err_vc0**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc0**
0	0x0	**intr_stat_ppfsm_timeout_vc0**

NVCSI_STREAM_5_INTR_STATUS_VC1_0

Offset: 0xe08d
Byte Offset: 0x38234
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc1**

Bit	Reset	Description
4	0x0	**intr_stat_ph_single_crc_err_vc1**
3	0x0	**intr_stat_pd_wc_short_err_vc1**
2	0x0	**intr_stat_pd_crc_err_vc1**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc1**
0	0x0	**intr_stat_ppfsm_timeout_vc1**

NVCSI_STREAM_5_INTR_STATUS_VC2_0

Offset: 0xe08e
Byte Offset: 0x38238
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc2**
4	0x0	**intr_stat_ph_single_crc_err_vc2**
3	0x0	**intr_stat_pd_wc_short_err_vc2**
2	0x0	**intr_stat_pd_crc_err_vc2**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc2**
0	0x0	**intr_stat_ppfsm_timeout_vc2**

NVCSI_STREAM_5_INTR_STATUS_VC3_0

Offset: 0xe08f
Byte Offset: 0x3823c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc3**
4	0x0	**intr_stat_ph_single_crc_err_vc3**
3	0x0	**intr_stat_pd_wc_short_err_vc3**
2	0x0	**intr_stat_pd_crc_err_vc3**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc3**
0	0x0	**intr_stat_ppfsm_timeout_vc3**

NVCSI_STREAM_5_INTR_STATUS_VC4_0

Offset: 0xe090
Byte Offset: 0x38240
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc4**
4	0x0	**intr_stat_ph_single_crc_err_vc4**
3	0x0	**intr_stat_pd_wc_short_err_vc4**
2	0x0	**intr_stat_pd_crc_err_vc4**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc4**
0	0x0	**intr_stat_ppfsm_timeout_vc4**

NVCSI_STREAM_5_INTR_STATUS_VC5_0

Offset: 0xe091
Byte Offset: 0x38244
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc5**
4	0x0	**intr_stat_ph_single_crc_err_vc5**
3	0x0	**intr_stat_pd_wc_short_err_vc5**
2	0x0	**intr_stat_pd_crc_err_vc5**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc5**
0	0x0	**intr_stat_ppfsm_timeout_vc5**

NVCSI_STREAM_5_INTR_STATUS_VC6_0

Offset: 0xe092
Byte Offset: 0x38248
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc6**
4	0x0	**intr_stat_ph_single_crc_err_vc6**
3	0x0	**intr_stat_pd_wc_short_err_vc6**
2	0x0	**intr_stat_pd_crc_err_vc6**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc6**
0	0x0	**intr_stat_ppfsm_timeout_vc6**

NVCSI_STREAM_5_INTR_STATUS_VC7_0

Offset: 0xe093
Byte Offset: 0x3824c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc7**
4	0x0	**intr_stat_ph_single_crc_err_vc7**
3	0x0	**intr_stat_pd_wc_short_err_vc7**
2	0x0	**intr_stat_pd_crc_err_vc7**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc7**
0	0x0	**intr_stat_ppfsm_timeout_vc7**

NVCSI_STREAM_5_INTR_STATUS_VC8_0

Offset: 0xe094
Byte Offset: 0x38250
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc8**
4	0x0	**intr_stat_ph_single_crc_err_vc8**
3	0x0	**intr_stat_pd_wc_short_err_vc8**
2	0x0	**intr_stat_pd_crc_err_vc8**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc8**
0	0x0	**intr_stat_ppfsm_timeout_vc8**

NVCSI_STREAM_5_INTR_STATUS_VC9_0

Offset: 0xe095
Byte Offset: 0x38254
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc9**
4	0x0	**intr_stat_ph_single_crc_err_vc9**
3	0x0	**intr_stat_pd_wc_short_err_vc9**
2	0x0	**intr_stat_pd_crc_err_vc9**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc9**
0	0x0	**intr_stat_ppfsm_timeout_vc9**

NVCSI_STREAM_5_INTR_STATUS_VC10_0

Offset: 0xe096
Byte Offset: 0x38258
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc10**
4	0x0	**intr_stat_ph_single_crc_err_vc10**
3	0x0	**intr_stat_pd_wc_short_err_vc10**
2	0x0	**intr_stat_pd_crc_err_vc10**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc10**
0	0x0	**intr_stat_ppfsm_timeout_vc10**

NVCSI_STREAM_5_INTR_STATUS_VC11_0

Offset: 0xe097
Byte Offset: 0x3825c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc11**
4	0x0	**intr_stat_ph_single_crc_err_vc11**
3	0x0	**intr_stat_pd_wc_short_err_vc11**
2	0x0	**intr_stat_pd_crc_err_vc11**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc11**
0	0x0	**intr_stat_ppfsm_timeout_vc11**

NVCSI_STREAM_5_INTR_STATUS_VC12_0

Offset: 0xe098
Byte Offset: 0x38260
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc12**
4	0x0	**intr_stat_ph_single_crc_err_vc12**
3	0x0	**intr_stat_pd_wc_short_err_vc12**
2	0x0	**intr_stat_pd_crc_err_vc12**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc12**
0	0x0	**intr_stat_ppfsm_timeout_vc12**

NVCSI_STREAM_5_INTR_STATUS_VC13_0

Offset: 0xe099
Byte Offset: 0x38264
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc13**
4	0x0	**intr_stat_ph_single_crc_err_vc13**
3	0x0	**intr_stat_pd_wc_short_err_vc13**
2	0x0	**intr_stat_pd_crc_err_vc13**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc13**
0	0x0	**intr_stat_ppfsm_timeout_vc13**

NVCSI_STREAM_5_INTR_STATUS_VC14_0

Offset: 0xe09a
Byte Offset: 0x38268
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc14**
4	0x0	**intr_stat_ph_single_crc_err_vc14**
3	0x0	**intr_stat_pd_wc_short_err_vc14**
2	0x0	**intr_stat_pd_crc_err_vc14**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc14**
0	0x0	**intr_stat_ppfsm_timeout_vc14**

NVCSI_STREAM_5_INTR_STATUS_VC15_0

Offset: 0xe09b
Byte Offset: 0x3826c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_stat_embedded_line_crc_err_vc15**
4	0x0	**intr_stat_ph_single_crc_err_vc15**
3	0x0	**intr_stat_pd_wc_short_err_vc15**
2	0x0	**intr_stat_pd_crc_err_vc15**
1	0x0	**intr_stat_ph_ecc_single_bit_err_vc15**
0	0x0	**intr_stat_ppfsm_timeout_vc15**

NVCSI_STREAM_5_INTR_MASK_NOVC_0

Offset: 0xe09c
Byte Offset: 0x38270
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**intr_mask_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**intr_mask_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_5_INTR_MASK_VC0_0

Offset: 0xe09d
Byte Offset: 0x38274
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc0**
4	0x0	**intr_mask_ph_single_crc_err_vc0**

Bit	Reset	Description
3	0x0	**intr_mask_pd_wc_short_err_vc0**
2	0x0	**intr_mask_pd_crc_err_vc0**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc0**
0	0x0	**intr_mask_ppfsm_timeout_vc0**

NVCSI_STREAM_5_INTR_MASK_VC1_0

Offset: 0xe09e
Byte Offset: 0x38278
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc1**
4	0x0	**intr_mask_ph_single_crc_err_vc1**
3	0x0	**intr_mask_pd_wc_short_err_vc1**
2	0x0	**intr_mask_pd_crc_err_vc1**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc1**
0	0x0	**intr_mask_ppfsm_timeout_vc1**

NVCSI_STREAM_5_INTR_MASK_VC2_0

Offset: 0xe09f
Byte Offset: 0x3827c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc2**

Bit	Reset	Description
4	0x0	**intr_mask_ph_single_crc_err_vc2**
3	0x0	**intr_mask_pd_wc_short_err_vc2**
2	0x0	**intr_mask_pd_crc_err_vc2**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc2**
0	0x0	**intr_mask_ppfsm_timeout_vc2**

NVCSI_STREAM_5_INTR_MASK_VC3_0

Offset: 0xe0a0
Byte Offset: 0x38280
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc3**
4	0x0	**intr_mask_ph_single_crc_err_vc3**
3	0x0	**intr_mask_pd_wc_short_err_vc3**
2	0x0	**intr_mask_pd_crc_err_vc3**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc3**
0	0x0	**intr_mask_ppfsm_timeout_vc3**

NVCSI_STREAM_5_INTR_MASK_VC4_0

Offset: 0xe0a1
Byte Offset: 0x38284
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc4**
4	0x0	**intr_mask_ph_single_crc_err_vc4**
3	0x0	**intr_mask_pd_wc_short_err_vc4**
2	0x0	**intr_mask_pd_crc_err_vc4**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc4**
0	0x0	**intr_mask_ppfsm_timeout_vc4**

NVCSI_STREAM_5_INTR_MASK_VC5_0

Offset: 0xe0a2
Byte Offset: 0x38288
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc5**
4	0x0	**intr_mask_ph_single_crc_err_vc5**
3	0x0	**intr_mask_pd_wc_short_err_vc5**
2	0x0	**intr_mask_pd_crc_err_vc5**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc5**
0	0x0	**intr_mask_ppfsm_timeout_vc5**

NVCSI_STREAM_5_INTR_MASK_VC6_0

Offset: 0xe0a3
Byte Offset: 0x3828c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc6**
4	0x0	**intr_mask_ph_single_crc_err_vc6**
3	0x0	**intr_mask_pd_wc_short_err_vc6**
2	0x0	**intr_mask_pd_crc_err_vc6**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc6**
0	0x0	**intr_mask_ppfsm_timeout_vc6**

NVCSI_STREAM_5_INTR_MASK_VC7_0

Offset: 0xe0a4
Byte Offset: 0x38290
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc7**
4	0x0	**intr_mask_ph_single_crc_err_vc7**
3	0x0	**intr_mask_pd_wc_short_err_vc7**
2	0x0	**intr_mask_pd_crc_err_vc7**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc7**
0	0x0	**intr_mask_ppfsm_timeout_vc7**

NVCSI_STREAM_5_INTR_MASK_VC8_0

Offset: 0xe0a5
Byte Offset: 0x38294
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc8**
4	0x0	**intr_mask_ph_single_crc_err_vc8**
3	0x0	**intr_mask_pd_wc_short_err_vc8**
2	0x0	**intr_mask_pd_crc_err_vc8**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc8**
0	0x0	**intr_mask_ppfsm_timeout_vc8**

NVCSI_STREAM_5_INTR_MASK_VC9_0

Offset: 0xe0a6
Byte Offset: 0x38298
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc9**
4	0x0	**intr_mask_ph_single_crc_err_vc9**
3	0x0	**intr_mask_pd_wc_short_err_vc9**
2	0x0	**intr_mask_pd_crc_err_vc9**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc9**
0	0x0	**intr_mask_ppfsm_timeout_vc9**

NVCSI_STREAM_5_INTR_MASK_VC10_0

Offset: 0xe0a7
Byte Offset: 0x3829c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc10**
4	0x0	**intr_mask_ph_single_crc_err_vc10**
3	0x0	**intr_mask_pd_wc_short_err_vc10**
2	0x0	**intr_mask_pd_crc_err_vc10**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc10**
0	0x0	**intr_mask_ppfsm_timeout_vc10**

NVCSI_STREAM_5_INTR_MASK_VC11_0

Offset: 0xe0a8
Byte Offset: 0x382a0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc11**
4	0x0	**intr_mask_ph_single_crc_err_vc11**
3	0x0	**intr_mask_pd_wc_short_err_vc11**
2	0x0	**intr_mask_pd_crc_err_vc11**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc11**
0	0x0	**intr_mask_ppfsm_timeout_vc11**

NVCSI_STREAM_5_INTR_MASK_VC12_0

Offset: 0xe0a9
Byte Offset: 0x382a4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc12**
4	0x0	**intr_mask_ph_single_crc_err_vc12**
3	0x0	**intr_mask_pd_wc_short_err_vc12**
2	0x0	**intr_mask_pd_crc_err_vc12**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc12**
0	0x0	**intr_mask_ppfsm_timeout_vc12**

NVCSI_STREAM_5_INTR_MASK_VC13_0

Offset: 0xe0aa
Byte Offset: 0x382a8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc13**
4	0x0	**intr_mask_ph_single_crc_err_vc13**
3	0x0	**intr_mask_pd_wc_short_err_vc13**
2	0x0	**intr_mask_pd_crc_err_vc13**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc13**
0	0x0	**intr_mask_ppfsm_timeout_vc13**

NVCSI_STREAM_5_INTR_MASK_VC14_0

Offset: 0xe0ab
Byte Offset: 0x382ac
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc14**
4	0x0	**intr_mask_ph_single_crc_err_vc14**
3	0x0	**intr_mask_pd_wc_short_err_vc14**
2	0x0	**intr_mask_pd_crc_err_vc14**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc14**
0	0x0	**intr_mask_ppfsm_timeout_vc14**

NVCSI_STREAM_5_INTR_MASK_VC15_0

Offset: 0xe0ac
Byte Offset: 0x382b0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**intr_mask_embedded_line_crc_err_vc15**
4	0x0	**intr_mask_ph_single_crc_err_vc15**
3	0x0	**intr_mask_pd_wc_short_err_vc15**
2	0x0	**intr_mask_pd_crc_err_vc15**
1	0x0	**intr_mask_ph_ecc_single_bit_err_vc15**
0	0x0	**intr_mask_ppfsm_timeout_vc15**

NVCSI_STREAM_5_CORRECTABLE_ERR_INTR_STATUS_0

Offset: 0xe0ad
Byte Offset: 0x382b4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**err_intr_stat_novc**: Not VC related interrupt
15	0x0	**err_intr_stat_vc15**: VC15 event
14	0x0	**err_intr_stat_vc14**: VC14 event
13	0x0	**err_intr_stat_vc13**: VC13 event
12	0x0	**err_intr_stat_vc12**: VC12 event
11	0x0	**err_intr_stat_vc11**: VC11 event
10	0x0	**err_intr_stat_vc10**: VC10 event
9	0x0	**err_intr_stat_vc9**: VC9 event
8	0x0	**err_intr_stat_vc8**: VC8 event
7	0x0	**err_intr_stat_vc7**: VC7 event
6	0x0	**err_intr_stat_vc6**: VC6 event
5	0x0	**err_intr_stat_vc5**: VC5 event
4	0x0	**err_intr_stat_vc4**: VC4 event
3	0x0	**err_intr_stat_vc3**: VC3 event
2	0x0	**err_intr_stat_vc2**: VC2 event
1	0x0	**err_intr_stat_vc1**: VC1 event
0	0x0	**err_intr_stat_vc0**: VC0 event

NVCSI_STREAM_5_CORRECTABLE_ERR_INTR_STATUS_NOVC_0

Offset: 0xe0ae
Byte Offset: 0x382b8
Read/Write: R/W

Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**err_intr_stat_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**err_intr_stat_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_5_CORRECTABLE_ERR_INTR_STATUS_VC0_0

Offset: 0xe0af
Byte Offset: 0x382bc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc0**
4	0x0	**err_intr_stat_ph_single_crc_err_vc0**
3	0x0	**err_intr_stat_pd_wc_short_err_vc0**
2	0x0	**err_intr_stat_pd_crc_err_vc0**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc0**
0	0x0	**err_intr_stat_ppfsm_timeout_vc0**

NVCSI_STREAM_5_CORRECTABLE_ERR_INTR_STATUS_VC1_0

Offset: 0xe0b0
Byte Offset: 0x382c0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc1**
4	0x0	**err_intr_stat_ph_single_crc_err_vc1**
3	0x0	**err_intr_stat_pd_wc_short_err_vc1**
2	0x0	**err_intr_stat_pd_crc_err_vc1**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc1**
0	0x0	**err_intr_stat_ppfsm_timeout_vc1**

NVCSI_STREAM_5_CORRECTABLE_ERR_INTR_STATUS_VC2_0

Offset: 0xe0b1
Byte Offset: 0x382c4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc2**
4	0x0	**err_intr_stat_ph_single_crc_err_vc2**
3	0x0	**err_intr_stat_pd_wc_short_err_vc2**
2	0x0	**err_intr_stat_pd_crc_err_vc2**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc2**
0	0x0	**err_intr_stat_ppfsm_timeout_vc2**

NVCSI_STREAM_5_CORRECTABLE_ERR_INTR_STATUS_VC3_0

Offset: 0xe0b2
Byte Offset: 0x382c8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc3**
4	0x0	**err_intr_stat_ph_single_crc_err_vc3**
3	0x0	**err_intr_stat_pd_wc_short_err_vc3**
2	0x0	**err_intr_stat_pd_crc_err_vc3**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc3**
0	0x0	**err_intr_stat_ppfsm_timeout_vc3**

NVCSI_STREAM_5_CORRECTABLE_ERR_INTR_STATUS_VC4_0

Offset: 0xe0b3
Byte Offset: 0x382cc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc4**
4	0x0	**err_intr_stat_ph_single_crc_err_vc4**
3	0x0	**err_intr_stat_pd_wc_short_err_vc4**
2	0x0	**err_intr_stat_pd_crc_err_vc4**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc4**
0	0x0	**err_intr_stat_ppfsm_timeout_vc4**

NVCSI_STREAM_5_CORRECTABLE_ERR_INTR_STATUS_VC5_0

Offset: 0xe0b4
Byte Offset: 0x382d0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc5**
4	0x0	**err_intr_stat_ph_single_crc_err_vc5**
3	0x0	**err_intr_stat_pd_wc_short_err_vc5**
2	0x0	**err_intr_stat_pd_crc_err_vc5**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc5**
0	0x0	**err_intr_stat_ppfsm_timeout_vc5**

NVCSI_STREAM_5_CORRECTABLE_ERR_INTR_STATUS_VC6_0

Offset: 0xe0b5
Byte Offset: 0x382d4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc6**
4	0x0	**err_intr_stat_ph_single_crc_err_vc6**
3	0x0	**err_intr_stat_pd_wc_short_err_vc6**
2	0x0	**err_intr_stat_pd_crc_err_vc6**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc6**
0	0x0	**err_intr_stat_ppfsm_timeout_vc6**

NVCSI_STREAM_5_CORRECTABLE_ERR_INTR_STATUS_VC7_0

Offset: 0xe0b6
Byte Offset: 0x382d8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc7**
4	0x0	**err_intr_stat_ph_single_crc_err_vc7**
3	0x0	**err_intr_stat_pd_wc_short_err_vc7**
2	0x0	**err_intr_stat_pd_crc_err_vc7**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc7**
0	0x0	**err_intr_stat_ppfsm_timeout_vc7**

NVCSI_STREAM_5_CORRECTABLE_ERR_INTR_STATUS_VC8_0

Offset: 0xe0b7
Byte Offset: 0x382dc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc8**
4	0x0	**err_intr_stat_ph_single_crc_err_vc8**
3	0x0	**err_intr_stat_pd_wc_short_err_vc8**
2	0x0	**err_intr_stat_pd_crc_err_vc8**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc8**
0	0x0	**err_intr_stat_ppfsm_timeout_vc8**

NVCSI_STREAM_5_CORRECTABLE_ERR_INTR_STATUS_VC9_0

Offset: 0xe0b8
Byte Offset: 0x382e0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc9**
4	0x0	**err_intr_stat_ph_single_crc_err_vc9**
3	0x0	**err_intr_stat_pd_wc_short_err_vc9**
2	0x0	**err_intr_stat_pd_crc_err_vc9**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc9**
0	0x0	**err_intr_stat_ppfsm_timeout_vc9**

NVCSI_STREAM_5_CORRECTABLE_ERR_INTR_STATUS_VC10_0

Offset: 0xe0b9
Byte Offset: 0x382e4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc10**
4	0x0	**err_intr_stat_ph_single_crc_err_vc10**
3	0x0	**err_intr_stat_pd_wc_short_err_vc10**
2	0x0	**err_intr_stat_pd_crc_err_vc10**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc10**
0	0x0	**err_intr_stat_ppfsm_timeout_vc10**

NVCSI_STREAM_5_CORRECTABLE_ERR_INTR_STATUS_VC11_0

Offset: 0xe0ba
Byte Offset: 0x382e8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc11**
4	0x0	**err_intr_stat_ph_single_crc_err_vc11**
3	0x0	**err_intr_stat_pd_wc_short_err_vc11**
2	0x0	**err_intr_stat_pd_crc_err_vc11**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc11**
0	0x0	**err_intr_stat_ppfsm_timeout_vc11**

NVCSI_STREAM_5_CORRECTABLE_ERR_INTR_STATUS_VC12_0

Offset: 0xe0bb
Byte Offset: 0x382ec
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc12**
4	0x0	**err_intr_stat_ph_single_crc_err_vc12**
3	0x0	**err_intr_stat_pd_wc_short_err_vc12**
2	0x0	**err_intr_stat_pd_crc_err_vc12**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc12**
0	0x0	**err_intr_stat_ppfsm_timeout_vc12**

NVCSI_STREAM_5_CORRECTABLE_ERR_INTR_STATUS_VC13_0

Offset: 0xe0bc
Byte Offset: 0x382f0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc13**
4	0x0	**err_intr_stat_ph_single_crc_err_vc13**
3	0x0	**err_intr_stat_pd_wc_short_err_vc13**
2	0x0	**err_intr_stat_pd_crc_err_vc13**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc13**
0	0x0	**err_intr_stat_ppfsm_timeout_vc13**

NVCSI_STREAM_5_CORRECTABLE_ERR_INTR_STATUS_VC14_0

Offset: 0xe0bd
Byte Offset: 0x382f4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc14**
4	0x0	**err_intr_stat_ph_single_crc_err_vc14**
3	0x0	**err_intr_stat_pd_wc_short_err_vc14**
2	0x0	**err_intr_stat_pd_crc_err_vc14**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc14**
0	0x0	**err_intr_stat_ppfsm_timeout_vc14**

NVCSI_STREAM_5_CORRECTABLE_ERR_INTR_STATUS_VC15_0

Offset: 0xe0be
Byte Offset: 0x382f8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc15**
4	0x0	**err_intr_stat_ph_single_crc_err_vc15**
3	0x0	**err_intr_stat_pd_wc_short_err_vc15**
2	0x0	**err_intr_stat_pd_crc_err_vc15**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc15**
0	0x0	**err_intr_stat_ppfsm_timeout_vc15**

NVCSI_STREAM_5_UNCORRECTABLE_ERR_INTR_STATUS_0

Offset: 0xe0bf
Byte Offset: 0x382fc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**err_intr_stat_novc**: Not VC related interrupt
15	0x0	**err_intr_stat_vc15**: VC15 event
14	0x0	**err_intr_stat_vc14**: VC14 event
13	0x0	**err_intr_stat_vc13**: VC13 event
12	0x0	**err_intr_stat_vc12**: VC12 event
11	0x0	**err_intr_stat_vc11**: VC11 event
10	0x0	**err_intr_stat_vc10**: VC10 event
9	0x0	**err_intr_stat_vc9**: VC9 event
8	0x0	**err_intr_stat_vc8**: VC8 event
7	0x0	**err_intr_stat_vc7**: VC7 event

Bit	Reset	Description
6	0x0	**err_intr_stat_vc6**: VC6 event
5	0x0	**err_intr_stat_vc5**: VC5 event
4	0x0	**err_intr_stat_vc4**: VC4 event
3	0x0	**err_intr_stat_vc3**: VC3 event
2	0x0	**err_intr_stat_vc2**: VC2 event
1	0x0	**err_intr_stat_vc1**: VC1 event
0	0x0	**err_intr_stat_vc0**: VC0 event

NVCSI_STREAM_5_UNCORRECTABLE_ERR_INTR_STATUS_NOVC_0

Offset: 0xe0c0
Byte Offset: 0x38300
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**err_intr_stat_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**err_intr_stat_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_5_UNCORRECTABLE_ERR_INTR_STATUS_VC0_0

Offset: 0xe0c1
Byte Offset: 0x38304
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc0**
4	0x0	**err_intr_stat_ph_single_crc_err_vc0**
3	0x0	**err_intr_stat_pd_wc_short_err_vc0**
2	0x0	**err_intr_stat_pd_crc_err_vc0**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc0**
0	0x0	**err_intr_stat_ppfsm_timeout_vc0**

NVCSI_STREAM_5_UNCORRECTABLE_ERR_INTR_STATUS_VC1_0

Offset: 0xe0c2
Byte Offset: 0x38308
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc1**
4	0x0	**err_intr_stat_ph_single_crc_err_vc1**
3	0x0	**err_intr_stat_pd_wc_short_err_vc1**
2	0x0	**err_intr_stat_pd_crc_err_vc1**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc1**
0	0x0	**err_intr_stat_ppfsm_timeout_vc1**

NVCSI_STREAM_5_UNCORRECTABLE_ERR_INTR_STATUS_VC2_0

Offset: 0xe0c3
Byte Offset: 0x3830c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc2**
4	0x0	**err_intr_stat_ph_single_crc_err_vc2**
3	0x0	**err_intr_stat_pd_wc_short_err_vc2**
2	0x0	**err_intr_stat_pd_crc_err_vc2**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc2**
0	0x0	**err_intr_stat_ppfsm_timeout_vc2**

NVCSI_STREAM_5_UNCORRECTABLE_ERR_INTR_STATUS_VC3_0

Offset: 0xe0c4
Byte Offset: 0x38310
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc3**
4	0x0	**err_intr_stat_ph_single_crc_err_vc3**
3	0x0	**err_intr_stat_pd_wc_short_err_vc3**
2	0x0	**err_intr_stat_pd_crc_err_vc3**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc3**
0	0x0	**err_intr_stat_ppfsm_timeout_vc3**

NVCSI_STREAM_5_UNCORRECTABLE_ERR_INTR_STATUS_VC4_0

Offset: 0xe0c5
Byte Offset: 0x38314
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc4**
4	0x0	**err_intr_stat_ph_single_crc_err_vc4**
3	0x0	**err_intr_stat_pd_wc_short_err_vc4**
2	0x0	**err_intr_stat_pd_crc_err_vc4**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc4**
0	0x0	**err_intr_stat_ppfsm_timeout_vc4**

NVCSI_STREAM_5_UNCORRECTABLE_ERR_INTR_STATUS_VC5_0

Offset: 0xe0c6
Byte Offset: 0x38318
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc5**
4	0x0	**err_intr_stat_ph_single_crc_err_vc5**
3	0x0	**err_intr_stat_pd_wc_short_err_vc5**
2	0x0	**err_intr_stat_pd_crc_err_vc5**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc5**
0	0x0	**err_intr_stat_ppfsm_timeout_vc5**

NVCSI_STREAM_5_UNCORRECTABLE_ERR_INTR_STATUS_VC6_0

Offset: 0xe0c7
Byte Offset: 0x3831c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc6**
4	0x0	**err_intr_stat_ph_single_crc_err_vc6**
3	0x0	**err_intr_stat_pd_wc_short_err_vc6**
2	0x0	**err_intr_stat_pd_crc_err_vc6**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc6**
0	0x0	**err_intr_stat_ppfsm_timeout_vc6**

NVCSI_STREAM_5_UNCORRECTABLE_ERR_INTR_STATUS_VC7_0

Offset: 0xe0c8
Byte Offset: 0x38320
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc7**
4	0x0	**err_intr_stat_ph_single_crc_err_vc7**
3	0x0	**err_intr_stat_pd_wc_short_err_vc7**
2	0x0	**err_intr_stat_pd_crc_err_vc7**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc7**
0	0x0	**err_intr_stat_ppfsm_timeout_vc7**

NVCSI_STREAM_5_UNCORRECTABLE_ERR_INTR_STATUS_VC8_0

Offset: 0xe0c9
Byte Offset: 0x38324
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc8**
4	0x0	**err_intr_stat_ph_single_crc_err_vc8**
3	0x0	**err_intr_stat_pd_wc_short_err_vc8**
2	0x0	**err_intr_stat_pd_crc_err_vc8**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc8**
0	0x0	**err_intr_stat_ppfsm_timeout_vc8**

NVCSI_STREAM_5_UNCORRECTABLE_ERR_INTR_STATUS_VC9_0

Offset: 0xe0ca
Byte Offset: 0x38328
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc9**
4	0x0	**err_intr_stat_ph_single_crc_err_vc9**
3	0x0	**err_intr_stat_pd_wc_short_err_vc9**
2	0x0	**err_intr_stat_pd_crc_err_vc9**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc9**
0	0x0	**err_intr_stat_ppfsm_timeout_vc9**

NVCSI_STREAM_5_UNCORRECTABLE_ERR_INTR_STATUS_VC10_0

Offset: 0xe0cb
Byte Offset: 0x3832c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc10**
4	0x0	**err_intr_stat_ph_single_crc_err_vc10**
3	0x0	**err_intr_stat_pd_wc_short_err_vc10**
2	0x0	**err_intr_stat_pd_crc_err_vc10**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc10**
0	0x0	**err_intr_stat_ppfsm_timeout_vc10**

NVCSI_STREAM_5_UNCORRECTABLE_ERR_INTR_STATUS_VC11_0

Offset: 0xe0cc
Byte Offset: 0x38330
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc11**
4	0x0	**err_intr_stat_ph_single_crc_err_vc11**
3	0x0	**err_intr_stat_pd_wc_short_err_vc11**
2	0x0	**err_intr_stat_pd_crc_err_vc11**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc11**
0	0x0	**err_intr_stat_ppfsm_timeout_vc11**

NVCSI_STREAM_5_UNCORRECTABLE_ERR_INTR_STATUS_VC12_0

Offset: 0xe0cd
Byte Offset: 0x38334
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc12**
4	0x0	**err_intr_stat_ph_single_crc_err_vc12**
3	0x0	**err_intr_stat_pd_wc_short_err_vc12**
2	0x0	**err_intr_stat_pd_crc_err_vc12**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc12**
0	0x0	**err_intr_stat_ppfsm_timeout_vc12**

NVCSI_STREAM_5_UNCORRECTABLE_ERR_INTR_STATUS_VC13_0

Offset: 0xe0ce
Byte Offset: 0x38338
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc13**
4	0x0	**err_intr_stat_ph_single_crc_err_vc13**
3	0x0	**err_intr_stat_pd_wc_short_err_vc13**
2	0x0	**err_intr_stat_pd_crc_err_vc13**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc13**
0	0x0	**err_intr_stat_ppfsm_timeout_vc13**

NVCSI_STREAM_5_UNCORRECTABLE_ERR_INTR_STATUS_VC14_0

Offset: 0xe0cf
Byte Offset: 0x3833c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc14**
4	0x0	**err_intr_stat_ph_single_crc_err_vc14**
3	0x0	**err_intr_stat_pd_wc_short_err_vc14**
2	0x0	**err_intr_stat_pd_crc_err_vc14**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc14**
0	0x0	**err_intr_stat_ppfsm_timeout_vc14**

NVCSI_STREAM_5_UNCORRECTABLE_ERR_INTR_STATUS_VC15_0

Offset: 0xe0d0
Byte Offset: 0x38340
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_stat_embedded_line_crc_err_vc15**
4	0x0	**err_intr_stat_ph_single_crc_err_vc15**
3	0x0	**err_intr_stat_pd_wc_short_err_vc15**
2	0x0	**err_intr_stat_pd_crc_err_vc15**
1	0x0	**err_intr_stat_ph_ecc_single_bit_err_vc15**
0	0x0	**err_intr_stat_ppfsm_timeout_vc15**

NVCSI_STREAM_5_ERR_INTR_MASK_NOVC_0

Offset: 0xe0d1
Byte Offset: 0x38344
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**err_intr_mask_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**err_intr_mask_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_5_ERR_INTR_MASK_VC0_0

Offset: 0xe0d2
Byte Offset: 0x38348
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc0**
4	0x0	**err_intr_mask_ph_single_crc_err_vc0**
3	0x0	**err_intr_mask_pd_wc_short_err_vc0**
2	0x0	**err_intr_mask_pd_crc_err_vc0**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc0**
0	0x0	**err_intr_mask_ppfsm_timeout_vc0**

NVCSI_STREAM_5_ERR_INTR_MASK_VC1_0

Offset: 0xe0d3
Byte Offset: 0x3834c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc1**

Bit	Reset	Description
4	0x0	**err_intr_mask_ph_single_crc_err_vc1**
3	0x0	**err_intr_mask_pd_wc_short_err_vc1**
2	0x0	**err_intr_mask_pd_crc_err_vc1**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc1**
0	0x0	**err_intr_mask_ppfsm_timeout_vc1**

NVCSI_STREAM_5_ERR_INTR_MASK_VC2_0

Offset: 0xe0d4
Byte Offset: 0x38350
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc2**
4	0x0	**err_intr_mask_ph_single_crc_err_vc2**
3	0x0	**err_intr_mask_pd_wc_short_err_vc2**
2	0x0	**err_intr_mask_pd_crc_err_vc2**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc2**
0	0x0	**err_intr_mask_ppfsm_timeout_vc2**

NVCSI_STREAM_5_ERR_INTR_MASK_VC3_0

Offset: 0xe0d5
Byte Offset: 0x38354
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc3**
4	0x0	**err_intr_mask_ph_single_crc_err_vc3**
3	0x0	**err_intr_mask_pd_wc_short_err_vc3**
2	0x0	**err_intr_mask_pd_crc_err_vc3**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc3**
0	0x0	**err_intr_mask_ppfsm_timeout_vc3**

NVCSI_STREAM_5_ERR_INTR_MASK_VC4_0

Offset: 0xe0d6
Byte Offset: 0x38358
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc4**
4	0x0	**err_intr_mask_ph_single_crc_err_vc4**
3	0x0	**err_intr_mask_pd_wc_short_err_vc4**
2	0x0	**err_intr_mask_pd_crc_err_vc4**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc4**
0	0x0	**err_intr_mask_ppfsm_timeout_vc4**

NVCSI_STREAM_5_ERR_INTR_MASK_VC5_0

Offset: 0xe0d7
Byte Offset: 0x3835c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc5**
4	0x0	**err_intr_mask_ph_single_crc_err_vc5**
3	0x0	**err_intr_mask_pd_wc_short_err_vc5**
2	0x0	**err_intr_mask_pd_crc_err_vc5**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc5**
0	0x0	**err_intr_mask_ppfsm_timeout_vc5**

NVCSI_STREAM_5_ERR_INTR_MASK_VC6_0

Offset: 0xe0d8
Byte Offset: 0x38360
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc6**
4	0x0	**err_intr_mask_ph_single_crc_err_vc6**
3	0x0	**err_intr_mask_pd_wc_short_err_vc6**
2	0x0	**err_intr_mask_pd_crc_err_vc6**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc6**
0	0x0	**err_intr_mask_ppfsm_timeout_vc6**

NVCSI_STREAM_5_ERR_INTR_MASK_VC7_0

Offset: 0xe0d9
Byte Offset: 0x38364
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc7**
4	0x0	**err_intr_mask_ph_single_crc_err_vc7**
3	0x0	**err_intr_mask_pd_wc_short_err_vc7**
2	0x0	**err_intr_mask_pd_crc_err_vc7**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc7**
0	0x0	**err_intr_mask_ppfsm_timeout_vc7**

NVCSI_STREAM_5_ERR_INTR_MASK_VC8_0

Offset: 0xe0da
Byte Offset: 0x38368
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc8**
4	0x0	**err_intr_mask_ph_single_crc_err_vc8**
3	0x0	**err_intr_mask_pd_wc_short_err_vc8**
2	0x0	**err_intr_mask_pd_crc_err_vc8**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc8**
0	0x0	**err_intr_mask_ppfsm_timeout_vc8**

NVCSI_STREAM_5_ERR_INTR_MASK_VC9_0

Offset: 0xe0db
Byte Offset: 0x3836c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc9**
4	0x0	**err_intr_mask_ph_single_crc_err_vc9**
3	0x0	**err_intr_mask_pd_wc_short_err_vc9**
2	0x0	**err_intr_mask_pd_crc_err_vc9**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc9**
0	0x0	**err_intr_mask_ppfsm_timeout_vc9**

NVCSI_STREAM_5_ERR_INTR_MASK_VC10_0

Offset: 0xe0dc
Byte Offset: 0x38370
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc10**
4	0x0	**err_intr_mask_ph_single_crc_err_vc10**
3	0x0	**err_intr_mask_pd_wc_short_err_vc10**
2	0x0	**err_intr_mask_pd_crc_err_vc10**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc10**
0	0x0	**err_intr_mask_ppfsm_timeout_vc10**

NVCSI_STREAM_5_ERR_INTR_MASK_VC11_0

Offset: 0xe0dd
Byte Offset: 0x38374
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc11**
4	0x0	**err_intr_mask_ph_single_crc_err_vc11**
3	0x0	**err_intr_mask_pd_wc_short_err_vc11**
2	0x0	**err_intr_mask_pd_crc_err_vc11**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc11**
0	0x0	**err_intr_mask_ppfsm_timeout_vc11**

NVCSI_STREAM_5_ERR_INTR_MASK_VC12_0

Offset: 0xe0de
Byte Offset: 0x38378
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc12**
4	0x0	**err_intr_mask_ph_single_crc_err_vc12**
3	0x0	**err_intr_mask_pd_wc_short_err_vc12**
2	0x0	**err_intr_mask_pd_crc_err_vc12**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc12**
0	0x0	**err_intr_mask_ppfsm_timeout_vc12**

NVCSI_STREAM_5_ERR_INTR_MASK_VC13_0

Offset: 0xe0df
Byte Offset: 0x3837c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc13**
4	0x0	**err_intr_mask_ph_single_crc_err_vc13**
3	0x0	**err_intr_mask_pd_wc_short_err_vc13**
2	0x0	**err_intr_mask_pd_crc_err_vc13**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc13**
0	0x0	**err_intr_mask_ppfsm_timeout_vc13**

NVCSI_STREAM_5_ERR_INTR_MASK_VC14_0

Offset: 0xe0e0
Byte Offset: 0x38380
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc14**
4	0x0	**err_intr_mask_ph_single_crc_err_vc14**
3	0x0	**err_intr_mask_pd_wc_short_err_vc14**
2	0x0	**err_intr_mask_pd_crc_err_vc14**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc14**
0	0x0	**err_intr_mask_ppfsm_timeout_vc14**

NVCSI_STREAM_5_ERR_INTR_MASK_VC15_0

Offset: 0xe0e1
Byte Offset: 0x38384
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_mask_embedded_line_crc_err_vc15**
4	0x0	**err_intr_mask_ph_single_crc_err_vc15**
3	0x0	**err_intr_mask_pd_wc_short_err_vc15**
2	0x0	**err_intr_mask_pd_crc_err_vc15**
1	0x0	**err_intr_mask_ph_ecc_single_bit_err_vc15**
0	0x0	**err_intr_mask_ppfsm_timeout_vc15**

NVCSI_STREAM_5_ERR_INTR_TYPE_NOVC_0

Offset: 0xe0e2
Byte Offset: 0x38388
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**err_intr_type_ph_both_crc_err**: packet header two CRC error for CPHY
0	0x0	**err_intr_type_ph_ecc_multi_bit_err**: packet header muti bit ecc error for DPHY

NVCSI_STREAM_5_ERR_INTR_TYPE_VC0_0

Offset: 0xe0e3
Byte Offset: 0x3838c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc0**
4	0x0	**err_intr_type_ph_single_crc_err_vc0**

Bit	Reset	Description
3	0x0	**err_intr_type_pd_wc_short_err_vc0**
2	0x0	**err_intr_type_pd_crc_err_vc0**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc0**
0	0x0	**err_intr_type_ppfsm_timeout_vc0**

NVCSI_STREAM_5_ERR_INTR_TYPE_VC1_0

Offset: 0xe0e4
Byte Offset: 0x38390
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc1**
4	0x0	**err_intr_type_ph_single_crc_err_vc1**
3	0x0	**err_intr_type_pd_wc_short_err_vc1**
2	0x0	**err_intr_type_pd_crc_err_vc1**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc1**
0	0x0	**err_intr_type_ppfsm_timeout_vc1**

NVCSI_STREAM_5_ERR_INTR_TYPE_VC2_0

Offset: 0xe0e5
Byte Offset: 0x38394
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc2**

Bit	Reset	Description
4	0x0	**err_intr_type_ph_single_crc_err_vc2**
3	0x0	**err_intr_type_pd_wc_short_err_vc2**
2	0x0	**err_intr_type_pd_crc_err_vc2**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc2**
0	0x0	**err_intr_type_ppfsm_timeout_vc2**

NVCSI_STREAM_5_ERR_INTR_TYPE_VC3_0

Offset: 0xe0e6
Byte Offset: 0x38398
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc3**
4	0x0	**err_intr_type_ph_single_crc_err_vc3**
3	0x0	**err_intr_type_pd_wc_short_err_vc3**
2	0x0	**err_intr_type_pd_crc_err_vc3**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc3**
0	0x0	**err_intr_type_ppfsm_timeout_vc3**

NVCSI_STREAM_5_ERR_INTR_TYPE_VC4_0

Offset: 0xe0e7
Byte Offset: 0x3839c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc4**
4	0x0	**err_intr_type_ph_single_crc_err_vc4**
3	0x0	**err_intr_type_pd_wc_short_err_vc4**
2	0x0	**err_intr_type_pd_crc_err_vc4**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc4**
0	0x0	**err_intr_type_ppfsm_timeout_vc4**

NVCSI_STREAM_5_ERR_INTR_TYPE_VC5_0

Offset: 0xe0e8
Byte Offset: 0x383a0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc5**
4	0x0	**err_intr_type_ph_single_crc_err_vc5**
3	0x0	**err_intr_type_pd_wc_short_err_vc5**
2	0x0	**err_intr_type_pd_crc_err_vc5**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc5**
0	0x0	**err_intr_type_ppfsm_timeout_vc5**

NVCSI_STREAM_5_ERR_INTR_TYPE_VC6_0

Offset: 0xe0e9
Byte Offset: 0x383a4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc6**
4	0x0	**err_intr_type_ph_single_crc_err_vc6**
3	0x0	**err_intr_type_pd_wc_short_err_vc6**
2	0x0	**err_intr_type_pd_crc_err_vc6**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc6**
0	0x0	**err_intr_type_ppfsm_timeout_vc6**

NVCSI_STREAM_5_ERR_INTR_TYPE_VC7_0

Offset: 0xe0ea
Byte Offset: 0x383a8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc7**
4	0x0	**err_intr_type_ph_single_crc_err_vc7**
3	0x0	**err_intr_type_pd_wc_short_err_vc7**
2	0x0	**err_intr_type_pd_crc_err_vc7**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc7**
0	0x0	**err_intr_type_ppfsm_timeout_vc7**

NVCSI_STREAM_5_ERR_INTR_TYPE_VC8_0

Offset: 0xe0eb
Byte Offset: 0x383ac
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc8**
4	0x0	**err_intr_type_ph_single_crc_err_vc8**
3	0x0	**err_intr_type_pd_wc_short_err_vc8**
2	0x0	**err_intr_type_pd_crc_err_vc8**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc8**
0	0x0	**err_intr_type_ppfsm_timeout_vc8**

NVCSI_STREAM_5_ERR_INTR_TYPE_VC9_0

Offset: 0xe0ec
Byte Offset: 0x383b0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc9**
4	0x0	**err_intr_type_ph_single_crc_err_vc9**
3	0x0	**err_intr_type_pd_wc_short_err_vc9**
2	0x0	**err_intr_type_pd_crc_err_vc9**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc9**
0	0x0	**err_intr_type_ppfsm_timeout_vc9**

NVCSI_STREAM_5_ERR_INTR_TYPE_VC10_0

Offset: 0xe0ed
Byte Offset: 0x383b4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc10**
4	0x0	**err_intr_type_ph_single_crc_err_vc10**
3	0x0	**err_intr_type_pd_wc_short_err_vc10**
2	0x0	**err_intr_type_pd_crc_err_vc10**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc10**
0	0x0	**err_intr_type_ppfsm_timeout_vc10**

NVCSI_STREAM_5_ERR_INTR_TYPE_VC11_0

Offset: 0xe0ee
Byte Offset: 0x383b8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc11**
4	0x0	**err_intr_type_ph_single_crc_err_vc11**
3	0x0	**err_intr_type_pd_wc_short_err_vc11**
2	0x0	**err_intr_type_pd_crc_err_vc11**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc11**
0	0x0	**err_intr_type_ppfsm_timeout_vc11**

NVCSI_STREAM_5_ERR_INTR_TYPE_VC12_0

Offset: 0xe0ef
Byte Offset: 0x383bc
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc12**
4	0x0	**err_intr_type_ph_single_crc_err_vc12**
3	0x0	**err_intr_type_pd_wc_short_err_vc12**
2	0x0	**err_intr_type_pd_crc_err_vc12**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc12**
0	0x0	**err_intr_type_ppfsm_timeout_vc12**

NVCSI_STREAM_5_ERR_INTR_TYPE_VC13_0

Offset: 0xe0f0
Byte Offset: 0x383c0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc13**
4	0x0	**err_intr_type_ph_single_crc_err_vc13**
3	0x0	**err_intr_type_pd_wc_short_err_vc13**
2	0x0	**err_intr_type_pd_crc_err_vc13**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc13**
0	0x0	**err_intr_type_ppfsm_timeout_vc13**

NVCSI_STREAM_5_ERR_INTR_TYPE_VC14_0

Offset: 0xe0f1
Byte Offset: 0x383c4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc14**
4	0x0	**err_intr_type_ph_single_crc_err_vc14**
3	0x0	**err_intr_type_pd_wc_short_err_vc14**
2	0x0	**err_intr_type_pd_crc_err_vc14**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc14**
0	0x0	**err_intr_type_ppfsm_timeout_vc14**

NVCSI_STREAM_5_ERR_INTR_TYPE_VC15_0

Offset: 0xe0f2
Byte Offset: 0x383c8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**err_intr_type_embedded_line_crc_err_vc15**
4	0x0	**err_intr_type_ph_single_crc_err_vc15**
3	0x0	**err_intr_type_pd_wc_short_err_vc15**
2	0x0	**err_intr_type_pd_crc_err_vc15**
1	0x0	**err_intr_type_ph_ecc_single_bit_err_vc15**
0	0x0	**err_intr_type_ppfsm_timeout_vc15**

NVCSI_STREAM_5_TPG_ENABLE_0

Offset: 0xe0f3
Byte Offset: 0x383cc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	DISABLE	**TPG_ENABLE**: Enable the TPG path **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_5_TPG_VC_ENABLE_0

Offset: 0xe0f4
Byte Offset: 0x383d0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15	DISABLE	**VC15_EN**: Enable bit for the VC15 **0** = DISABLE **1** = ENABLE
14	DISABLE	**VC14_EN**: Enable bit for the VC14 **0** = DISABLE **1** = ENABLE
13	DISABLE	**VC13_EN**: Enable bit for the VC13 **0** = DISABLE **1** = ENABLE
12	DISABLE	**VC12_EN**: Enable bit for the VC12 **0** = DISABLE **1** = ENABLE
11	DISABLE	**VC11_EN**: Enable bit for the VC11 **0** = DISABLE **1** = ENABLE
10	DISABLE	**VC10_EN**: Enable bit for the VC10 **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
9	DISABLE	**VC9_EN**: Enable bit for the VC9 **0** = DISABLE **1** = ENABLE
8	DISABLE	**VC8_EN**: Enable bit for the VC8 **0** = DISABLE **1** = ENABLE
7	DISABLE	**VC7_EN**: Enable bit for the VC7 **0** = DISABLE **1** = ENABLE
6	DISABLE	**VC6_EN**: Enable bit for the VC6 **0** = DISABLE **1** = ENABLE
5	DISABLE	**VC5_EN**: Enable bit for the VC5 **0** = DISABLE **1** = ENABLE
4	DISABLE	**VC4_EN**: Enable bit for the VC4 **0** = DISABLE **1** = ENABLE
3	DISABLE	**VC3_EN**: Enable bit for the VC3 **0** = DISABLE **1** = ENABLE
2	DISABLE	**VC2_EN**: Enable bit for the VC2 **0** = DISABLE **1** = ENABLE
1	DISABLE	**VC1_EN**: Enable bit for the VC1 **0** = DISABLE **1** = ENABLE
0	DISABLE	**VC0_EN**: Enable bit for the VC0 **0** = DISABLE **1** = ENABLE

NVCSI_STREAM_5_LOAD_TPG_CFG_0

Offset: 0xe0f5
Byte Offset: 0x383d4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**LOAD**: Load the shadow register

NVCSI_STREAM_5_TPG_CTRL_0

Offset: 0xe0f6
Byte Offset: 0x383d8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000008a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1000,1010)

Bit	Reset	Description
7:4	0x8	**DATA_SPEED**: Control the data generation speed, valid range is 1~8.
3	ENABLE	**SKIP_LS_LE_PKT**: If the LS/LE packet need to generated. **0** = DISABLE **1** = ENABLE
2	DISABLE	**OVERRIDE_CRC**: Override the packet header CRC and payload CRC. **0** = DISABLE **1** = ENABLE
1	CORE	**DEST**: The TPG pattern is send to PP or send to CIL for TX. **0** = CIL **1** = CORE

Bit	Reset	Description
0	DPHY	**PHY_MODE**: CPHY or DPHY packet structure for TPG **0** = DPHY **1** = CPHY

NVCSI_STREAM_5_TPG_VBLANK_0

Offset: 0xe0f7
Byte Offset: 0x383dc
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000800 (0b0000,0000,0000,0000,0000,1000,0000,0000)

Bit	Reset	Description
31:0	0x800	**VBLANK**: The vblank value, unit is nvcsi clock cycles.

NVCSI_STREAM_5_TPG_HBLANK_0

Offset: 0xe0f8
Byte Offset: 0x383e0
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000800 (0b0000,0000,0000,0000,0000,1000,0000,0000)

Bit	Reset	Description
31:0	0x800	**HBLANK**: The hblank value, unit is nvcsi clock cycles.

NVCSI_STREAM_5_TPG_STATUS_0

Offset: 0xe0f9
Byte Offset: 0x383e4
Read/Write: RO
Parity Protection: N
Shadow: N

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0001ffff (0bxxxx,xxxx,xxxx,xxx1,1111,1111,1111,1111)

Bit	Reset	Description
16	IDLE	**VC15_STATUS**: Indicate the VC15 status **0** = BUSY **1** = IDLE
15	IDLE	**VC14_STATUS**: Indicate the VC14 status **0** = BUSY **1** = IDLE
14	IDLE	**VC13_STATUS**: Indicate the VC13 status **0** = BUSY **1** = IDLE
13	IDLE	**VC12_STATUS**: Indicate the VC12 status **0** = BUSY **1** = IDLE
12	IDLE	**VC11_STATUS**: Indicate the VC11 status **0** = BUSY **1** = IDLE
11	IDLE	**VC10_STATUS**: Indicate the VC10 status **0** = BUSY **1** = IDLE
10	IDLE	**VC9_STATUS**: Indicate the VC9 status **0** = BUSY **1** = IDLE
9	IDLE	**VC8_STATUS**: Indicate the VC8 status **0** = BUSY **1** = IDLE
8	IDLE	**VC7_STATUS**: Indicate the VC7 status **0** = BUSY **1** = IDLE

Bit	Reset	Description
7	IDLE	**VC6_STATUS**: Indicate the VC6 status **0** = BUSY **1** = IDLE
6	IDLE	**VC5_STATUS**: Indicate the VC5 status **0** = BUSY **1** = IDLE
5	IDLE	**VC4_STATUS**: Indicate the VC4 status **0** = BUSY **1** = IDLE
4	IDLE	**VC3_STATUS**: Indicate the VC3 status **0** = BUSY **1** = IDLE
3	IDLE	**VC2_STATUS**: Indicate the VC2 status **0** = BUSY **1** = IDLE
2	IDLE	**VC1_STATUS**: Indicate the VC1 status **0** = BUSY **1** = IDLE
1	IDLE	**VC0_STATUS**: Indicate the VC0 status **0** = BUSY **1** = IDLE
0	IDLE	**STATUS**: Indicate the TPG is in idle state, all packet has been send and all VC are disabled. **0** = BUSY **1** = IDLE

NVCSI_STREAM_5_TPG_PH_ECC_0

Offset: 0xe0fa
Byte Offset: 0x383e8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:24	0x0	**LINE_ECC**: This field is for the long packet header ECC.
23:18	0x0	**EOL_ECC**: This field is for the EOL short packet ECC.
17:12	0x0	**SOL_ECC**: This field is for the SOL short packet ECC.
11:6	0x0	**EOF_ECC**: This field is for the EOF short packet ECC.
5:0	0x0	**SOF_ECC**: The TPG will not generate ECC for a packet. When using the TPG, SW should set the PP to skip the ecc check. To verify the ecc logic for safety BIST, SW can write a pre-calculated ECC for the TPG, when use with this mode, the TPG should generate a grescale pattern. This field is for the SOF short packet ECC.

NVCSI_STREAM_5_TPG_PF_CRC_0

Offset: 0xe0fb
Byte Offset: 0x383ec
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PF_CRC**: This field is for the long packet payload CRC override.

NVCSI_STREAM_5_TPG_PH_SOF_CRC_0

Offset: 0xe0fc
Byte Offset: 0x383f0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This field is for the CPHY SOF packet second packet header CRC override.
15:0	0x0	**CRC_0**: This field is for the CPHY SOF packet first packet header CRC override.

NVCSI_STREAM_5_TPG_PH_EOF_CRC_0

Offset: 0xe0fd
Byte Offset: 0x383f4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This field is for the CPHY EOF packet second packet header CRC override.
15:0	0x0	**CRC_0**: This field is for the CPHY EOF packet first packet header CRC override.

NVCSI_STREAM_5_TPG_PH_SOL_CRC_0

Offset: 0xe0fe
Byte Offset: 0x383f8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This field is for the CPHY SOL packet second packet header CRC override.
15:0	0x0	**CRC_0**: This field is for the CPHY SOL packet first packet header CRC override.

NVCSI_STREAM_5_TPG_PH_EOL_CRC_0

Offset: 0xe0ff
Byte Offset: 0x383fc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This field is for the CPHY EOL packet second packet header CRC override.
15:0	0x0	**CRC_0**: This field is for the CPHY EOL packet first packet header CRC override.

NVCSI_STREAM_5_TPG_PH_LONG_PKT_CRC_0

Offset: 0xe100
Byte Offset: 0x38400
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CRC_1**: This filed is for the CPHY long packet second packet header CRC override.
15:0	0x0	**CRC_0**: This filed is for the CPHY long packet first packet header CRC override.

NVCSI_STREAM_5_TPG_PKT_DELIMETER_0

Offset: 0xe101
Byte Offset: 0x38404
Read/Write: R/W
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0011)

Bit	Reset	Description
15:0	0x3	**NUM**: This field define the cycle number between two packet. Model the LP11 period between two packet, valid range is 1 to 65535.

NVCSI_STREAM_5_VC0_TPG_GAIN_CTRL_0

Offset: 0xe102
Byte Offset: 0x38408
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_5_VC1_TPG_GAIN_CTRL_0

Offset: 0xe103
Byte Offset: 0x3840c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_5_VC2_TPG_GAIN_CTRL_0

Offset: 0xe104
Byte Offset: 0x38410
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_5_VC3_TPG_GAIN_CTRL_0

Offset: 0xe105
Byte Offset: 0x38414
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_5_VC4_TPG_GAIN_CTRL_0

Offset: 0xe106
Byte Offset: 0x38418
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_5_VC5_TPG_GAIN_CTRL_0

Offset: 0xe107
Byte Offset: 0x3841c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_5_VC6_TPG_GAIN_CTRL_0

Offset: 0xe108
Byte Offset: 0x38420
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_5_VC7_TPG_GAIN_CTRL_0

Offset: 0xe109
Byte Offset: 0x38424
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_5_VC8_TPG_GAIN_CTRL_0

Offset: 0xe10a
Byte Offset: 0x38428
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_5_VC9_TPG_GAIN_CTRL_0

Offset: 0xe10b
Byte Offset: 0x3842c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_5_VC10_TPG_GAIN_CTRL_0

Offset: 0xe10c
Byte Offset: 0x38430
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_5_VC11_TPG_GAIN_CTRL_0

Offset: 0xe10d
Byte Offset: 0x38434
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_5_VC12_TPG_GAIN_CTRL_0

Offset: 0xe10e
Byte Offset: 0x38438
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_5_VC13_TPG_GAIN_CTRL_0

Offset: 0xe10f
Byte Offset: 0x3843c
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_5_VC14_TPG_GAIN_CTRL_0

Offset: 0xe110
Byte Offset: 0x38440
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_5_VC15_TPG_GAIN_CTRL_0

Offset: 0xe111
Byte Offset: 0x38444
Read/Write: R/W
Parity Protection: N
Shadow: Y
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x011)

Bit	Reset	Description
2:0	0x3	**GAIN_RATIO**: Gain ratio for this VC 0: 8:1 ratio 1: 4:1 ratio 2: 2:1 ratio 3: 1:1 ratio 4: 0.5:1 ratio 5: 0.25:1 ratio 6: 0.125:1 ratio

NVCSI_STREAM_5_SPARE_0
Offset: 0xe112
Byte Offset: 0x38448
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**stream_reg**

7.2.1.3.7 NVCSI Global Registers

NVCSI_CFG_NVCSI_INCR_SYNCPT_0

Offset: 0x0
Byte Offset: 0x0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:10	0x0	**NVCSI_COND**: Condition mapped from raise/wait **0** = IMMEDIATE **1** = OP_DONE **2** = RD_DONE **3** = REG_WR_SAFE **4** = ENGINE_IDLE **5** = COND_5 **6** = COND_6 **7** = COND_7 **8** = COND_8 **9** = COND_9 **10** = COND_10 **11** = COND_11 **12** = COND_12 **13** = COND_13 **14** = COND_14 **15** = COND_15 **16** = COND_16 **17** = COND_17 **18** = COND_18 **19** = COND_19 **20** = COND_20 **21** = COND_21 **22** = COND_22 **23** = COND_23 **24** = COND_24 **25** = COND_25 **26** = COND_26 **27** = COND_27 **28** = COND_28 **29** = COND_29 **30** = COND_30 **31** = COND_31 **32** = COND_32
9:0	0x0	**NVCSI_INDX**: syncpt index value

NVCSI_CFG_NVCSI_INCR_SYNCPT_CNTRL_0

Offset: 0x1
Byte Offset: 0x4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,xxxx,xxx0)

Bit	Reset	Description
8	0x0	**NVCSI_INCR_SYNCPT_NO_STALL**: If NO_STALL is 1, then when FIFOs are full, INCR_SYNCPT methods will be dropped and the INCR_SYNCPT_ERROR[COND] bit will be set. If NO_STALL is 0, then when FIFOs are full, the client host interface will be stalled.
0	0x0	**NVCSI_INCR_SYNCPT_SOFT_RESET**: If SOFT_RESET is set, then all internal state of the client syncpt block will be reset. To do soft reset, first set SOFT_RESET of all Host Controller clients affected, then clear all SOFT_RESETs.

NVCSI_CFG_NVCSI_INCR_SYNCPT_ERROR_0

Offset: 0x2
Byte Offset: 0x8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**NVCSI_COND_STATUS**: COND_STATUS[COND] is set if the FIFO for COND overflows. This bit is sticky and will remain set until cleared. Cleared by writing 1.

NVCSI_GLOBAL_NVCSI_ID_0

Offset: 0x20
Byte Offset: 0x80
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000030 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0011,0000)

Bit	Reset	Description
7:4	0x3	**ver_id**: ver_id update indicates a major IP change

Bit	Reset	Description
3:0	0x0	**rev_id**: rev_id update indicates a minor IP change

NVCSI_GLOBAL_DEBUG_CONTROL_0_0

Offset: 0x21
Byte Offset: 0x84
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,xxxx,x000)

Bit	Reset	Description
15:8	0x0	**debug_0_event_sel**: Value ranges from 0 to 148, Selects the corresponding event.
2:0	NO_STREAM	**debug_0_stream_sel**: 0x1 to 0x6 = Selects Stream 0 to Stream 5 **0** = NO_STREAM **1** = STREAM_0 **2** = STREAM_1 **3** = STREAM_2 **4** = STREAM_3 **5** = STREAM_4 **6** = STREAM_5

NVCSI_GLOBAL_DEBUG_CONTROL_1_0

Offset: 0x22
Byte Offset: 0x88
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,xxxx,x000)

Bit	Reset	Description
15:8	0x0	**debug_1_event_sel**: Value ranges from 0 to 148, Selects the corresponding event.

Bit	Reset	Description
2:0	NO_STREAM	**debug_1_stream_sel**: 0x1 to 0x6 = Selects Stream 0 to Stream 5 **0** = NO_STREAM **1** = STREAM_0 **2** = STREAM_1 **3** = STREAM_2 **4** = STREAM_3 **5** = STREAM_4 **6** = STREAM_5

NVCSI_GLOBAL_DEBUG_CONTROL_2_0

Offset: 0x23
Byte Offset: 0x8c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,xxxx,x000)

Bit	Reset	Description
15:8	0x0	**debug_2_event_sel**: Value ranges from 0 to 148, Selects the corresponding event.
2:0	NO_STREAM	**debug_2_stream_sel**: 0x1 to 0x6 = Selects Stream 0 to Stream 5 **0** = NO_STREAM **1** = STREAM_0 **2** = STREAM_1 **3** = STREAM_2 **4** = STREAM_3 **5** = STREAM_4 **6** = STREAM_5

NVCSI_GLOBAL_DEBUG_CONTROL_3_0

Offset: 0x24
Byte Offset: 0x90
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,xxxx,x000)

Bit	Reset	Description
15:8	0x0	**debug_3_event_sel**: Value ranges from 0 to 148, Selects the corresponding event.
2:0	NO_STREAM	**debug_3_stream_sel**: 0x1 to 0x6 = Selects Stream 0 to Stream 5 **0** = NO_STREAM **1** = STREAM_0 **2** = STREAM_1 **3** = STREAM_2 **4** = STREAM_3 **5** = STREAM_4 **6** = STREAM_5

NVCSI_GLOBAL_DEBUG_CONTROL_4_0

Offset: 0x25
Byte Offset: 0x94
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,xxxx,x000)

Bit	Reset	Description
15:8	0x0	**debug_4_event_sel**: Value ranges from 0 to 148, Selects the corresponding event.
2:0	NO_STREAM	**debug_4_stream_sel**: 0x1 to 0x6 = Selects Stream 0 to Stream 5 **0** = NO_STREAM **1** = STREAM_0 **2** = STREAM_1 **3** = STREAM_2 **4** = STREAM_3 **5** = STREAM_4 **6** = STREAM_5

NVCSI_GLOBAL_DEBUG_CONTROL_5_0

Offset: 0x26
Byte Offset: 0x98
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,xxxx,x000)

Bit	Reset	Description
15:8	0x0	**debug_5_event_sel**: Value ranges from 0 to 148, Selects the corresponding event.
2:0	NO_STREAM	**debug_5_stream_sel**: 0x1 to 0x6 = Selects Stream 0 to Stream 5 **0** = NO_STREAM **1** = STREAM_0 **2** = STREAM_1 **3** = STREAM_2 **4** = STREAM_3 **5** = STREAM_4 **6** = STREAM_5

NVCSI_GLOBAL_DEBUG_COUNTER_0_0

Offset: 0x27
Byte Offset: 0x9c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DEBUG_COUNTER_0**: 32 bit debug counter for counting stream related events for debug. Writing "0x1" to this register clears DEBUG_COUNTER_0

NVCSI_GLOBAL_DEBUG_COUNTER_1_0

Offset: 0x28
Byte Offset: 0xa0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DEBUG_COUNTER_1**: 32 bit debug counter for counting stream related events for debug. Writing "0x1" to this register clears DEBUG_COUNTER_1

NVCSI_GLOBAL_DEBUG_COUNTER_2_0

Offset: 0x29
Byte Offset: 0xa4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DEBUG_COUNTER_2**: 32 bit debug counter for counting stream related events for debug. Writing "0x1" to this register clears DEBUG_COUNTER_2

NVCSI_GLOBAL_DEBUG_COUNTER_3_0

Offset: 0x2a
Byte Offset: 0xa8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DEBUG_COUNTER_3**: 32 bit debug counter for counting stream related events for debug. Writing "0x1" to this register clears DEBUG_COUNTER_3

NVCSI_GLOBAL_DEBUG_COUNTER_4_0

Offset: 0x2b
Byte Offset: 0xac
Read/Write: R/W
Parity Protection: N
Shadow: N

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DEBUG_COUNTER_4**: 32 bit debug counter for counting stream related events for debug. Writing "0x1" to this register clears DEBUG_COUNTER_4

NVCSI_GLOBAL_DEBUG_COUNTER_5_0

Offset: 0x2c
Byte Offset: 0xb0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DEBUG_COUNTER_5**: 32 bit debug counter for counting stream related events for debug. Writing "0x1" to this register clears DEBUG_COUNTER_5

NVCSI_GLOBAL_HOST1X_CTRL_0

Offset: 0x2d
Byte Offset: 0xb4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x80000014 (0b1xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0100)

Bit	Reset	Description
31	0x1	**TIMEOUT_EN**: Enable the timeout counter for Host Controller read access.
5:0	0x14	**TIMEOUT_PERIOD**: Timeout threshold for Host Controller read access.

NVCSI_GLOBAL_INTR_PEND_0

Offset: 0x2e
Byte Offset: 0xb8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23	0x0	**sw_debug_intr_pend_status**: Interrupt from SW trigger pending.
22	0x0	**host1x_intr_pend_status**: Interrupt from Host Controller control pending.
21	0x0	**phy_3_cilb_intr_1_pend_status**: Interrupt from cil H status 1 pending.
20	0x0	**phy_3_cila_intr_1_pend_status**: Interrupt from cil G status 1 pending.
19	0x0	**phy_2_cilb_intr_1_pend_status**: Interrupt from cil F status 1 pending.
18	0x0	**phy_2_cila_intr_1_pend_status**: Interrupt from cil E status 1 pending.
17	0x0	**phy_1_cilb_intr_1_pend_status**: Interrupt from cil D status 1 pending.
16	0x0	**phy_1_cila_intr_1_pend_status**: Interrupt from cil C status 1 pending.
15	0x0	**phy_0_cilb_intr_1_pend_status**: Interrupt from cil B status 1 pending.
14	0x0	**phy_0_cila_intr_1_pend_status**: Interrupt from cil A status 1 pending.
13	0x0	**phy_3_cilb_intr_0_pend_status**: Interrupt from cil H status 0 pending.
12	0x0	**phy_3_cila_intr_0_pend_status**: Interrupt from cil G status 0 pending.
11	0x0	**phy_2_cilb_intr_0_pend_status**: Interrupt from cil F status 0 pending.
10	0x0	**phy_2_cila_intr_0_pend_status**: Interrupt from cil E status 0 pending.
9	0x0	**phy_1_cilb_intr_0_pend_status**: Interrupt from cil D status 0 pending.
8	0x0	**phy_1_cila_intr_0_pend_status**: Interrupt from cil C status 0 pending.

Bit	Reset	Description
7	0x0	**phy_0_cilb_intr_0_pend_status**: Interrupt from cil B status 0 pending.
6	0x0	**phy_0_cila_intr_0_pend_status**: Interrupt from cil A status 0 pending.
5	0x0	**stream_5_intr_pend_status**: Interrupt from stream 5 pending.
4	0x0	**stream_4_intr_pend_status**: Interrupt from stream 4 pending.
3	0x0	**stream_3_intr_pend_status**: Interrupt from stream 3 pending.
2	0x0	**stream_2_intr_pend_status**: Interrupt from stream 2 pending.
1	0x0	**stream_1_intr_pend_status**: Interrupt from stream 1 pending.
0	0x0	**stream_0_intr_pend_status**: Interrupt from stream 0 pending.

NVCSI_GLOBAL_CORRECTABLE_ERR_INTR_PEND_0

FAULT_UDF_TAG_NVCSI_OTHER_CORRECTABLE_ERROR Correctable Error Interrupt (to HSM) pending register
Offset: 0x2f
Byte Offset: 0xbc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**host1x_err_intr_pend_status**: Error status from Host Controller control pending.
13	0x0	**phy_3_cilb_err_intr_pend_status**: Error status from cil H pending.
12	0x0	**phy_3_cila_err_intr_pend_status**: Error status from cil G pending.
11	0x0	**phy_2_cilb_err_intr_pend_status**: Error status from cil F pending.
10	0x0	**phy_2_cila_err_intr_pend_status**: Error status from cil E pending.

Bit	Reset	Description
9	0x0	**phy_1_cilb_err_intr_pend_status**: Error status from cil D pending.
8	0x0	**phy_1_cila_err_intr_pend_status**: Error status from cil C pending.
7	0x0	**phy_0_cilb_err_intr_pend_status**: Error status from cil B pending.
6	0x0	**phy_0_cila_err_intr_pend_status**: Error status from cil A pending.
5	0x0	**stream_5_err_intr_pend_status**: Error status from stream 5 pending.
4	0x0	**stream_4_err_intr_pend_status**: Error status from stream 4 pending.
3	0x0	**stream_3_err_intr_pend_status**: Error status from stream 3 pending.
2	0x0	**stream_2_err_intr_pend_status**: Error status from stream 2 pending.
1	0x0	**stream_1_err_intr_pend_status**: Error status from stream 1 pending.
0	0x0	**stream_0_err_intr_pend_status**: Error status from stream 0 pending.

NVCSI_GLOBAL_UNCORRECTABLE_ERR_INTR_PEND_0

FAULT_UDF_TAG_NVCSI_OTHER_UNCORRECTABLE_ERROR Uncorrectable Error Interrupt (to HSM) pending register
Offset: 0x30
Byte Offset: 0xc0
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**host1x_err_intr_pend_status**: Error status from Host Controller control pending.
13	0x0	**phy_3_cilb_err_intr_pend_status**: Error status from cil H pending.
12	0x0	**phy_3_cila_err_intr_pend_status**: Error status from cil G pending.

Bit	Reset	Description
11	0x0	**phy_2_cilb_err_intr_pend_status**: Error status from cil F pending.
10	0x0	**phy_2_cila_err_intr_pend_status**: Error status from cil E pending.
9	0x0	**phy_1_cilb_err_intr_pend_status**: Error status from cil D pending.
8	0x0	**phy_1_cila_err_intr_pend_status**: Error status from cil C pending.
7	0x0	**phy_0_cilb_err_intr_pend_status**: Error status from cil B pending.
6	0x0	**phy_0_cila_err_intr_pend_status**: Error status from cil A pending.
5	0x0	**stream_5_err_intr_pend_status**: Error status from stream 5 pending.
4	0x0	**stream_4_err_intr_pend_status**: Error status from stream 4 pending.
3	0x0	**stream_3_err_intr_pend_status**: Error status from stream 3 pending.
2	0x0	**stream_2_err_intr_pend_status**: Error status from stream 2 pending.
1	0x0	**stream_1_err_intr_pend_status**: Error status from stream 1 pending.
0	0x0	**stream_0_err_intr_pend_status**: Error status from stream 0 pending.

NVCSI_GLOBAL_INTR_STATUS_HOST1X_0

Offset: 0x31
Byte Offset: 0xc4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**intr_host1x_timeout_err**: Host Controller read access timeout

NVCSI_GLOBAL_INTR_MASK_HOST1X_0

Offset: 0x32
Byte Offset: 0xc8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**intr_mask_host1x_timeout_err**: Mask the Host Controller read access timeout error, when set to 1, the interrrupt will not send out.

NVCSI_GLOBAL_CORRECTABLE_ERR_INTR_STATUS_HOST1X_0

Offset: 0x33
Byte Offset: 0xcc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**err_intr_host1x_timeout_err**: Host Controller read access timeout

NVCSI_GLOBAL_UNCORRECTABLE_ERR_INTR_STATUS_HOST1X_0

Offset: 0x34
Byte Offset: 0xd0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**err_intr_host1x_timeout_err**: Host Controller read access timeout

NVCSI_GLOBAL_ERR_INTR_MASK_HOST1X_0

Offset: 0x35
Byte Offset: 0xd4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**err_intr_mask_host1x_timeout_err**: Mask the Host Controller read access timeout error, when set to 1, the error will not send out.

NVCSI_GLOBAL_ERR_INTR_TYPE_HOST1X_0

Offset: 0x36
Byte Offset: 0xd8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**err_intr_type_host1x_timeout_err**: The error type for the Host Controller read access timeout send out. 0: corrected error. 1: uncorrected error.

NVCSI_GLOBAL_SW_DEBUG_INTR_STATUS_0

Offset: 0x38
Byte Offset: 0xe0

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SW_DEBUG_INTR**: SW trigger interrupt status

NVCSI_GLOBAL_SW_DEBUG_INTR_TRIG_0

Offset: 0x39
Byte Offset: 0xe4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRIGGER**: Write 1 to this register trigger a SW interrupt

NVCSI_GLOBAL_SW_ERR_CODE_0

SW Error Reporting

In certain cases, the IP driver may implement diagnostics in SW.

Errors detected by such SW diagnostics need to be reported to HSM. To avoid time consuming IPCs, IP SW can directly write an IP SW defined Error Code into the SW_ERR_CODE register and assert the SW Error. The IP Error Collator adds an additional error input with the below options : err_type = SW err_user - Adds err__user[31:0] as an input to the error collator for the err_x.

Offset: 0x3a
Byte Offset: 0xe8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CODE**: SW Defined Error Code. Bits from this register connect to the IPs Error Collator err_<x>_user[31:0] signal.

NVCSI_GLOBAL_SW_ERR_ASSERT_0

Offset: 0x3b
Byte Offset: 0xec
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SW_ASSERT**: When written with value 1'b1, this self clearing bit generates a pulse of 1 clock cycle from the unit. This pulse is dual_rail_converted by the IP {bit,~bit} and connected to the Error Collator err_x

NVCSI_GLOBAL_SPARE0_0

Offset: 0x3c
Byte Offset: 0xf0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**global_reg**

NVCSI_GLOBAL_SPARE1_0

Offset: 0x3d
Byte Offset: 0xf4
Read/Write: R/W
Parity Protection: N
Shadow: N

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**global_reg**

NVCSI_GLOBAL_SPARE2_0
Offset: 0x3e
Byte Offset: 0xf8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**global_reg**

NVCSI_GLOBAL_SPARE3_0
Offset: 0x3f
Byte Offset: 0xfc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**global_reg**

NVCSI_GLOBAL_TPG_CTRL_VC0_0

Offset: 0x40
Byte Offset: 0x100
Read/Write: R/W
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000008 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1000)

Bit	Reset	Description
3	ENABLE	**PHASE_INC**: If the is field is set, the initial phase will increase 512 by frame. **0** = DISABLE **1** = ENABLE
2	DISABLE	**AUTO_STOP**: If the auto stop is set, this virtual channel will automaticly stop when the frame count to the TPG_FN_VCx.MAX_FN. **0** = DISABLE **1** = ENABLE
1:0	DIRECT	**MODE**: Pattern mode **0** = DIRECT **1** = PATCH **2** = SINE **3** = RSVD

NVCSI_GLOBAL_TPG_EMB_CTRL_VC0_0

Offset: 0x41
Byte Offset: 0x104
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01000000 (0bxxxx,xxx1,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
24	0x1	**PATTERN_MODE**: The embbed line payload.
23:20	0x0	**TOP_LINES**: The embbed line number before the frame.
19:16	0x0	**BOTTOM_LINES**: The embbed line number after the frame.
15:0	0x0	**WIDTH**: Embedded line data width, unit is byte.

NVCSI_GLOBAL_TPG_SIZE_VC0_0

Offset: 0x42
Byte Offset: 0x108
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00400040 (0b0000,0000,0100,0000,0000,0000,0100,0000)

Bit	Reset	Description
31:16	0x40	**Y_SIZE**: the frame virtical size
15:0	0x40	**X_SIZE**: The frame horizontal size

NVCSI_GLOBAL_TPG_FN_VC0_0

Offset: 0x43
Byte Offset: 0x10c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xffff0000 (0b1111,1111,1111,1111,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0xffff	**MAX_FN**: The max frame ID
15:0	0x0	**MIN_FN**: The min frame ID

NVCSI_GLOBAL_TPG_DT_VC0_0

Offset: 0x44
Byte Offset: 0x110
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000024 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,0100)

Bit	Reset	Description
5:0	0x24	**DT**: Data type YUV_420_8: 0x18 YUV_420_10: 0x19 YUV_420_8_Legacy: 0x1a YUV_420_8_CSPS: 0x1c YUV_420_10_CSPS: 0x1d YUV_422_8: 0x1e YUV_422_10: 0x1f RGB444: 0x20 RGB555: 0x21 RGB565: 0x22 RGB666: 0x23 RGB888: 0x24 RAW6: 0x28 RAW7: 0x29 RAW8: 0x2a RAW10: 0x2b RAW12: 0x2c RAW14: 0x2d RAW16: 0x2e RAW20: 0x2f

NVCSI_GLOBAL_TPG_RED_PHASE_VC0_0

Offset: 0x45
Byte Offset: 0x114
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_GREEN_PHASE_VC0_0

Offset: 0x46
Byte Offset: 0x118
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_BLUE_PHASE_VC0_0

Offset: 0x47
Byte Offset: 0x11c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_RED_FREQ_VC0_0

Offset: 0x48
Byte Offset: 0x120
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_GREEN_FREQ_VC0_0

Offset: 0x49
Byte Offset: 0x124
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_BLUE_FREQ_VC0_0

Offset: 0x4a
Byte Offset: 0x128
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_RED_FREQ_RATE_VC0_0

Offset: 0x4b
Byte Offset: 0x12c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_GREEN_FREQ_RATE_VC0_0

Offset: 0x4c
Byte Offset: 0x130

Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_BLUE_FREQ_RATE_VC0_0

Offset: 0x4d
Byte Offset: 0x134
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_SPARE0_VC0_0

Offset: 0x4e
Byte Offset: 0x138
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_SPARE1_VC0_0

Offset: 0x4f
Byte Offset: 0x13c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_CTRL_VC1_0

Offset: 0x50
Byte Offset: 0x140
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000008 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1000)

Bit	Reset	Description
3	ENABLE	**PHASE_INC**: If the is field is set, the initial phase will increase 512 by frame. **0** = DISABLE **1** = ENABLE
2	DISABLE	**AUTO_STOP**: If the auto stop is set, this virtual channel will automaticly stop when the frame count to the TPG_FN_VCx.MAX_FN. **0** = DISABLE **1** = ENABLE
1:0	DIRECT	**MODE**: Pattern mode **0** = DIRECT **1** = PATCH **2** = SINE **3** = RSVD

NVCSI_GLOBAL_TPG_EMB_CTRL_VC1_0

Offset: 0x51

Byte Offset: 0x144
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01000000 (0bxxxx,xxx1,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
24	0x1	**PATTERN_MODE**: The embbed line payload.
23:20	0x0	**TOP_LINES**: The embbed line number before the frame.
19:16	0x0	**BOTTOM_LINES**: The embbed line number after the frame.
15:0	0x0	**WIDTH**: Embedded line data width, unit is byte.

NVCSI_GLOBAL_TPG_SIZE_VC1_0

Offset: 0x52
Byte Offset: 0x148
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00400040 (0b0000,0000,0100,0000,0000,0000,0100,0000)

Bit	Reset	Description
31:16	0x40	**Y_SIZE**: the frame virtical size
15:0	0x40	**X_SIZE**: The frame horizontal size

NVCSI_GLOBAL_TPG_FN_VC1_0

Offset: 0x53
Byte Offset: 0x14c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0xffff0000 (0b1111,1111,1111,1111,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0xffff	**MAX_FN**: The max frame ID
15:0	0x0	**MIN_FN**: The min frame ID

NVCSI_GLOBAL_TPG_DT_VC1_0

Offset: 0x54
Byte Offset: 0x150
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000024 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,0100)

Bit	Reset	Description
5:0	0x24	**DT**: Data type YUV_420_8: 0x18 YUV_420_10: 0x19 YUV_420_8_Legacy: 0x1a YUV_420_8_CSPS: 0x1c YUV_420_10_CSPS: 0x1d YUV_422_8: 0x1e YUV_422_10: 0x1f RGB444: 0x20 RGB555: 0x21 RGB565: 0x22 RGB666: 0x23 RGB888: 0x24 RAW6: 0x28 RAW7: 0x29 RAW8: 0x2a RAW10: 0x2b RAW12: 0x2c RAW14: 0x2d RAW16: 0x2e RAW20: 0x2f

NVCSI_GLOBAL_TPG_RED_PHASE_VC1_0

Offset: 0x55
Byte Offset: 0x154
Read/Write: R/W
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_GREEN_PHASE_VC1_0

Offset: 0x56
Byte Offset: 0x158
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_BLUE_PHASE_VC1_0

Offset: 0x57
Byte Offset: 0x15c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_RED_FREQ_VC1_0

Offset: 0x58
Byte Offset: 0x160

Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_GREEN_FREQ_VC1_0

Offset: 0x59
Byte Offset: 0x164
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_BLUE_FREQ_VC1_0

Offset: 0x5a
Byte Offset: 0x168
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_RED_FREQ_RATE_VC1_0

Offset: 0x5b
Byte Offset: 0x16c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_GREEN_FREQ_RATE_VC1_0

Offset: 0x5c
Byte Offset: 0x170
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_BLUE_FREQ_RATE_VC1_0

Offset: 0x5d
Byte Offset: 0x174
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_SPARE0_VC1_0

Offset: 0x5e
Byte Offset: 0x178
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_SPARE1_VC1_0

Offset: 0x5f
Byte Offset: 0x17c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_CTRL_VC2_0

Offset: 0x60
Byte Offset: 0x180
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000008 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1000)

Bit	Reset	Description
3	ENABLE	**PHASE_INC**: If the is field is set, the initial phase will increase 512 by frame. **0** = DISABLE **1** = ENABLE
2	DISABLE	**AUTO_STOP**: If the auto stop is set, this virtual channel will automaticly stop when the frame count to the TPG_FN_VCx.MAX_FN. **0** = DISABLE **1** = ENABLE
1:0	DIRECT	**MODE**: Pattern mode **0** = DIRECT **1** = PATCH **2** = SINE **3** = RSVD

NVCSI_GLOBAL_TPG_EMB_CTRL_VC2_0

Offset: 0x61
Byte Offset: 0x184
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01000000 (0bxxxx,xxx1,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
24	0x1	**PATTERN_MODE**: The embbed line payload.
23:20	0x0	**TOP_LINES**: The embbed line number before the frame.
19:16	0x0	**BOTTOM_LINES**: The embbed line number after the frame.
15:0	0x0	**WIDTH**: Embedded line data width, unit is byte.

NVCSI_GLOBAL_TPG_SIZE_VC2_0

Offset: 0x62

Byte Offset: 0x188
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00400040 (0b0000,0000,0100,0000,0000,0000,0100,0000)

Bit	Reset	Description
31:16	0x40	**Y_SIZE**: the frame virtical size
15:0	0x40	**X_SIZE**: The frame horizontal size

NVCSI_GLOBAL_TPG_FN_VC2_0
Offset: 0x63
Byte Offset: 0x18c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xffff0000 (0b1111,1111,1111,1111,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0xffff	**MAX_FN**: The max frame ID
15:0	0x0	**MIN_FN**: The min frame ID

NVCSI_GLOBAL_TPG_DT_VC2_0
Offset: 0x64
Byte Offset: 0x190
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000024 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,0100)

Bit	Reset	Description
5:0	0x24	**DT**: Data type YUV_420_8: 0x18 YUV_420_10: 0x19 YUV_420_8_Legacy: 0x1a YUV_420_8_CSPS: 0x1c YUV_420_10_CSPS: 0x1d YUV_422_8: 0x1e YUV_422_10: 0x1f RGB444: 0x20 RGB555: 0x21 RGB565: 0x22 RGB666: 0x23 RGB888: 0x24 RAW6: 0x28 RAW7: 0x29 RAW8: 0x2a RAW10: 0x2b RAW12: 0x2c RAW14: 0x2d RAW16: 0x2e RAW20: 0x2f

NVCSI_GLOBAL_TPG_RED_PHASE_VC2_0

Offset: 0x65
Byte Offset: 0x194
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_GREEN_PHASE_VC2_0

Offset: 0x66
Byte Offset: 0x198
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_BLUE_PHASE_VC2_0

Offset: 0x67
Byte Offset: 0x19c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_RED_FREQ_VC2_0

Offset: 0x68
Byte Offset: 0x1a0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_GREEN_FREQ_VC2_0

Offset: 0x69
Byte Offset: 0x1a4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_BLUE_FREQ_VC2_0

Offset: 0x6a
Byte Offset: 0x1a8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_RED_FREQ_RATE_VC2_0

Offset: 0x6b
Byte Offset: 0x1ac
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_GREEN_FREQ_RATE_VC2_0

Offset: 0x6c
Byte Offset: 0x1b0

Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_BLUE_FREQ_RATE_VC2_0

Offset: 0x6d
Byte Offset: 0x1b4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_SPARE0_VC2_0

Offset: 0x6e
Byte Offset: 0x1b8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_SPARE1_VC2_0

Offset: 0x6f
Byte Offset: 0x1bc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_CTRL_VC3_0

Offset: 0x70
Byte Offset: 0x1c0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000008 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1000)

Bit	Reset	Description
3	ENABLE	**PHASE_INC**: If the is field is set, the initial phase will increase 512 by frame. **0** = DISABLE **1** = ENABLE
2	DISABLE	**AUTO_STOP**: If the auto stop is set, this virtual channel will automaticly stop when the frame count to the TPG_FN_VCx.MAX_FN. **0** = DISABLE **1** = ENABLE
1:0	DIRECT	**MODE**: Pattern mode **0** = DIRECT **1** = PATCH **2** = SINE **3** = RSVD

NVCSI_GLOBAL_TPG_EMB_CTRL_VC3_0

Offset: 0x71

Byte Offset: 0x1c4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01000000 (0bxxxx,xxx1,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
24	0x1	**PATTERN_MODE**: The embbed line payload.
23:20	0x0	**TOP_LINES**: The embbed line number before the frame.
19:16	0x0	**BOTTOM_LINES**: The embbed line number after the frame.
15:0	0x0	**WIDTH**: Embedded line data width, unit is byte.

NVCSI_GLOBAL_TPG_SIZE_VC3_0

Offset: 0x72
Byte Offset: 0x1c8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00400040 (0b0000,0000,0100,0000,0000,0000,0100,0000)

Bit	Reset	Description
31:16	0x40	**Y_SIZE**: the frame virtical size
15:0	0x40	**X_SIZE**: The frame horizontal size

NVCSI_GLOBAL_TPG_FN_VC3_0

Offset: 0x73
Byte Offset: 0x1cc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0xffff0000 (0b1111,1111,1111,1111,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0xffff	**MAX_FN**: The max frame ID
15:0	0x0	**MIN_FN**: The min frame ID

NVCSI_GLOBAL_TPG_DT_VC3_0

Offset: 0x74
Byte Offset: 0x1d0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000024 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,0100)

Bit	Reset	Description
5:0	0x24	**DT**: Data type YUV_420_8: 0x18 YUV_420_10: 0x19 YUV_420_8_Legacy: 0x1a YUV_420_8_CSPS: 0x1c YUV_420_10_CSPS: 0x1d YUV_422_8: 0x1e YUV_422_10: 0x1f RGB444: 0x20 RGB555: 0x21 RGB565: 0x22 RGB666: 0x23 RGB888: 0x24 RAW6: 0x28 RAW7: 0x29 RAW8: 0x2a RAW10: 0x2b RAW12: 0x2c RAW14: 0x2d RAW16: 0x2e RAW20: 0x2f

NVCSI_GLOBAL_TPG_RED_PHASE_VC3_0

Offset: 0x75
Byte Offset: 0x1d4
Read/Write: R/W
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_GREEN_PHASE_VC3_0

Offset: 0x76
Byte Offset: 0x1d8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_BLUE_PHASE_VC3_0

Offset: 0x77
Byte Offset: 0x1dc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_RED_FREQ_VC3_0

Offset: 0x78
Byte Offset: 0x1e0

Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_GREEN_FREQ_VC3_0

Offset: 0x79
Byte Offset: 0x1e4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_BLUE_FREQ_VC3_0

Offset: 0x7a
Byte Offset: 0x1e8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_RED_FREQ_RATE_VC3_0

Offset: 0x7b
Byte Offset: 0x1ec
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_GREEN_FREQ_RATE_VC3_0

Offset: 0x7c
Byte Offset: 0x1f0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_BLUE_FREQ_RATE_VC3_0

Offset: 0x7d
Byte Offset: 0x1f4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_SPARE0_VC3_0

Offset: 0x7e
Byte Offset: 0x1f8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_SPARE1_VC3_0

Offset: 0x7f
Byte Offset: 0x1fc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_CTRL_VC4_0

Offset: 0x80
Byte Offset: 0x200
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000008 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1000)

Bit	Reset	Description
3	ENABLE	**PHASE_INC**: If the is field is set, the initial phase will increase 512 by frame. **0** = DISABLE **1** = ENABLE
2	DISABLE	**AUTO_STOP**: If the auto stop is set, this virtual channel will automaticly stop when the frame count to the TPG_FN_VCx.MAX_FN. **0** = DISABLE **1** = ENABLE
1:0	DIRECT	**MODE**: Pattern mode **0** = DIRECT **1** = PATCH **2** = SINE **3** = RSVD

NVCSI_GLOBAL_TPG_EMB_CTRL_VC4_0

Offset: 0x81
Byte Offset: 0x204
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01000000 (0bxxxx,xxx1,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
24	0x1	**PATTERN_MODE**: The embbed line payload.
23:20	0x0	**TOP_LINES**: The embbed line number before the frame.
19:16	0x0	**BOTTOM_LINES**: The embbed line number after the frame.
15:0	0x0	**WIDTH**: Embedded line data width, unit is byte.

NVCSI_GLOBAL_TPG_SIZE_VC4_0

Offset: 0x82

Byte Offset: 0x208
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00400040 (0b0000,0000,0100,0000,0000,0000,0100,0000)

Bit	Reset	Description
31:16	0x40	**Y_SIZE**: the frame virtical size
15:0	0x40	**X_SIZE**: The frame horizontal size

NVCSI_GLOBAL_TPG_FN_VC4_0
Offset: 0x83
Byte Offset: 0x20c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xffff0000 (0b1111,1111,1111,1111,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0xffff	**MAX_FN**: The max frame ID
15:0	0x0	**MIN_FN**: The min frame ID

NVCSI_GLOBAL_TPG_DT_VC4_0
Offset: 0x84
Byte Offset: 0x210
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000024 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,0100)

Bit	Reset	Description
5:0	0x24	**DT**: Data type YUV_420_8: 0x18 YUV_420_10: 0x19 YUV_420_8_Legacy: 0x1a YUV_420_8_CSPS: 0x1c YUV_420_10_CSPS: 0x1d YUV_422_8: 0x1e YUV_422_10: 0x1f RGB444: 0x20 RGB555: 0x21 RGB565: 0x22 RGB666: 0x23 RGB888: 0x24 RAW6: 0x28 RAW7: 0x29 RAW8: 0x2a RAW10: 0x2b RAW12: 0x2c RAW14: 0x2d RAW16: 0x2e RAW20: 0x2f

NVCSI_GLOBAL_TPG_RED_PHASE_VC4_0

Offset: 0x85
Byte Offset: 0x214
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_GREEN_PHASE_VC4_0

Offset: 0x86
Byte Offset: 0x218
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_BLUE_PHASE_VC4_0

Offset: 0x87
Byte Offset: 0x21c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_RED_FREQ_VC4_0

Offset: 0x88
Byte Offset: 0x220
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_GREEN_FREQ_VC4_0

Offset: 0x89
Byte Offset: 0x224
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_BLUE_FREQ_VC4_0

Offset: 0x8a
Byte Offset: 0x228
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_RED_FREQ_RATE_VC4_0

Offset: 0x8b
Byte Offset: 0x22c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_GREEN_FREQ_RATE_VC4_0

Offset: 0x8c
Byte Offset: 0x230

Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_BLUE_FREQ_RATE_VC4_0

Offset: 0x8d
Byte Offset: 0x234
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_SPARE0_VC4_0

Offset: 0x8e
Byte Offset: 0x238
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_SPARE1_VC4_0

Offset: 0x8f
Byte Offset: 0x23c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_CTRL_VC5_0

Offset: 0x90
Byte Offset: 0x240
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000008 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1000)

Bit	Reset	Description
3	ENABLE	**PHASE_INC**: If the is field is set, the initial phase will increase 512 by frame. **0** = DISABLE **1** = ENABLE
2	DISABLE	**AUTO_STOP**: If the auto stop is set, this virtual channel will automaticly stop when the frame count to the TPG_FN_VCx.MAX_FN. **0** = DISABLE **1** = ENABLE
1:0	DIRECT	**MODE**: Pattern mode **0** = DIRECT **1** = PATCH **2** = SINE **3** = RSVD

NVCSI_GLOBAL_TPG_EMB_CTRL_VC5_0

Offset: 0x91

Byte Offset: 0x244
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01000000 (0bxxxx,xxx1,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
24	0x1	**PATTERN_MODE**: The embbed line payload.
23:20	0x0	**TOP_LINES**: The embbed line number before the frame.
19:16	0x0	**BOTTOM_LINES**: The embbed line number after the frame.
15:0	0x0	**WIDTH**: Embedded line data width, unit is byte.

NVCSI_GLOBAL_TPG_SIZE_VC5_0

Offset: 0x92
Byte Offset: 0x248
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00400040 (0b0000,0000,0100,0000,0000,0000,0100,0000)

Bit	Reset	Description
31:16	0x40	**Y_SIZE**: the frame virtical size
15:0	0x40	**X_SIZE**: The frame horizontal size

NVCSI_GLOBAL_TPG_FN_VC5_0

Offset: 0x93
Byte Offset: 0x24c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0xffff0000 (0b1111,1111,1111,1111,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0xffff	**MAX_FN**: The max frame ID
15:0	0x0	**MIN_FN**: The min frame ID

NVCSI_GLOBAL_TPG_DT_VC5_0

Offset: 0x94
Byte Offset: 0x250
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000024 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,0100)

Bit	Reset	Description
5:0	0x24	**DT**: Data type YUV_420_8: 0x18 YUV_420_10: 0x19 YUV_420_8_Legacy: 0x1a YUV_420_8_CSPS: 0x1c YUV_420_10_CSPS: 0x1d YUV_422_8: 0x1e YUV_422_10: 0x1f RGB444: 0x20 RGB555: 0x21 RGB565: 0x22 RGB666: 0x23 RGB888: 0x24 RAW6: 0x28 RAW7: 0x29 RAW8: 0x2a RAW10: 0x2b RAW12: 0x2c RAW14: 0x2d RAW16: 0x2e RAW20: 0x2f

NVCSI_GLOBAL_TPG_RED_PHASE_VC5_0

Offset: 0x95
Byte Offset: 0x254
Read/Write: R/W
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_GREEN_PHASE_VC5_0

Offset: 0x96
Byte Offset: 0x258
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_BLUE_PHASE_VC5_0

Offset: 0x97
Byte Offset: 0x25c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_RED_FREQ_VC5_0

Offset: 0x98
Byte Offset: 0x260

Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_GREEN_FREQ_VC5_0

Offset: 0x99
Byte Offset: 0x264
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_BLUE_FREQ_VC5_0

Offset: 0x9a
Byte Offset: 0x268
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_RED_FREQ_RATE_VC5_0

Offset: 0x9b
Byte Offset: 0x26c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_GREEN_FREQ_RATE_VC5_0

Offset: 0x9c
Byte Offset: 0x270
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_BLUE_FREQ_RATE_VC5_0

Offset: 0x9d
Byte Offset: 0x274
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_SPARE0_VC5_0
Offset: 0x9e
Byte Offset: 0x278
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_SPARE1_VC5_0
Offset: 0x9f
Byte Offset: 0x27c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_CTRL_VC6_0
Offset: 0xa0
Byte Offset: 0x280
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000008 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1000)

Bit	Reset	Description
3	ENABLE	**PHASE_INC**: If the is field is set, the initial phase will increase 512 by frame. **0** = DISABLE **1** = ENABLE
2	DISABLE	**AUTO_STOP**: If the auto stop is set, this virtual channel will automaticly stop when the frame count to the TPG_FN_VCx.MAX_FN. **0** = DISABLE **1** = ENABLE
1:0	DIRECT	**MODE**: Pattern mode **0** = DIRECT **1** = PATCH **2** = SINE **3** = RSVD

NVCSI_GLOBAL_TPG_EMB_CTRL_VC6_0

Offset: 0xa1
Byte Offset: 0x284
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01000000 (0bxxxx,xxx1,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
24	0x1	**PATTERN_MODE**: The embbed line payload.
23:20	0x0	**TOP_LINES**: The embbed line number before the frame.
19:16	0x0	**BOTTOM_LINES**: The embbed line number after the frame.
15:0	0x0	**WIDTH**: Embedded line data width, unit is byte.

NVCSI_GLOBAL_TPG_SIZE_VC6_0

Offset: 0xa2

Byte Offset: 0x288
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00400040 (0b0000,0000,0100,0000,0000,0000,0100,0000)

Bit	Reset	Description
31:16	0x40	**Y_SIZE**: the frame virtical size
15:0	0x40	**X_SIZE**: The frame horizontal size

NVCSI_GLOBAL_TPG_FN_VC6_0
Offset: 0xa3
Byte Offset: 0x28c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xffff0000 (0b1111,1111,1111,1111,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0xffff	**MAX_FN**: The max frame ID
15:0	0x0	**MIN_FN**: The min frame ID

NVCSI_GLOBAL_TPG_DT_VC6_0
Offset: 0xa4
Byte Offset: 0x290
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000024 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,0100)

Bit	Reset	Description
5:0	0x24	**DT**: Data type YUV_420_8: 0x18 YUV_420_10: 0x19 YUV_420_8_Legacy: 0x1a YUV_420_8_CSPS: 0x1c YUV_420_10_CSPS: 0x1d YUV_422_8: 0x1e YUV_422_10: 0x1f RGB444: 0x20 RGB555: 0x21 RGB565: 0x22 RGB666: 0x23 RGB888: 0x24 RAW6: 0x28 RAW7: 0x29 RAW8: 0x2a RAW10: 0x2b RAW12: 0x2c RAW14: 0x2d RAW16: 0x2e RAW20: 0x2f

NVCSI_GLOBAL_TPG_RED_PHASE_VC6_0

Offset: 0xa5
Byte Offset: 0x294
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_GREEN_PHASE_VC6_0

Offset: 0xa6
Byte Offset: 0x298
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_BLUE_PHASE_VC6_0

Offset: 0xa7
Byte Offset: 0x29c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_RED_FREQ_VC6_0

Offset: 0xa8
Byte Offset: 0x2a0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_GREEN_FREQ_VC6_0

Offset: 0xa9
Byte Offset: 0x2a4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_BLUE_FREQ_VC6_0

Offset: 0xaa
Byte Offset: 0x2a8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_RED_FREQ_RATE_VC6_0

Offset: 0xab
Byte Offset: 0x2ac
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_GREEN_FREQ_RATE_VC6_0

Offset: 0xac
Byte Offset: 0x2b0

Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_BLUE_FREQ_RATE_VC6_0

Offset: 0xad
Byte Offset: 0x2b4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_SPARE0_VC6_0

Offset: 0xae
Byte Offset: 0x2b8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_SPARE1_VC6_0

Offset: 0xaf
Byte Offset: 0x2bc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_CTRL_VC7_0

Offset: 0xb0
Byte Offset: 0x2c0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000008 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1000)

Bit	Reset	Description
3	ENABLE	**PHASE_INC**: If the is field is set, the initial phase will increase 512 by frame. **0** = DISABLE **1** = ENABLE
2	DISABLE	**AUTO_STOP**: If the auto stop is set, this virtual channel will automaticly stop when the frame count to the TPG_FN_VCx.MAX_FN. **0** = DISABLE **1** = ENABLE
1:0	DIRECT	**MODE**: Pattern mode **0** = DIRECT **1** = PATCH **2** = SINE **3** = RSVD

NVCSI_GLOBAL_TPG_EMB_CTRL_VC7_0

Offset: 0xb1

Byte Offset: 0x2c4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01000000 (0bxxxx,xxx1,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
24	0x1	**PATTERN_MODE**: The embbed line payload.
23:20	0x0	**TOP_LINES**: The embbed line number before the frame.
19:16	0x0	**BOTTOM_LINES**: The embbed line number after the frame.
15:0	0x0	**WIDTH**: Embedded line data width, unit is byte.

NVCSI_GLOBAL_TPG_SIZE_VC7_0

Offset: 0xb2
Byte Offset: 0x2c8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00400040 (0b0000,0000,0100,0000,0000,0000,0100,0000)

Bit	Reset	Description
31:16	0x40	**Y_SIZE**: the frame virtical size
15:0	0x40	**X_SIZE**: The frame horizontal size

NVCSI_GLOBAL_TPG_FN_VC7_0

Offset: 0xb3
Byte Offset: 0x2cc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0xffff0000 (0b1111,1111,1111,1111,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0xffff	**MAX_FN**: The max frame ID
15:0	0x0	**MIN_FN**: The min frame ID

NVCSI_GLOBAL_TPG_DT_VC7_0

Offset: 0xb4
Byte Offset: 0x2d0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000024 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,0100)

Bit	Reset	Description
5:0	0x24	**DT**: Data type YUV_420_8: 0x18 YUV_420_10: 0x19 YUV_420_8_Legacy: 0x1a YUV_420_8_CSPS: 0x1c YUV_420_10_CSPS: 0x1d YUV_422_8: 0x1e YUV_422_10: 0x1f RGB444: 0x20 RGB555: 0x21 RGB565: 0x22 RGB666: 0x23 RGB888: 0x24 RAW6: 0x28 RAW7: 0x29 RAW8: 0x2a RAW10: 0x2b RAW12: 0x2c RAW14: 0x2d RAW16: 0x2e RAW20: 0x2f

NVCSI_GLOBAL_TPG_RED_PHASE_VC7_0

Offset: 0xb5
Byte Offset: 0x2d4
Read/Write: R/W
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_GREEN_PHASE_VC7_0
Offset: 0xb6
Byte Offset: 0x2d8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_BLUE_PHASE_VC7_0
Offset: 0xb7
Byte Offset: 0x2dc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_RED_FREQ_VC7_0
Offset: 0xb8
Byte Offset: 0x2e0

Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_GREEN_FREQ_VC7_0

Offset: 0xb9
Byte Offset: 0x2e4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_BLUE_FREQ_VC7_0

Offset: 0xba
Byte Offset: 0x2e8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_RED_FREQ_RATE_VC7_0

Offset: 0xbb
Byte Offset: 0x2ec
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_GREEN_FREQ_RATE_VC7_0

Offset: 0xbc
Byte Offset: 0x2f0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_BLUE_FREQ_RATE_VC7_0

Offset: 0xbd
Byte Offset: 0x2f4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_SPARE0_VC7_0
Offset: 0xbe
Byte Offset: 0x2f8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_SPARE1_VC7_0
Offset: 0xbf
Byte Offset: 0x2fc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_CTRL_VC8_0
Offset: 0xc0
Byte Offset: 0x300
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000008 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1000)

Bit	Reset	Description
3	ENABLE	**PHASE_INC**: If the is field is set, the initial phase will increase 512 by frame. **0** = DISABLE **1** = ENABLE
2	DISABLE	**AUTO_STOP**: If the auto stop is set, this virtual channel will automaticly stop when the frame count to the TPG_FN_VCx.MAX_FN. **0** = DISABLE **1** = ENABLE
1:0	DIRECT	**MODE**: Pattern mode **0** = DIRECT **1** = PATCH **2** = SINE **3** = RSVD

NVCSI_GLOBAL_TPG_EMB_CTRL_VC8_0

Offset: 0xc1
Byte Offset: 0x304
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01000000 (0bxxxx,xxx1,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
24	0x1	**PATTERN_MODE**: The embbed line payload.
23:20	0x0	**TOP_LINES**: The embbed line number before the frame.
19:16	0x0	**BOTTOM_LINES**: The embbed line number after the frame.
15:0	0x0	**WIDTH**: Embedded line data width, unit is byte.

NVCSI_GLOBAL_TPG_SIZE_VC8_0

Offset: 0xc2

Byte Offset: 0x308
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00400040 (0b0000,0000,0100,0000,0000,0000,0100,0000)

Bit	Reset	Description
31:16	0x40	**Y_SIZE**: the frame virtical size
15:0	0x40	**X_SIZE**: The frame horizontal size

NVCSI_GLOBAL_TPG_FN_VC8_0
Offset: 0xc3
Byte Offset: 0x30c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xffff0000 (0b1111,1111,1111,1111,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0xffff	**MAX_FN**: The max frame ID
15:0	0x0	**MIN_FN**: The min frame ID

NVCSI_GLOBAL_TPG_DT_VC8_0
Offset: 0xc4
Byte Offset: 0x310
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000024 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,0100)

Bit	Reset	Description
5:0	0x24	**DT**: Data type YUV_420_8: 0x18 YUV_420_10: 0x19 YUV_420_8_Legacy: 0x1a YUV_420_8_CSPS: 0x1c YUV_420_10_CSPS: 0x1d YUV_422_8: 0x1e YUV_422_10: 0x1f RGB444: 0x20 RGB555: 0x21 RGB565: 0x22 RGB666: 0x23 RGB888: 0x24 RAW6: 0x28 RAW7: 0x29 RAW8: 0x2a RAW10: 0x2b RAW12: 0x2c RAW14: 0x2d RAW16: 0x2e RAW20: 0x2f

NVCSI_GLOBAL_TPG_RED_PHASE_VC8_0

Offset: 0xc5
Byte Offset: 0x314
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_GREEN_PHASE_VC8_0

Offset: 0xc6
Byte Offset: 0x318
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_BLUE_PHASE_VC8_0

Offset: 0xc7
Byte Offset: 0x31c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_RED_FREQ_VC8_0

Offset: 0xc8
Byte Offset: 0x320
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_GREEN_FREQ_VC8_0

Offset: 0xc9
Byte Offset: 0x324
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_BLUE_FREQ_VC8_0

Offset: 0xca
Byte Offset: 0x328
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_RED_FREQ_RATE_VC8_0

Offset: 0xcb
Byte Offset: 0x32c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_GREEN_FREQ_RATE_VC8_0

Offset: 0xcc
Byte Offset: 0x330

Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_BLUE_FREQ_RATE_VC8_0

Offset: 0xcd
Byte Offset: 0x334
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_SPARE0_VC8_0

Offset: 0xce
Byte Offset: 0x338
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_SPARE1_VC8_0

Offset: 0xcf
Byte Offset: 0x33c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_CTRL_VC9_0

Offset: 0xd0
Byte Offset: 0x340
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000008 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1000)

Bit	Reset	Description
3	ENABLE	**PHASE_INC**: If the is field is set, the initial phase will increase 512 by frame. **0** = DISABLE **1** = ENABLE
2	DISABLE	**AUTO_STOP**: If the auto stop is set, this virtual channel will automaticly stop when the frame count to the TPG_FN_VCx.MAX_FN. **0** = DISABLE **1** = ENABLE
1:0	DIRECT	**MODE**: Pattern mode **0** = DIRECT **1** = PATCH **2** = SINE **3** = RSVD

NVCSI_GLOBAL_TPG_EMB_CTRL_VC9_0

Offset: 0xd1

Byte Offset: 0x344
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01000000 (0bxxxx,xxx1,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
24	0x1	**PATTERN_MODE**: The embbed line payload.
23:20	0x0	**TOP_LINES**: The embbed line number before the frame.
19:16	0x0	**BOTTOM_LINES**: The embbed line number after the frame.
15:0	0x0	**WIDTH**: Embedded line data width, unit is byte.

NVCSI_GLOBAL_TPG_SIZE_VC9_0

Offset: 0xd2
Byte Offset: 0x348
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00400040 (0b0000,0000,0100,0000,0000,0000,0100,0000)

Bit	Reset	Description
31:16	0x40	**Y_SIZE**: the frame virtical size
15:0	0x40	**X_SIZE**: The frame horizontal size

NVCSI_GLOBAL_TPG_FN_VC9_0

Offset: 0xd3
Byte Offset: 0x34c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0xffff0000 (0b1111,1111,1111,1111,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0xffff	**MAX_FN**: The max frame ID
15:0	0x0	**MIN_FN**: The min frame ID

NVCSI_GLOBAL_TPG_DT_VC9_0

Offset: 0xd4
Byte Offset: 0x350
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000024 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,0100)

Bit	Reset	Description
5:0	0x24	**DT**: Data type YUV_420_8: 0x18 YUV_420_10: 0x19 YUV_420_8_Legacy: 0x1a YUV_420_8_CSPS: 0x1c YUV_420_10_CSPS: 0x1d YUV_422_8: 0x1e YUV_422_10: 0x1f RGB444: 0x20 RGB555: 0x21 RGB565: 0x22 RGB666: 0x23 RGB888: 0x24 RAW6: 0x28 RAW7: 0x29 RAW8: 0x2a RAW10: 0x2b RAW12: 0x2c RAW14: 0x2d RAW16: 0x2e RAW20: 0x2f

NVCSI_GLOBAL_TPG_RED_PHASE_VC9_0

Offset: 0xd5
Byte Offset: 0x354
Read/Write: R/W
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_GREEN_PHASE_VC9_0

Offset: 0xd6
Byte Offset: 0x358
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_BLUE_PHASE_VC9_0

Offset: 0xd7
Byte Offset: 0x35c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_RED_FREQ_VC9_0

Offset: 0xd8
Byte Offset: 0x360

Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_GREEN_FREQ_VC9_0

Offset: 0xd9
Byte Offset: 0x364
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_BLUE_FREQ_VC9_0

Offset: 0xda
Byte Offset: 0x368
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_RED_FREQ_RATE_VC9_0

Offset: 0xdb
Byte Offset: 0x36c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_GREEN_FREQ_RATE_VC9_0

Offset: 0xdc
Byte Offset: 0x370
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_BLUE_FREQ_RATE_VC9_0

Offset: 0xdd
Byte Offset: 0x374
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_SPARE0_VC9_0
Offset: 0xde
Byte Offset: 0x378
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_SPARE1_VC9_0
Offset: 0xdf
Byte Offset: 0x37c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_CTRL_VC10_0
Offset: 0xe0
Byte Offset: 0x380
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000008 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1000)

Bit	Reset	Description
3	ENABLE	**PHASE_INC**: If the is field is set, the initial phase will increase 512 by frame. **0** = DISABLE **1** = ENABLE
2	DISABLE	**AUTO_STOP**: If the auto stop is set, this virtual channel will automaticly stop when the frame count to the TPG_FN_VCx.MAX_FN. **0** = DISABLE **1** = ENABLE
1:0	DIRECT	**MODE**: Pattern mode **0** = DIRECT **1** = PATCH **2** = SINE **3** = RSVD

NVCSI_GLOBAL_TPG_EMB_CTRL_VC10_0

Offset: 0xe1
Byte Offset: 0x384
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01000000 (0bxxxx,xxx1,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
24	0x1	**PATTERN_MODE**: The embbed line payload.
23:20	0x0	**TOP_LINES**: The embbed line number before the frame.
19:16	0x0	**BOTTOM_LINES**: The embbed line number after the frame.
15:0	0x0	**WIDTH**: Embedded line data width, unit is byte.

NVCSI_GLOBAL_TPG_SIZE_VC10_0

Offset: 0xe2

Byte Offset: 0x388
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00400040 (0b0000,0000,0100,0000,0000,0000,0100,0000)

Bit	Reset	Description
31:16	0x40	**Y_SIZE**: the frame virtical size
15:0	0x40	**X_SIZE**: The frame horizontal size

NVCSI_GLOBAL_TPG_FN_VC10_0
Offset: 0xe3
Byte Offset: 0x38c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xffff0000 (0b1111,1111,1111,1111,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0xffff	**MAX_FN**: The max frame ID
15:0	0x0	**MIN_FN**: The min frame ID

NVCSI_GLOBAL_TPG_DT_VC10_0
Offset: 0xe4
Byte Offset: 0x390
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000024 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,0100)

Bit	Reset	Description
5:0	0x24	**DT**: Data type YUV_420_8: 0x18 YUV_420_10: 0x19 YUV_420_8_Legacy: 0x1a YUV_420_8_CSPS: 0x1c YUV_420_10_CSPS: 0x1d YUV_422_8: 0x1e YUV_422_10: 0x1f RGB444: 0x20 RGB555: 0x21 RGB565: 0x22 RGB666: 0x23 RGB888: 0x24 RAW6: 0x28 RAW7: 0x29 RAW8: 0x2a RAW10: 0x2b RAW12: 0x2c RAW14: 0x2d RAW16: 0x2e RAW20: 0x2f

NVCSI_GLOBAL_TPG_RED_PHASE_VC10_0

Offset: 0xe5
Byte Offset: 0x394
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_GREEN_PHASE_VC10_0

Offset: 0xe6
Byte Offset: 0x398
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_BLUE_PHASE_VC10_0

Offset: 0xe7
Byte Offset: 0x39c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_RED_FREQ_VC10_0

Offset: 0xe8
Byte Offset: 0x3a0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_GREEN_FREQ_VC10_0

Offset: 0xe9
Byte Offset: 0x3a4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_BLUE_FREQ_VC10_0

Offset: 0xea
Byte Offset: 0x3a8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_RED_FREQ_RATE_VC10_0

Offset: 0xeb
Byte Offset: 0x3ac
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_GREEN_FREQ_RATE_VC10_0

Offset: 0xec
Byte Offset: 0x3b0

Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_BLUE_FREQ_RATE_VC10_0

Offset: 0xed
Byte Offset: 0x3b4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_SPARE0_VC10_0

Offset: 0xee
Byte Offset: 0x3b8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_SPARE1_VC10_0

Offset: 0xef
Byte Offset: 0x3bc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_CTRL_VC11_0

Offset: 0xf0
Byte Offset: 0x3c0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000008 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1000)

Bit	Reset	Description
3	ENABLE	**PHASE_INC**: If the is field is set, the initial phase will increase 512 by frame. **0** = DISABLE **1** = ENABLE
2	DISABLE	**AUTO_STOP**: If the auto stop is set, this virtual channel will automaticly stop when the frame count to the TPG_FN_VCx.MAX_FN. **0** = DISABLE **1** = ENABLE
1:0	DIRECT	**MODE**: Pattern mode **0** = DIRECT **1** = PATCH **2** = SINE **3** = RSVD

NVCSI_GLOBAL_TPG_EMB_CTRL_VC11_0

Offset: 0xf1

Byte Offset: 0x3c4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01000000 (0bxxxx,xxx1,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
24	0x1	**PATTERN_MODE**: The embbed line payload.
23:20	0x0	**TOP_LINES**: The embbed line number before the frame.
19:16	0x0	**BOTTOM_LINES**: The embbed line number after the frame.
15:0	0x0	**WIDTH**: Embedded line data width, unit is byte.

NVCSI_GLOBAL_TPG_SIZE_VC11_0

Offset: 0xf2
Byte Offset: 0x3c8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00400040 (0b0000,0000,0100,0000,0000,0000,0100,0000)

Bit	Reset	Description
31:16	0x40	**Y_SIZE**: the frame virtical size
15:0	0x40	**X_SIZE**: The frame horizontal size

NVCSI_GLOBAL_TPG_FN_VC11_0

Offset: 0xf3
Byte Offset: 0x3cc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0xffff0000 (0b1111,1111,1111,1111,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0xffff	**MAX_FN**: The max frame ID
15:0	0x0	**MIN_FN**: The min frame ID

NVCSI_GLOBAL_TPG_DT_VC11_0

Offset: 0xf4
Byte Offset: 0x3d0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000024 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,0100)

Bit	Reset	Description
5:0	0x24	**DT**: Data type YUV_420_8: 0x18 YUV_420_10: 0x19 YUV_420_8_Legacy: 0x1a YUV_420_8_CSPS: 0x1c YUV_420_10_CSPS: 0x1d YUV_422_8: 0x1e YUV_422_10: 0x1f RGB444: 0x20 RGB555: 0x21 RGB565: 0x22 RGB666: 0x23 RGB888: 0x24 RAW6: 0x28 RAW7: 0x29 RAW8: 0x2a RAW10: 0x2b RAW12: 0x2c RAW14: 0x2d RAW16: 0x2e RAW20: 0x2f

NVCSI_GLOBAL_TPG_RED_PHASE_VC11_0

Offset: 0xf5
Byte Offset: 0x3d4
Read/Write: R/W
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_GREEN_PHASE_VC11_0

Offset: 0xf6
Byte Offset: 0x3d8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_BLUE_PHASE_VC11_0

Offset: 0xf7
Byte Offset: 0x3dc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_RED_FREQ_VC11_0

Offset: 0xf8
Byte Offset: 0x3e0

Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_GREEN_FREQ_VC11_0

Offset: 0xf9
Byte Offset: 0x3e4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_BLUE_FREQ_VC11_0

Offset: 0xfa
Byte Offset: 0x3e8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_RED_FREQ_RATE_VC11_0

Offset: 0xfb
Byte Offset: 0x3ec
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_GREEN_FREQ_RATE_VC11_0

Offset: 0xfc
Byte Offset: 0x3f0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_BLUE_FREQ_RATE_VC11_0

Offset: 0xfd
Byte Offset: 0x3f4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_SPARE0_VC11_0
Offset: 0xfe
Byte Offset: 0x3f8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_SPARE1_VC11_0
Offset: 0xff
Byte Offset: 0x3fc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_CTRL_VC12_0
Offset: 0x100
Byte Offset: 0x400
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000008 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1000)

Bit	Reset	Description
3	ENABLE	**PHASE_INC**: If the is field is set, the initial phase will increase 512 by frame. **0** = DISABLE **1** = ENABLE
2	DISABLE	**AUTO_STOP**: If the auto stop is set, this virtual channel will automaticly stop when the frame count to the TPG_FN_VCx.MAX_FN. **0** = DISABLE **1** = ENABLE
1:0	DIRECT	**MODE**: Pattern mode **0** = DIRECT **1** = PATCH **2** = SINE **3** = RSVD

NVCSI_GLOBAL_TPG_EMB_CTRL_VC12_0

Offset: 0x101
Byte Offset: 0x404
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01000000 (0bxxxx,xxx1,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
24	0x1	**PATTERN_MODE**: The embbed line payload.
23:20	0x0	**TOP_LINES**: The embbed line number before the frame.
19:16	0x0	**BOTTOM_LINES**: The embbed line number after the frame.
15:0	0x0	**WIDTH**: Embedded line data width, unit is byte.

NVCSI_GLOBAL_TPG_SIZE_VC12_0

Offset: 0x102

Byte Offset: 0x408
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00400040 (0b0000,0000,0100,0000,0000,0000,0100,0000)

Bit	Reset	Description
31:16	0x40	**Y_SIZE**: the frame virtical size
15:0	0x40	**X_SIZE**: The frame horizontal size

NVCSI_GLOBAL_TPG_FN_VC12_0
Offset: 0x103
Byte Offset: 0x40c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xffff0000 (0b1111,1111,1111,1111,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0xffff	**MAX_FN**: The max frame ID
15:0	0x0	**MIN_FN**: The min frame ID

NVCSI_GLOBAL_TPG_DT_VC12_0
Offset: 0x104
Byte Offset: 0x410
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000024 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,0100)

Bit	Reset	Description
5:0	0x24	**DT**: Data type YUV_420_8: 0x18 YUV_420_10: 0x19 YUV_420_8_Legacy: 0x1a YUV_420_8_CSPS: 0x1c YUV_420_10_CSPS: 0x1d YUV_422_8: 0x1e YUV_422_10: 0x1f RGB444: 0x20 RGB555: 0x21 RGB565: 0x22 RGB666: 0x23 RGB888: 0x24 RAW6: 0x28 RAW7: 0x29 RAW8: 0x2a RAW10: 0x2b RAW12: 0x2c RAW14: 0x2d RAW16: 0x2e RAW20: 0x2f

NVCSI_GLOBAL_TPG_RED_PHASE_VC12_0

Offset: 0x105
Byte Offset: 0x414
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_GREEN_PHASE_VC12_0

Offset: 0x106
Byte Offset: 0x418
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_BLUE_PHASE_VC12_0

Offset: 0x107
Byte Offset: 0x41c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_RED_FREQ_VC12_0

Offset: 0x108
Byte Offset: 0x420
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_GREEN_FREQ_VC12_0

Offset: 0x109
Byte Offset: 0x424
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_BLUE_FREQ_VC12_0

Offset: 0x10a
Byte Offset: 0x428
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_RED_FREQ_RATE_VC12_0

Offset: 0x10b
Byte Offset: 0x42c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_GREEN_FREQ_RATE_VC12_0

Offset: 0x10c
Byte Offset: 0x430

Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_BLUE_FREQ_RATE_VC12_0

Offset: 0x10d
Byte Offset: 0x434
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_SPARE0_VC12_0

Offset: 0x10e
Byte Offset: 0x438
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_SPARE1_VC12_0

Offset: 0x10f
Byte Offset: 0x43c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_CTRL_VC13_0

Offset: 0x110
Byte Offset: 0x440
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000008 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1000)

Bit	Reset	Description
3	ENABLE	**PHASE_INC**: If the is field is set, the initial phase will increase 512 by frame. **0** = DISABLE **1** = ENABLE
2	DISABLE	**AUTO_STOP**: If the auto stop is set, this virtual channel will automaticly stop when the frame count to the TPG_FN_VCx.MAX_FN. **0** = DISABLE **1** = ENABLE
1:0	DIRECT	**MODE**: Pattern mode **0** = DIRECT **1** = PATCH **2** = SINE **3** = RSVD

NVCSI_GLOBAL_TPG_EMB_CTRL_VC13_0

Offset: 0x111

Byte Offset: 0x444
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01000000 (0bxxxx,xxx1,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
24	0x1	**PATTERN_MODE**: The embbed line payload.
23:20	0x0	**TOP_LINES**: The embbed line number before the frame.
19:16	0x0	**BOTTOM_LINES**: The embbed line number after the frame.
15:0	0x0	**WIDTH**: Embedded line data width, unit is byte.

NVCSI_GLOBAL_TPG_SIZE_VC13_0

Offset: 0x112
Byte Offset: 0x448
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00400040 (0b0000,0000,0100,0000,0000,0000,0100,0000)

Bit	Reset	Description
31:16	0x40	**Y_SIZE**: the frame virtical size
15:0	0x40	**X_SIZE**: The frame horizontal size

NVCSI_GLOBAL_TPG_FN_VC13_0

Offset: 0x113
Byte Offset: 0x44c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0xffff0000 (0b1111,1111,1111,1111,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0xffff	**MAX_FN**: The max frame ID
15:0	0x0	**MIN_FN**: The min frame ID

NVCSI_GLOBAL_TPG_DT_VC13_0

Offset: 0x114
Byte Offset: 0x450
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000024 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,0100)

Bit	Reset	Description
5:0	0x24	**DT**: Data type YUV_420_8: 0x18 YUV_420_10: 0x19 YUV_420_8_Legacy: 0x1a YUV_420_8_CSPS: 0x1c YUV_420_10_CSPS: 0x1d YUV_422_8: 0x1e YUV_422_10: 0x1f RGB444: 0x20 RGB555: 0x21 RGB565: 0x22 RGB666: 0x23 RGB888: 0x24 RAW6: 0x28 RAW7: 0x29 RAW8: 0x2a RAW10: 0x2b RAW12: 0x2c RAW14: 0x2d RAW16: 0x2e RAW20: 0x2f

NVCSI_GLOBAL_TPG_RED_PHASE_VC13_0

Offset: 0x115
Byte Offset: 0x454
Read/Write: R/W
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_GREEN_PHASE_VC13_0

Offset: 0x116
Byte Offset: 0x458
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_BLUE_PHASE_VC13_0

Offset: 0x117
Byte Offset: 0x45c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_RED_FREQ_VC13_0

Offset: 0x118
Byte Offset: 0x460

Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_GREEN_FREQ_VC13_0

Offset: 0x119
Byte Offset: 0x464
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_BLUE_FREQ_VC13_0

Offset: 0x11a
Byte Offset: 0x468
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_RED_FREQ_RATE_VC13_0

Offset: 0x11b
Byte Offset: 0x46c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_GREEN_FREQ_RATE_VC13_0

Offset: 0x11c
Byte Offset: 0x470
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_BLUE_FREQ_RATE_VC13_0

Offset: 0x11d
Byte Offset: 0x474
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_SPARE0_VC13_0
Offset: 0x11e
Byte Offset: 0x478
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_SPARE1_VC13_0
Offset: 0x11f
Byte Offset: 0x47c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_CTRL_VC14_0
Offset: 0x120
Byte Offset: 0x480
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000008 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1000)

Bit	Reset	Description
3	ENABLE	**PHASE_INC**: If the is field is set, the initial phase will increase 512 by frame. **0** = DISABLE **1** = ENABLE
2	DISABLE	**AUTO_STOP**: If the auto stop is set, this virtual channel will automaticly stop when the frame count to the TPG_FN_VCx.MAX_FN. **0** = DISABLE **1** = ENABLE
1:0	DIRECT	**MODE**: Pattern mode **0** = DIRECT **1** = PATCH **2** = SINE **3** = RSVD

NVCSI_GLOBAL_TPG_EMB_CTRL_VC14_0

Offset: 0x121
Byte Offset: 0x484
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01000000 (0bxxxx,xxx1,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
24	0x1	**PATTERN_MODE**: The embbed line payload.
23:20	0x0	**TOP_LINES**: The embbed line number before the frame.
19:16	0x0	**BOTTOM_LINES**: The embbed line number after the frame.
15:0	0x0	**WIDTH**: Embedded line data width, unit is byte.

NVCSI_GLOBAL_TPG_SIZE_VC14_0

Offset: 0x122

Byte Offset: 0x488
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00400040 (0b0000,0000,0100,0000,0000,0000,0100,0000)

Bit	Reset	Description
31:16	0x40	**Y_SIZE**: the frame virtical size
15:0	0x40	**X_SIZE**: The frame horizontal size

NVCSI_GLOBAL_TPG_FN_VC14_0
Offset: 0x123
Byte Offset: 0x48c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xffff0000 (0b1111,1111,1111,1111,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0xffff	**MAX_FN**: The max frame ID
15:0	0x0	**MIN_FN**: The min frame ID

NVCSI_GLOBAL_TPG_DT_VC14_0
Offset: 0x124
Byte Offset: 0x490
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000024 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,0100)

Bit	Reset	Description
5:0	0x24	**DT**: Data type YUV_420_8: 0x18 YUV_420_10: 0x19 YUV_420_8_Legacy: 0x1a YUV_420_8_CSPS: 0x1c YUV_420_10_CSPS: 0x1d YUV_422_8: 0x1e YUV_422_10: 0x1f RGB444: 0x20 RGB555: 0x21 RGB565: 0x22 RGB666: 0x23 RGB888: 0x24 RAW6: 0x28 RAW7: 0x29 RAW8: 0x2a RAW10: 0x2b RAW12: 0x2c RAW14: 0x2d RAW16: 0x2e RAW20: 0x2f

NVCSI_GLOBAL_TPG_RED_PHASE_VC14_0

Offset: 0x125
Byte Offset: 0x494
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_GREEN_PHASE_VC14_0

Offset: 0x126
Byte Offset: 0x498
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_BLUE_PHASE_VC14_0

Offset: 0x127
Byte Offset: 0x49c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_RED_FREQ_VC14_0

Offset: 0x128
Byte Offset: 0x4a0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_GREEN_FREQ_VC14_0

Offset: 0x129
Byte Offset: 0x4a4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_BLUE_FREQ_VC14_0

Offset: 0x12a
Byte Offset: 0x4a8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_RED_FREQ_RATE_VC14_0

Offset: 0x12b
Byte Offset: 0x4ac
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_GREEN_FREQ_RATE_VC14_0

Offset: 0x12c
Byte Offset: 0x4b0

Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_BLUE_FREQ_RATE_VC14_0

Offset: 0x12d
Byte Offset: 0x4b4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_SPARE0_VC14_0

Offset: 0x12e
Byte Offset: 0x4b8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_SPARE1_VC14_0

Offset: 0x12f
Byte Offset: 0x4bc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_CTRL_VC15_0

Offset: 0x130
Byte Offset: 0x4c0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000008 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1000)

Bit	Reset	Description
3	ENABLE	**PHASE_INC**: If the is field is set, the initial phase will increase 512 by frame. **0** = DISABLE **1** = ENABLE
2	DISABLE	**AUTO_STOP**: If the auto stop is set, this virtual channel will automaticly stop when the frame count to the TPG_FN_VCx.MAX_FN. **0** = DISABLE **1** = ENABLE
1:0	DIRECT	**MODE**: Pattern mode **0** = DIRECT **1** = PATCH **2** = SINE **3** = RSVD

NVCSI_GLOBAL_TPG_EMB_CTRL_VC15_0

Offset: 0x131

Byte Offset: 0x4c4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01000000 (0bxxxx,xxx1,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
24	0x1	**PATTERN_MODE**: The embbed line payload.
23:20	0x0	**TOP_LINES**: The embbed line number before the frame.
19:16	0x0	**BOTTOM_LINES**: The embbed line number after the frame.
15:0	0x0	**WIDTH**: Embedded line data width, unit is byte.

NVCSI_GLOBAL_TPG_SIZE_VC15_0

Offset: 0x132
Byte Offset: 0x4c8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00400040 (0b0000,0000,0100,0000,0000,0000,0100,0000)

Bit	Reset	Description
31:16	0x40	**Y_SIZE**: the frame virtical size
15:0	0x40	**X_SIZE**: The frame horizontal size

NVCSI_GLOBAL_TPG_FN_VC15_0

Offset: 0x133
Byte Offset: 0x4cc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0xffff0000 (0b1111,1111,1111,1111,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0xffff	**MAX_FN**: The max frame ID
15:0	0x0	**MIN_FN**: The min frame ID

NVCSI_GLOBAL_TPG_DT_VC15_0

Offset: 0x134
Byte Offset: 0x4d0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000024 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,0100)

Bit	Reset	Description
5:0	0x24	**DT**: Data type YUV_420_8: 0x18 YUV_420_10: 0x19 YUV_420_8_Legacy: 0x1a YUV_420_8_CSPS: 0x1c YUV_420_10_CSPS: 0x1d YUV_422_8: 0x1e YUV_422_10: 0x1f RGB444: 0x20 RGB555: 0x21 RGB565: 0x22 RGB666: 0x23 RGB888: 0x24 RAW6: 0x28 RAW7: 0x29 RAW8: 0x2a RAW10: 0x2b RAW12: 0x2c RAW14: 0x2d RAW16: 0x2e RAW20: 0x2f

NVCSI_GLOBAL_TPG_RED_PHASE_VC15_0

Offset: 0x135
Byte Offset: 0x4d4
Read/Write: R/W
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_GREEN_PHASE_VC15_0

Offset: 0x136
Byte Offset: 0x4d8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_BLUE_PHASE_VC15_0

Offset: 0x137
Byte Offset: 0x4dc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14:0	0x0	**PHASE**: initial phase

NVCSI_GLOBAL_TPG_RED_FREQ_VC15_0

Offset: 0x138
Byte Offset: 0x4e0

Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_GREEN_FREQ_VC15_0

Offset: 0x139
Byte Offset: 0x4e4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_BLUE_FREQ_VC15_0

Offset: 0x13a
Byte Offset: 0x4e8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
30:16	0x0	**VERT_INIT_FREQ**
14:0	0x0	**HOR_INIT_FREQ**

NVCSI_GLOBAL_TPG_RED_FREQ_RATE_VC15_0

Offset: 0x13b
Byte Offset: 0x4ec
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_GREEN_FREQ_RATE_VC15_0

Offset: 0x13c
Byte Offset: 0x4f0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_BLUE_FREQ_RATE_VC15_0

Offset: 0x13d
Byte Offset: 0x4f4
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
24:16	0x0	**VERT_FREQ_RATE**
8:0	0x0	**HOR_FREQ_RATE**

NVCSI_GLOBAL_TPG_SPARE0_VC15_0

Offset: 0x13e
Byte Offset: 0x4f8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_SPARE1_VC15_0

Offset: 0x13f
Byte Offset: 0x4fc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE**

NVCSI_GLOBAL_TPG_PRBS_CTRL_0_0

Offset: 0x140
Byte Offset: 0x500
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	DISABLE	**PRBS_MODE_EN**: **0** = DISABLE **1** = ENABLE
0	FIX_SEED	**INIT_MODE**: **0** = FIX_SEED **1** = FRAME_ID_SEED

NVCSI_GLOBAL_TPG_PRBS_CTRL_1_0
Offset: 0x141
Byte Offset: 0x504
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x37a672ac (0b0011,0111,1010,0110,0111,0010,1010,1100)

Bit	Reset	Description
31:0	0x37a672ac	**PRBS_SEED_LOW**

NVCSI_GLOBAL_TPG_PRBS_CTRL_2_0
Offset: 0x142
Byte Offset: 0x508
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x08c20001 (0b0000,1000,1100,0010,0000,0000,0000,0001)

Bit	Reset	Description
31:0	0x8c20001	**PRBS_SEED_HIGH**

7.2.1.3.8 NVCSI PHY Registers

NVCSI_PHY_0_NVCSI_CIL_PHY_CTRL_0

Offset: 0x4700
Byte Offset: 0x11c00
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DPHY	**CFG_PHY_MODE**: The CPHY/DPHY type mode. **0** = DPHY **1** = CPHY

NVCSI_PHY_0_LM_SLCG_CTRL_0

Offset: 0x4701
Byte Offset: 0x11c04
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	DISABLE	**cfg_lm1_slcg_override**: Enable the SLCG override for lane merger 1. **0** = DISABLE **1** = ENABLE
0	DISABLE	**cfg_lm0_slcg_override**: Enable the SLCG override for lane merger 0. **0** = DISABLE **1** = ENABLE

NVCSI_PHY_0_LM_EN_CTRL_0

Offset: 0x4702
Byte Offset: 0x11c08
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x0000060c (0bxxxx,xxxx,xxxx,xxxx,xxx0,0110,xx00,11x0)

Bit	Reset	Description
12:9	0x3	**cfg_lm1_water_mark**: Water mark for LM0 detect stuck
8	DISABLE	**cfg_lm1_en**: Lane Merger 1 enable **0** = DISABLE **1** = ENABLE
5:2	0x3	**cfg_lm0_water_mark**: Water mark for LM0 detect stuck
0	DISABLE	**cfg_lm0_en**: Lane Merger 0 enable **0** = DISABLE **1** = ENABLE

NVCSI_PHY_0_LM_SW_RESET_0

Offset: 0x4703
Byte Offset: 0x11c0c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	DISABLE	**cfg_lm1_swreset**: Reset the lane merger 1. **0** = DISABLE **1** = ENABLE
0	DISABLE	**cfg_lm0_swreset**: Reset the lane merger 0. **0** = DISABLE **1** = ENABLE

NVCSI_PHY_0_NVCSI_CIL_CONFIG_0

Offset: 0x4704
Byte Offset: 0x11c10
Read/Write: R/W

Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,x000,xxxx,x000)

Bit	Reset	Description
10:8	0x0	**DATA_LANE_B**: Lane number for lane merger B. 000: 0 lanes active 001: 1 lanes active 010: 2 lanes active Others: illegal
2:0	0x0	**DATA_LANE_A**: Lane number for lane merger A. 000: 0 lanes active 001: 1 lanes active 010: 2 lanes active 011: 3 lanes active (valid only in CPHY mode. NA in DPHY mode) 100: 4 lanes active, CLK from Partition A is used. 101: 4 lanes active, CLK from Partition B is used. Others: illegal

NVCSI_PHY_0_NVCSI_CIL_CLKEN_OVERRIDE_CTRL_0

Offset: 0x4705
Byte Offset: 0x11c14
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**CLKEN_OVERRIDE**: CLKENABLE OVERRIDE FOR CIL. **1** = ENABLE **0** = DISABLE

NVCSI_PHY_0_NVCSI_CIL_PAD_CONFIG_0

Offset: 0x4706
Byte Offset: 0x11c18
Read/Write: R/W
Parity Protection: Y
Shadow: N

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00040100 (0bxxxx,x000,0000,0100,0000,xxx1,0000,0000)

Bit	Reset	Description
26:25	0x0	**VD09REG_LEVEL**: 2-bit regulator level control. control voltage level output of 0.9v regulator. 00:0.9v, 01:0.85v, 10: 0.95v, 11: 1v
24:23	0x0	**VD09REG_LEAKER**: 2-bit regulator leaker control. 11 will give best performance, but consumers more DC quiescent current. (control 0.9v internal regulator for all partitions)
22:21	0x0	**VD04REG_LEVEL**: 2-bit regulator level control control voltage level output of 0.4V regulator 00: 0.4v 01: 0.38V 10: 0.42V 11: 0.44V
20:19	0x0	**VD04REG_LEAKER**: 2-bit regulator leaker control. 11 will give best performance, but consumers more DC quiescent current. (control 0.4v internal tx regulator for all partitions)
18:16	0x4	**LPRX_LEVEL_SEL**: Internal vref adjustment for LP self-biased receiver(when LP_RX_SELECT_IO1_[A/B]=1)
15:12	0x0	**LOADADJ**: LOAD ADJ value to be connected to the pad not used for CSI functional mode. used only for IO bist
8	0x1	**PDVCLAMP**: Power down regular which supplies current to de-serializer logic. Active High.
7:5	0x0	**SEL_CKTEST**
4	0x0	**E_CKTEST**: Enable clock test output.
3:1	0x0	**SEL_ATEST**
0	0x0	**E_ATEST**: Enable analog test voltage output.

NVCSI_PHY_0_NVCSI_CIL_PAD_CONFIG_1_0

Offset: 0x4707
Byte Offset: 0x11c1c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE_TOP**: spare register bits for top level control

NVCSI_PHY_0_NVCSI_CIL_LANE_SWIZZLE_CTRL_0

Offset: 0x4708
Byte Offset: 0x11c20
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4:0	0x0	**LANE_SWIZZLE_CTRL**: Lane Swizzle control for Brick. Valid in both CPHY and DPHY modes. For CPHY, all modes are supported. But for DPHY, some mode are not supported with different lane count. Only for 4 lanes: 00011 A0 A1 B0 B1 --> A0 B0 A1 B1 00100 A0 A1 B0 B1 --> A0 B1 A1 B0 00101 A0 A1 B0 B1 --> A0 B1 B0 A1 00010 A0 A1 B0 B1 --> A0 B0 B1 A1 01000 A0 A1 B0 B1 --> A1 B0 B1 A0 01001 A0 A1 B0 B1 --> A1 B0 A0 B1 01010 A0 A1 B0 B1 --> A1 B1 A0 B0 01011 A0 A1 B0 B1 --> A1 B1 B0 A0 01100 A0 A1 B0 B1 --> B0 A1 A0 B1 01101 A0 A1 B0 B1 --> B0 A1 B1 A0 01110 A0 A1 B0 B1 --> B0 A0 B1 A1 01111 A0 A1 B0 B1 --> B0 A0 A1 B1 10000 A0 A1 B0 B1 --> B0 B1 A1 A0 10001 A0 A1 B0 B1 --> B0 B1 A0 A1 10010 A0 A1 B0 B1 --> B1 A1 B0 A0 10011 A0 A1 B0 B1 --> B1 A1 A0 B0 10100 A0 A1 B0 B1 --> B1 B0 A0 A1 10101 A0 A1 B0 B1 --> B1 B0 A1 A0 10110 A0 A1 B0 B1 --> B1 A0 A1 B0 10111 A0 A1 B0 B1 --> B1 A0 B0 A1 Support for 1/2/4 lanes 00000 A0 A1 B0 B1 --> A0 A1 B0 B1 00001 A0 A1 B0 B1 --> A0 A1 B1 B0 00110 A0 A1 B0 B1 --> A1 A0 B0 B1 00111 A0 A1 B0 B1 --> A1 A0 B1 B0

NVCSI_PHY_0_NVCSI_CIL_BK_MODE_STATUS_0

Offset: 0x4709

Byte Offset: 0x11c24
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
1:0	X	**BK_MODE**: 00: two independent 2x bricks 01: one 4x brick, received clock from partition A is used. Clock from partition B is not used 10: one 4x brick, received clock from partition B is used. Clock from partition A is not used 11: illegal

NVCSI_PHY_0_NVCSI_CIL_TX_TIMING_0_0

Offset: 0x470a
Byte Offset: 0x11c28
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00006030 (0bxxxx,xxxx,xxxx,xxxx,0110,0000,0011,0000)

Bit	Reset	Description
15:8	0x60	**THSEXIT**: THSEXIT length, in slow clock cycle number
7:0	0x30	**TLPX**: TLPX length, in slow clock cycle number

NVCSI_PHY_0_NVCSI_CIL_TX_TIMING_1_0

Offset: 0x470b
Byte Offset: 0x11c2c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00101030 (0bxxxx,xxxx,0001,0000,0001,0000,0011,0000)

Bit	Reset	Description
23:16	0x10	**T3POST**: T3POST length, in slow clock cycle number
15:8	0x10	**T3PREBEGIN**: T3PREBEGIN length, in slow clock cycle number
7:0	0x30	**T3PREPARE**: T3PREPARE length, in slow clock cycle number

NVCSI_PHY_0_NVCSI_CIL_TX_TIMING_2_0

Offset: 0x470c
Byte Offset: 0x11c30
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:8	0xfff	**T3CALALTSEQ**: T3CALALTSEQ length, in slow clock cycle number
7:0	0xff	**T3CALPREAMBLE**: T3CALPREAMBLE length, in slow clock cycle number

NVCSI_PHY_0_NVCSI_CIL_TX_CALIB_CTRL_0

Offset: 0x470d
Byte Offset: 0x11c34
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000004 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0100)

Bit	Reset	Description
17:2	0x1	**CALIB_SEED**: Calibration sequence format 2 PRBS seed. Should set to a non-zero value.

Bit	Reset	Description
1	FMT_1	**CALIB_SEQ**: Calibration sequence format. **0** = FMT_1 **1** = FMT_2
0	DISABLE	**CALIB_EN**: Calibration sequence enable **0** = DISABLE **1** = ENABLE

NVCSI_PHY_0_NVCSI_CIL_TX_CTRL_0

Offset: 0x470e
Byte Offset: 0x11c38
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000004 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0100)

Bit	Reset	Description
4	DISABLE	**TX_ENABLE**: Enable the TX SCIL. **0** = DISABLE **1** = ENABLE
3	SINGLE	**PAD_BK_MODE**: Set the pad to one stream or two independent stream usage. **0** = SINGLE **1** = DUAL
2:0	0x4	**LANE_NUM**: Valid value: 1/2/4

NVCSI_PHY_0_NVCSI_CIL_TX_SW_RESET_0_0

Offset: 0x470f
Byte Offset: 0x11c3c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	ENABLE	**SW_RESET**: Reset the TX lane **0** = DISABLE **1** = ENABLE

NVCSI_PHY_0_NVCSI_CIL_CPHY_BIST_CONFIG_0_0

Offset: 0x4710
Byte Offset: 0x11c40
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000009 (0bxxxx,xxxx,xx00,0000,0000,0000,0000,1001)

Bit	Reset	Description
21	DISABLE	**REALIGN_EN**: If set to enable, the BIST logic will re-align the reference symbol to match the received symbol when there is clock slip by wire state error. **0** = DISABLE **1** = ENABLE
20:3	0x1	**PRBS_SEED**: Seed of the PRBS pattern. Should set to a non-zero value.
2:1	0x0	**PRBS_PATTERN**: PRBS pattern PRBS9: X0+X5+X9 PRBS11: X0+X9+X11 PRBS18: X0+X11+X18 **0** = PRBS9 **1** = PRBS11 **2** = PRBS18
0	MODE1	**BIST_MODE**: BIST control mode MODE0: Set the expected error number, BIST logic will keep comparing the received symbols and the reference PRBS generated symbols. When the expected error number got, BIST logic stop. And report how many word are totally compared. MODE11: Set the expected word number, BIST logic will keep comparing the received symbols and the reference PRBS generated symbols. When the expected words(symbols) are compared, BIST logic stop and report how many errors are found. **0** = MODE0 **1** = MODE1

NVCSI_PHY_0_NVCSI_CIL_CPHY_BIST_CONFIG_1_0

Offset: 0x4711
Byte Offset: 0x11c44
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000100 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx1,0000,0000)

Bit	Reset	Description
8	0x1	**ERR_TYPE**: The expecte error number type. SYMBOL_NUM: symbol error number. **0** = WORD_NUM :WORD_NUM: word error number. **1** = SYMBOL_NUM
7:0	0x0	**ERR_NUM**: The expected error number, used for BIST MODE 0.

NVCSI_PHY_0_NVCSI_CIL_CPHY_BIST_CONFIG_2_0

Offset: 0x4712
Byte Offset: 0x11c48
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WORD_NUM_LOW**: The low 32 bits of the expect word number, used for BIST MODE 1.

NVCSI_PHY_0_NVCSI_CIL_CPHY_BIST_CONFIG_3_0

Offset: 0x4713
Byte Offset: 0x11c4c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WORD_NUM_HIGH**: The higher 16 bits of the expected word number, used for BIST MODE 1.

NVCSI_PHY_0_NVCSI_CIL_IDAC_CALIB_START_0

Offset: 0x4714
Byte Offset: 0x11c50
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**IDAC_CAL_TRIGGER**: Trigger the IDAC calibration.

NVCSI_PHY_0_NVCSI_CIL_IDAC_CALIB_CTRL_0

Offset: 0x4715
Byte Offset: 0x11c54
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,0000,xxxx,x000,0000,0101)

Bit	Reset	Description
19	0x0	**IDAC_CAL_EN_TRIO1_B**: Enable IDAC calibration for trio B1
18	0x0	**IDAC_CAL_EN_TRIO0_B**: Enable IDAC calibration for trio B0
17	0x0	**IDAC_CAL_EN_TRIO1_A**: Enable IDAC calibration for trio A1
16	0x0	**IDAC_CAL_EN_TRIO0_A**: Enable IDAC calibration for trio A0

Bit	Reset	Description
10:0	0x5	**IDAC_SETTLE_TIME**: How many cycle for settle to sample the done from pad.

NVCSI_PHY_0_NVCSI_CIL_IDAC_CALIB_STATUS_0

Offset: 0x4716
Byte Offset: 0x11c58
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,0000,0000,0000,xxxx,0000,0000,0000)

Bit	Reset	Description
27	0x0	**IDAC_CALIB_MAXOUT_RXCA_TRIO1_B**: IDAC calibration maxout for trio B1 RXCA.
26	0x0	**IDAC_CALIB_MAXOUT_RXBC_TRIO1_B**: IDAC calibration maxout for trio B1 RXBC.
25	0x0	**IDAC_CALIB_MAXOUT_RXAB_TRIO1_B**: IDAC calibration maxout for trio B1 RXAB.
24	0x0	**IDAC_CALIB_MAXOUT_RXCA_TRIO0_B**: IDAC calibration maxout for trio B0 RXCA.
23	0x0	**IDAC_CALIB_MAXOUT_RXBC_TRIO0_B**: IDAC calibration maxout for trio B0 RXBC.
22	0x0	**IDAC_CALIB_MAXOUT_RXAB_TRIO0_B**: IDAC calibration maxout for trio B0 RXAB.
21	0x0	**IDAC_CALIB_MAXOUT_RXCA_TRIO1_A**: IDAC calibration maxout for trio A1 RXCA.
20	0x0	**IDAC_CALIB_MAXOUT_RXBC_TRIO1_A**: IDAC calibration maxout for trio A1 RXBC.
19	0x0	**IDAC_CALIB_MAXOUT_RXAB_TRIO1_A**: IDAC calibration maxout for trio A1 RXAB.
18	0x0	**IDAC_CALIB_MAXOUT_RXCA_TRIO0_A**: IDAC calibration maxout for trio A0 RXCA.
17	0x0	**IDAC_CALIB_MAXOUT_RXBC_TRIO0_A**: IDAC calibration maxout for trio A0 RXBC.
16	0x0	**IDAC_CALIB_MAXOUT_RXAB_TRIO0_A**: IDAC calibration maxout for trio A0 RXAB.
11	0x0	**IDAC_CALIB_DONE_RXCA_TRIO1_B**: IDAC calibration done for trio B1 RXCA.

Bit	Reset	Description
10	0x0	**IDAC_CALIB_DONE_RXBC_TRIO1_B**: IDAC calibration done for trio B1 RXBC.
9	0x0	**IDAC_CALIB_DONE_RXAB_TRIO1_B**: IDAC calibration done for trio B1 RXAB.
8	0x0	**IDAC_CALIB_DONE_RXCA_TRIO0_B**: IDAC calibration done for trio B0 RXCA.
7	0x0	**IDAC_CALIB_DONE_RXBC_TRIO0_B**: IDAC calibration done for trio B0 RXBC.
6	0x0	**IDAC_CALIB_DONE_RXAB_TRIO0_B**: IDAC calibration done for trio B0 RXAB.
5	0x0	**IDAC_CALIB_DONE_RXCA_TRIO1_A**: IDAC calibration done for trio A1 RXCA.
4	0x0	**IDAC_CALIB_DONE_RXBC_TRIO1_A**: IDAC calibration done for trio A1 RXBC.
3	0x0	**IDAC_CALIB_DONE_RXAB_TRIO1_A**: IDAC calibration done for trio A1 RXAB.
2	0x0	**IDAC_CALIB_DONE_RXCA_TRIO0_A**: IDAC calibration done for trio A0 RXCA.
1	0x0	**IDAC_CALIB_DONE_RXBC_TRIO0_A**: IDAC calibration done for trio A0 RXBC.
0	0x0	**IDAC_CALIB_DONE_RXAB_TRIO0_A**: IDAC calibration done for trio A0 RXAB.

NVCSI_PHY_0_NVCSI_CIL_IDAC_CALIB_STATUS1_0

Offset: 0x4717
Byte Offset: 0x11c5c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x21084210 (0bxx10,0001,0000,1000,0100,0010,0001,0000)

Bit	Reset	Description
29:25	0x10	**RXCA_IDACADJ_TRIO1_A**: IDAC calibration result for trio A1 RXCA
24:20	0x10	**RXBC_IDACADJ_TRIO1_A**: IDAC calibration result for trio A1 RXBC
19:15	0x10	**RXAB_IDACADJ_TRIO1_A**: IDAC calibration result for trio A1 RXAB

Bit	Reset	Description
14:10	0x10	**RXCA_IDACADJ_TRIO0_A**: IDAC calibration result for trio A0 RXCA
9:5	0x10	**RXBC_IDACADJ_TRIO0_A**: IDAC calibration result for trio A0 RXBC
4:0	0x10	**RXAB_IDACADJ_TRIO0_A**: IDAC calibration result for trio A0 RXAB

NVCSI_PHY_0_NVCSI_CIL_IDAC_CALIB_STATUS2_0

Offset: 0x4718
Byte Offset: 0x11c60
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x21084210 (0bxx10,0001,0000,1000,0100,0010,0001,0000)

Bit	Reset	Description
29:25	0x10	**RXCA_IDACADJ_TRIO1_B**: IDAC calibration result for trio B1 RXCA
24:20	0x10	**RXBC_IDACADJ_TRIO1_B**: IDAC calibration result for trio B1 RXBC
19:15	0x10	**RXAB_IDACADJ_TRIO1_B**: IDAC calibration result for trio B1 RXAB
14:10	0x10	**RXCA_IDACADJ_TRIO0_B**: IDAC calibration result for trio B0 RXCA
9:5	0x10	**RXBC_IDACADJ_TRIO0_B**: IDAC calibration result for trio B0 RXBC
4:0	0x10	**RXAB_IDACADJ_TRIO0_B**: IDAC calibration result for trio B0 RXAB

NVCSI_PHY_0_NVCSI_CIL_IDAC_OVERRIDE_A_0

Offset: 0x4719
Byte Offset: 0x11c64
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x84218421 (0b1000,0100,0010,0001,1000,0100,0010,0001)

Bit	Reset	Description
31:27	0x10	**IDAC_VALUE_RXCA_TRIO1_A**: The override value
26:22	0x10	**IDAC_VALUE_RXBC_TRIO1_A**: The override value
21:17	0x10	**IDAC_VALUE_RXAB_TRIO1_A**: The override value
16	0x1	**SW_SET_IDAC_TRIO1_A**: Enable the SW override of the IDAC value
15:11	0x10	**IDAC_VALUE_RXCA_TRIO0_A**: The override value
10:6	0x10	**IDAC_VALUE_RXBC_TRIO0_A**: The override value
5:1	0x10	**IDAC_VALUE_RXAB_TRIO0_A**: The override value
0	0x1	**SW_SET_IDAC_TRIO0_A**: Enable the SW override of the IDAC value

NVCSI_PHY_0_NVCSI_CIL_IDAC_OVERRIDE_B_0

Offset: 0x471a
Byte Offset: 0x11c68
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x84218421 (0b1000,0100,0010,0001,1000,0100,0010,0001)

Bit	Reset	Description
31:27	0x10	**IDAC_VALUE_RXCA_TRIO1_B**: The override value
26:22	0x10	**IDAC_VALUE_RXBC_TRIO1_B**: The override value
21:17	0x10	**IDAC_VALUE_RXAB_TRIO1_B**: The override value
16	0x1	**SW_SET_IDAC_TRIO1_B**: Enable the SW override of the IDAC value
15:11	0x10	**IDAC_VALUE_RXCA_TRIO0_B**: The override value

Bit	Reset	Description
10:6	0x10	**IDAC_VALUE_RXBC_TRIO0_B**: The override value
5:1	0x10	**IDAC_VALUE_RXAB_TRIO0_B**: The override value
0	0x1	**SW_SET_IDAC_TRIO0_B**: Enable the SW override of the IDAC value

NVCSI_PHY_0_NVCSI_CIL_IDAC_CALIB_DEBUG_0

Offset: 0x471b
Byte Offset: 0x11c6c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000XXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
11	X	**ZI_HSRX_RXCA_TRIO1_B**
10	X	**ZI_HSRX_RXBC_TRIO1_B**
9	X	**ZI_HSRX_RXAB_TRIO1_B**
8	X	**ZI_HSRX_RXCA_TRIO0_B**
7	X	**ZI_HSRX_RXBC_TRIO0_B**
6	X	**ZI_HSRX_RXAB_TRIO0_B**
5	X	**ZI_HSRX_RXCA_TRIO1_A**
4	X	**ZI_HSRX_RXBC_TRIO1_A**
3	X	**ZI_HSRX_RXAB_TRIO1_A**
2	X	**ZI_HSRX_RXCA_TRIO0_A**
1	X	**ZI_HSRX_RXBC_TRIO0_A**
0	X	**ZI_HSRX_RXAB_TRIO0_A**

NVCSI_PHY_0_NVCSI_CIL_TEST_CONTROL_0

Offset: 0x471c
Byte Offset: 0x11c70

Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**HS_EN_DATA_LANE1_B**: Force to enable the e_input_hs of the data lane b1, used for impedance measurements of HS mode
4	0x0	**HS_EN_DATA_LANE0_B**: Force to enable the e_input_hs of the data lane b0, used for impedance measurements of HS mode
3	0x0	**HS_EN_DATA_LANE1_A**: Force to enable the e_input_hs of the data lane a1, used for impedance measurements of HS mode
2	0x0	**HS_EN_DATA_LANE0_A**: Force to enable the e_input_hs of the data lane a0, used for impedance measurements of HS mode
1	0x0	**HS_EN_CLK_LANE_B**: Force to enable the e_input_hs of the clock lane b, used for impedance measurements of HS mode
0	0x0	**HS_EN_CLK_LANE_A**: Force to enable the e_input_hs of the clock lane a, used for impedance measurements of HS mode

NVCSI_PHY_0_NVCSI_CIL_PULLDN_CONTROL_0

Offset: 0x471d
Byte Offset: 0x11c74
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**E_PULLDN_IO1_B**
4	0x0	**E_PULLDN_IO0_B**
3	0x0	**E_PULLDN_IO1_A**
2	0x0	**E_PULLDN_IO0_A**
1	0x0	**E_PULLDN_CLK_B**

Bit	Reset	Description
0	0x0	**E_PULLDN_CLK_A**

NVCSI_PHY_0_NVCSI_CIL_SPARE_0

Spare register
Offset: 0x471e
Byte Offset: 0x11c78
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**phy_reg**

NVCSI_PHY_0_NVCSI_CIL_A_SW_RESET_0

Offset: 0x471f
Byte Offset: 0x11c7c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**SW_RESET1_A**: SOFT RESET FOR LANE A1. Active High **1** = ENABLE **0** = DISABLE
0	0x0	**SW_RESET0_A**: SOFT RESET FOR LANE A0. Active High **1** = ENABLE **0** = DISABLE

NVCSI_PHY_0_NVCSI_CIL_A_CLKEN_OVERRIDE_CTRL_0

Offset: 0x4720
Byte Offset: 0x11c80
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**CLKEN_OVERRIDE1_A**: CLKENABLE OVERRIDE FOR LANE A1. Active High **1** = ENABLE **0** = DISABLE
0	0x0	**CLKEN_OVERRIDE0_A**: CLKENABLE OVERRIDE FOR LANE A0. Active High **1** = ENABLE **0** = DISABLE

NVCSI_PHY_0_NVCSI_CIL_A_CTLE_CTRL_0

Offset: 0x4721
Byte Offset: 0x11c84
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000888 (0bxxxx,xxxx,0000,0000,0000,1000,1000,1000)

Bit	Reset	Description
23:20	0x0	**AFE_DCGAIN_IO1_A**: RX AFE DC gain control
19:16	0x0	**AFE_DCGAIN_IO0_A**: RX AFE DC gain control
15:12	0x0	**AFE_DCGAIN_CLK_A**: RX AFE DC gain control
11:8	0x8	**AFE_HFGAIN_IO1_A**: RX AFE_CTLE high frequency gain control, 16 curves, ac_gain=code*1db
7:4	0x8	**AFE_HFGAIN_IO0_A**: RX AFE_CTLE high frequency gain control, 16 curves, ac_gain=code*1db

Bit	Reset	Description
3:0	0x8	**AFE_HFGAIN_CLK_A**: RX AFE_CTLE high frequency gain control, 16 curves, ac_gain=code*1db

NVCSI_PHY_0_NVCSI_CIL_A_CTLE_CTRL1_0

Offset: 0x4722
Byte Offset: 0x11c88
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000554 (0bxxxx,xxxx,0000,0000,0000,0101,0101,0100)

Bit	Reset	Description
23:20	0x0	**AFE_CTRL_IO1_A**
19:16	0x0	**AFE_CTRL_IO0_A**
15:12	0x0	**AFE_CTRL_CLK_A**
11:10	0x1	**AFE_CURRENT_IO1_A**
9:8	0x1	**AFE_CURRENT_IO0_A**
7:6	0x1	**AFE_CURRENT_CLK_A**
5:4	0x1	**AFE_FREQBAND_IO1_A**: RX AFE bandwidth selection. 2'b00: <1.5GS/s, 2'b01: 1.5~3GS/s, 2'b10: 3~5GS/s, 2'b11: reserved.
3:2	0x1	**AFE_FREQBAND_IO0_A**: RX AFE bandwidth selection. 2'b00: <1.5GS/s, 2'b01: 1.5~3GS/s, 2'b10: 3~5GS/s, 2'b11: reserved.
1:0	0x0	**AFE_FREQBAND_CLK_A**: RX AFE bandwidth selection. 2'b00: <1.5GS/s, 2'b01: 1.5~3GS/s, 2'b10: 3~5GS/s, 2'b11: reserved.

NVCSI_PHY_0_NVCSI_CIL_A_PAD_CONFIG_0

Offset: 0x4723
Byte Offset: 0x11c8c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00070000 (0b0000,0000,0000,x111,xxxx,0000,0000,0000)

Bit	Reset	Description
31	0x0	**LP_RX_SELECT_IO1_A**: LP mode receiver select for partition A lane 1. DPHY mode: controls IO1P_A and IO1N_A CPHY mode: controls IO1P_A, IO1N_A and IOCLKN_A.
30	0x0	**LP_RX_SELECT_IO0_A**: LP mode receiver select for partition A lane 0. DPHY mode: controls IO0P_A and IO0N_A CPHY mode: controls IO0P_A, IO0N_A and IOCLKP_A.
29	0x0	**LP_RX_SELECT_CLK_A**: LP mode receiver select. 0: Schmitt receiver. 1: self-biased receiver. DPHY mode: controls IOCLKP_A and IOCLKN_A. CPHY mode: dummy.
28	0x0	**E_LPRX_HYS_N_IO1_A**: Enable LP-RX hysteresis. 0: enable hysteresis 1: disable hysteresis
27	0x0	**E_LPRX_HYS_N_IO0_A**: Enable LP-RX hysteresis. 0: enable hysteresis 1: disable hysteresis
26	0x0	**E_LPRX_HYS_N_CLK_A**: Enable LP-RX hysteresis. 0: enable hysteresis 1: disable hysteresis
25	0x0	**E_HS_TERM_N_IO1_A**: Enable the HS termination of Partition A Lane 1.
24	0x0	**E_HS_TERM_N_IO0_A**: Enable the HS termination of Partition A Lane 0.
23	0x0	**E_HS_TERM_N_CLK_A**: Enable the HS termination of Clock partition A.
22	0x0	**E_INPUT_LP_IO1_A**: Enable LP receiver of Partition A Lane 1 Applicable in both CPHY and DPHY case.
21	0x0	**E_INPUT_LP_IO0_A**: Enable LP receiver of Partition A Lane 0 Applicable in both CPHY and DPHY case.
20	0x0	**E_INPUT_LP_CLK_A**: enable LP receiver of Clock partition A Applicable in DPHY case. N/A for CPHY
18	0x1	**PD_CLK_A**: Power down for CLk of Partition A. Applicable in DPHY case. N/A for CPHY
17	0x1	**PD_IO1_A**: Power down for Trio 1 and Lane 1 of Partition A. Applicable in both CPHY and DPHY case.

Bit	Reset	Description
16	0x1	**PD_IO0_A**: Power down for Trio 0 and Lane 0 of Partition A. Applicable in both CPHY and DPHY case.
11:8	0x0	**SPARE_CLK_A**: Spare control bits for pad control. Bit 11 is used as E_PULLDN_CLK_A control (0 - Pull down of Clock Lane enabled, 1 - Pull down of Clock Lane enabled). Functionality of other bits currently not defined. Undefined register bits are reserved for future use. Pull down may be enabled when Clock Lane is unused to minimized interference.
7:4	0x0	**SPARE_IO1_A**: Spare control bits for pad control. Functionality of other bits currently not defined. Undefined register bits are reserved for future use. Pull down may be enabled when Data Lane-1 is unused to minimized interference.
3:0	0x0	**SPARE_IO0_A**: Spare control bits for pad control. Functionality of other bits currently not defined. Undefined register bits are reserved for future use. Pull down may be enabled when Data Lane-0 is unused to minimized interference.

NVCSI_PHY_0_NVCSI_CIL_A_PAD_CONFIG_1_0

Offset: 0x4724
Byte Offset: 0x11c90
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000002 (0b0000,0000,0000,0000,0000,0000,xx00,0010)

Bit	Reset	Description
31:24	0x0	**OUTADJ_IO1_A**: Trio 1 output delay trimmer, each tap delays 9ps
23:16	0x0	**OUTADJ_IO0_A**: Trio 0 output delay trimmer, each tap delays 9ps
15:8	0x0	**OUTADJ_CLK_A**: Clock bit output delay trimmer, each tap delays 9ps
5	0x0	**HS_BSO_A**: 1= power-on 0.4v-TX-regulator during LP-mode; 0= power-off 0.4v-TX-regulator during LP-mode;
4	0x0	**CPHY_MID_STRENGTH_TRIO1_A**: 0: when bump is driven to mid-voltage-level, both 100-ohm-pull-up and 100-ohm-pull-down is turned on to drive mid-voltage-level; 1: when bump is driven to mid-voltage-level, both pull-up and pull-down are turned off to become tri-stated;

Bit	Reset	Description
3	0x0	**CPHY_MID_STRENGTH_TRIO0_A**: 0: when bump is driven to mid-voltage-level, both 100-ohm-pull-up and 100-ohm-pull-down is turned on to drive mid-voltage-level; 1: when bump is driven to mid-voltage-level, both pull-up and pull-down are turned off to become tri-stated;
2	DISABLE	**REV_CLK_A**: Reverse clock polarity **0** = DISABLE **1** = ENABLE
1	ENABLE	**E_PREDRV_FLOP_RESET_A**: HS predriver flop reset **0** = DISABLE **1** = ENABLE
0	DISABLE	**FCZERO_A**: Force clk bit to drive differential 0 **0** = DISABLE **1** = ENABLE

NVCSI_PHY_0_NVCSI_CIL_A_PAD_CONFIG_2_0

Offset: 0x4725
Byte Offset: 0x11c94
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,0000,0000,0000,xxxx,0000,0000,0000)

Bit	Reset	Description
27:24	0x0	**PEMPD_IO1_A**: Enable trio 1 HS driver pull down pre-emphasis.
23:20	0x0	**PEMPD_IO0_A**: Enable trio 0 HS driver pull down pre-emphasis.
19:16	0x0	**PEMPD_CLK_A**: Enable clock bit HS driver pull down pre-emphasis.
11:8	0x0	**PEMPU_IO1_A**: Enable trio 1 HS driver pull up pre-emphasis.
7:4	0x0	**PEMPU_IO0_A**: Enable trio 0 HS driver pull up pre-emphasis.
3:0	0x0	**PEMPU_CLK_A**: Enable clock bit HS driver pull up pre-emphasis.

NVCSI_PHY_0_NVCSI_CIL_A_PAD_CONFIG_3_0

Offset: 0x4726
Byte Offset: 0x11c98
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x06660666 (0bxxxx,0110,0110,0110,xxxx,0110,0110,0110)

Bit	Reset	Description
27:24	0x6	**LPDNADJ_IO1_A**: Driver pull down impedance control
23:20	0x6	**LPDNADJ_IO0_A**: Driver pull down impedance control
19:16	0x6	**LPDNADJ_CLK_A**: Driver pull down impedance control
11:8	0x6	**LPUPADJ_IO1_A**: Driver pull up impedance control
7:4	0x6	**LPUPADJ_IO0_A**: Driver pull up impedance control
3:0	0x6	**LPUPADJ_CLK_A**: Driver pull up impedance control

NVCSI_PHY_0_NVCSI_CIL_A_PAD_CONFIG_4_0

Offset: 0x4727
Byte Offset: 0x11c9c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,0000,0000,0000,xxxx,0000,0000,0000)

Bit	Reset	Description
27:24	0x0	**SLEWDNADJ_IO1_A**: Pull down slew rate adjust
23:20	0x0	**SLEWDNADJ_IO0_A**: Pull down slew rate adjust
19:16	0x0	**SLEWDNADJ_CLK_A**: Pull down slew rate adjust

Bit	Reset	Description
11:8	0x0	**SLEWUPADJ_IO1_A**: Pull up slew rate adjust
7:4	0x0	**SLEWUPADJ_IO0_A**: Pull up slew rate adjust
3:0	0x0	**SLEWUPADJ_CLK_A**: Pull up slew rate adjust

NVCSI_PHY_0_NVCSI_CIL_A_PAD_CONFIG_5_0

Offset: 0x4728
Byte Offset: 0x11ca0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000110 (0bxxxx,xxxx,xxxx,xxxx,xxxx,x001,0001,0000)

Bit	Reset	Description
10:7	0x2	**CDDNADJ_IO1_A**: Level adjust on low limit of detection
6:3	0x2	**CDDNADJ_IO0_A**: Level adjust on low limit of detection
2	DISABLE	**E_CD_IO1_A**: Data bits contention detector enable **0** = DISABLE **1** = ENABLE
1	DISABLE	**E_CD_IO0_A**: Data bits contention detector enable **0** = DISABLE **1** = ENABLE
0	DISABLE	**E_CD_CLK_A**: Clock bit contention detector enable **0** = DISABLE **1** = ENABLE

NVCSI_PHY_0_NVCSI_CIL_A_PAD_CONFIG_6_0

Offset: 0x4729
Byte Offset: 0x11ca4
Read/Write: R/W

Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**E_DEEP_LPBK_A**: Enable deep loopback
0	0x0	**E_SHALLOW_LPBK_A**: Enalbe shallow loopback

NVCSI_PHY_0_NVCSI_CIL_A_PAD_CD_STATUS_0

Offset: 0x472a
Byte Offset: 0x11ca8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**CDN_IO1_A**: trio1 n output, 1=contention detected.
4	0x0	**CDN_IO0_A**: trio0 n output, 1=contention detected.
3	0x0	**CDN_CLK_A**: Clock bit n output, 1=contention detected.
2	0x0	**CDP_IO1_A**: trio1 p output, 1=contention detected.
1	0x0	**CDP_IO0_A**: trio0 p output, 1=contention detected.
0	0x0	**CDP_CLK_A**: Clock bit p output, 1=contention detected.

NVCSI_PHY_0_NVCSI_CIL_A_DPHY_INADJ_CTRL_0

Offset: 0x472b
Byte Offset: 0x11cac
Read/Write: R/W
Parity Protection: Y

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,x000,0000,x000,0000,x000,0000)

Bit	Reset	Description
22	0x0	**SW_SET_DPHY_INADJ_CLK_A**: Software set for clock input delay trimmer
21:16	0x0	**DPHY_INADJ_CLK_A**: Programmable value for CLK input delay trimmer,
14	0x0	**SW_SET_DPHY_INADJ_IO1_A**: Software override for bit 1 input delay trimmer 1 - SW programmed value is applied to the pads. 2 - Trimmer value from HW calibration state machine is applied to the pads
13:8	0x0	**DPHY_INADJ_IO1_A**: Programmable value for bit 1 input delay trimmer, each tap delay of 9 ps
6	0x0	**SW_SET_DPHY_INADJ_IO0_A**: Software override for bit 0 input delay trimmer 1 - SW programmed value is applied to the pads. 2 - Trimmer value from HW calibration state machine is applied to the pads
5:0	0x0	**DPHY_INADJ_IO0_A**: Programmable value for bit 0 input delay trimmer,

NVCSI_PHY_0_NVCSI_CIL_A_CLK_DESKEW_CTRL_0

Offset: 0x472c
Byte Offset: 0x11cb0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000bf00 (0bxxxx,xxxx,xxxx,xxxx,1x11,1111,xx00,0000)

Bit	Reset	Description
15	ENABLE	**CLK_INADJ_SWEEP_CTRL_A**: Enable the clock trimmer sweep for D-PHY de-skew. **0** = DISABLE **1** = ENABLE
13:8	0x3f	**CLK_INADJ_LIMIT_HIGH_A**: The upper limit of the sweep range for trimmer sweep.
5:0	0x0	**CLK_INADJ_LIMIT_LOW_A**: The lower limit of the sweep range for trimmer sweep.

NVCSI_PHY_0_NVCSI_CIL_A_DATA_DESKEW_CTRL_0

Offset: 0x472d
Byte Offset: 0x11cb4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x9f80bf00 (0b1xx1,1111,1x00,0000,1x11,1111,xx00,0000)

Bit	Reset	Description
31	0x1	**DATA_INADJ_SWEEP_CTRL1_A**: Enable the data lane A1 trimmer sweep for D-PHY de-skew.
28:23	0x3f	**DATA_INADJ_LIMIT_HIGH1_A**: The upper limit of the sweep range for trimmer sweep.
21:16	0x0	**DATA_INADJ_LIMIT_LOW1_A**: The lower limit of the sweep range for trimmer sweep.
15	0x1	**DATA_INADJ_SWEEP_CTRL0_A**: Enable the data lane A0 trimmer sweep for D-PHY de-skew.
13:8	0x3f	**DATA_INADJ_LIMIT_HIGH0_A**: The upper limit of the sweep range for trimmer sweep.
5:0	0x0	**DATA_INADJ_LIMIT_LOW0_A**: The lower limit of the sweep range for trimmer sweep.

NVCSI_PHY_0_NVCSI_CIL_A_DPHY_DESKEW_STATUS_0

Offset: 0x472e
Byte Offset: 0x11cb8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000X0XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
15	X	**DPHY_CALIB_ERR_IO1_A**: calib error status
14	X	**DPHY_CALIB_DONE_IO1_A**: calib done status
7	X	**DPHY_CALIB_ERR_IO0_A**: calib error status

Bit	Reset	Description
6	X	**DPHY_CALIB_DONE_IO0_A**: calib done status
1	X	**DPHY_CALIB_ERR_CTRL_A**: calib error status
0	X	**DPHY_CALIB_DONE_CTRL_A**: calib done status

NVCSI_PHY_0_NVCSI_CIL_A_DPHY_DESKEW_DATA_CALIB_STATUS_LOW_0_0

Offset: 0x472f
Byte Offset: 0x11cbc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_A**

NVCSI_PHY_0_NVCSI_CIL_A_DPHY_DESKEW_DATA_CALIB_STATUS_HIGH_0_0

Offset: 0x4730
Byte Offset: 0x11cc0
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_A**

NVCSI_PHY_0_NVCSI_CIL_A_DPHY_DESKEW_DATA_CALIB_STATUS_LOW_1_0

Offset: 0x4731
Byte Offset: 0x11cc4
Read/Write: RO
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_A**

NVCSI_PHY_0_NVCSI_CIL_A_DPHY_DESKEW_DATA_CALIB_STATUS_HIGH_1_0

Offset: 0x4732
Byte Offset: 0x11cc8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_A**

NVCSI_PHY_0_NVCSI_CIL_A_DPHY_DESKEW_CLK_CALIB_STATUS_LOW_0_0

Offset: 0x4733
Byte Offset: 0x11ccc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_A**

NVCSI_PHY_0_NVCSI_CIL_A_DPHY_DESKEW_CLK_CALIB_STATUS_HIGH_0_0

Offset: 0x4734
Byte Offset: 0x11cd0
Read/Write: RO
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_A**

NVCSI_PHY_0_NVCSI_CIL_A_DPHY_DESKEW_CLK_CALIB_STATUS_LOW_1_0

Offset: 0x4735
Byte Offset: 0x11cd4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_A**

NVCSI_PHY_0_NVCSI_CIL_A_DPHY_DESKEW_CLK_CALIB_STATUS_HIGH_1_0

Offset: 0x4736
Byte Offset: 0x11cd8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_A**

NVCSI_PHY_0_NVCSI_CIL_A_DPHY_DESKEW_RESULT_STATUS_0

Offset: 0x4737
Byte Offset: 0x11cdc
Read/Write: R/W
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	DONE	**DESKEW_RESULT_STATUS1_A**: For lane A1. The result status of the deskew calibration, HW check this field before start the trimmer sweep, if it's 1'b1, the trimmer sweep will not perform. HW set this bit to 1 when the calibration is done. SW should clear this field after it read out the 2 64 bits calibration status. **0** = DONE **1** = PENDING
0	DONE	**DESKEW_RESULT_STATUS0_A**: For lane A0. The result status of the deskew calibration, HW check this field before start the trimmer sweep, if it's 1'b1, the trimmer sweep will not perform. HW set this bit to 1 when the calibration is done. SW should clear this field after it read out the 2 64 bits calibration status. **0** = DONE **1** = PENDING

NVCSI_PHY_0_NVCSI_CIL_A_POLARITY_SWIZZLE_CTRL_0

Offset: 0x4738
Byte Offset: 0x11ce0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xx00,0000,xxxx,x000)

Bit	Reset	Description
13:11	0x0	**POLARITY_SWIZZLE_CPHY1_A**: Polarity Swizle control for Lane A1 in CPHY mode. valid only in CPHY mode 000 - A B C --> A B C 001 - A B C --> A C B 010 - A B C --> B C A 011 - A B C --> B A C 100 - A B C --> C A B 101 - A B C --> C B A
10:8	0x0	**POLARITY_SWIZZLE_CPHY0_A**: Polarity Swizle control for Lane A0 in CPHY mode. valid only in CPHY mode 000 - A B C --> A B C 001 - A B C --> A C B 010 - A B C --> B C A 011 - A B C --> B A C 100 - A B C --> C A B 101 - A B C --> C B A

Bit	Reset	Description
2	0x0	**POLARITY_SWIZZLE_CLK_A**: Polarity Swizzle control for clock lane A. Valid only in DPHY mode, need to work with ALLOW_FIRST_BIT_ON_CLK_NEGEDGE if enabled.
1	0x0	**POLARITY_SWIZZLE_DPHY1_A**: Polarity Swizzle control for data lane A1. Valid only in DPHY mode.
0	0x0	**POLARITY_SWIZZLE_DPHY0_A**: Polarity Swizzle control for data lane A0. Valid only in DPHY mode.

NVCSI_PHY_0_NVCSI_CIL_A_DESKEW_CONTROL_0

Offset: 0x4739
Byte Offset: 0x11ce4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00007411 (0bxxxx,xxxx,xxxx,xxxx,0111,0100,0001,0001)

Bit	Reset	Description
15:13	0x3	**DESKEW_DATA_DELAY**: How many cycles delay introduced by re-timing flops
12	0x1	**DESKEW_LANE_SKEW_CHECK_EN**: Check the skew between the lanes, this is used to handle the error case that error happen in the CSI link caused some data lanes detect the HS sync word (0xb8) while other data lanes detect the deskew sync word (0xff). If this bit is not set with the error case, the lane FIFO will overflow and reset is required. With this bit set, the FIFO will be auto flused and the FIFO overflow will not happen.
11:8	0x4	**DESKEW_LANE_SKEW_TOLERANCE**: The allowed skew between the data lane, only used in sync word search
7:4	0x1	**DESKEW_COMPARE**: Regsiter select to control the number of comparisons to be done for each trimmer setting during deskew calibration.
3:0	0x1	**DESKEW_SETTLE**: Regsiter select to control the number of byte clocks to wait before INADJ value settles. This is used during deskew calibration

NVCSI_PHY_0_NVCSI_CIL_A_CONTROL_0

Offset: 0x473a
Byte Offset: 0x11ce8
Read/Write: R/W
Parity Protection: Y

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**BYPASS_LP_FILTER**: Bypass the glitch filter.
30:24	0x0	**CLK_SETTLE1_A**: Used only in CPHY mode, Settle time for clk start when moving A1 data lane from LP to HS (LP11->LP01->LP00).
23:17	0x0	**CLK_SETTLE0_A**: Used for both CPHY/DPY mode. For DPHY: Settle time for clk lane A0 when moving from LP to HS (LP11->LP01->LP00), this setting determines how many csicil clock cycles to wait after LP00. If the settle set to 0 or 8, the real setting is 1, otherwise, the real setting is (4+settle) For CPHY: Settle time for clk start when moving A0 data lane from LP to HS (LP11->LP01->LP00).
16:9	0x0	**THS_SETTLE1_A**: settle time for A1 data lane when moving from LP to HS (LP11->LP01->LP00), this setting determines how many LP clock cycles to wait, after LP00,before starting to look at the data.
8:1	0x0	**THS_SETTLE0_A**: settle time for A0 data lane when moving from LP to HS (LP11->LP01->LP00), this setting determines how many LP clock cycles to wait, after LP00,before starting to look at the data.
0	0x0	**BYPASS_LP_SEQ**: For DPHY: The clock lane LP signals should sequence through LP11->LP01->LP00 state, to indicate to CLOCK CIL about the mode switching to HS Rx mode. In case Camera is enabled earlier than CIL , it is highly likely that camera sends this control sequence sooner than cil can detect it. Enabling this bit allows the CLOCK CIL to overlook the LP control sequence and step in HS Rx mode directly looking at LP00 only. For CPHY: it's only used for debug purpose as it's embedded clock for CPHY, clock will always presents when the data lane toggle.

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_ERR_STATUS_0

Offset: 0x473b
Byte Offset: 0x11cec
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
29:24	X	**LANE_DEMAPPER_ERR_MUX1_A**: Debug register, error detected in CPHY de-mapper mux.
23	X	**LANE_DEMAPPER_ERR_RXCTRL1_A**: Debug register, Error detected in CPHY de-mapper rx ctrl.
22:16	X	**LANE_DECODER_ERR1_A**: Debug register, Error detected in CPHY de-coder.
13:8	X	**LANE_DEMAPPER_ERR_MUX0_A**: Debug register, Error detected in CPHY de-mapper mux.
7	X	**LANE_DEMAPPER_ERR_RXCTRL0_A**: Debug register, Error detected in CPHY de-mapper rx ctrl.
6:0	X	**LANE_DECODER_ERR0_A**: Debug register, Error detected in CPHY de-coder.

NVCSI_PHY_0_NVCSI_CIL_A_ESCAPE_MODE_COMMAND_0_0

Offset: 0x473c
Byte Offset: 0x11cf0
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_COMMAND0_A**: The received command from escape mode.

NVCSI_PHY_0_NVCSI_CIL_A_ESCAPE_MODE_DATA_0_0

Offset: 0x473d
Byte Offset: 0x11cf4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_DATA0_A**: The received data from excape mode.

NVCSI_PHY_0_NVCSI_CIL_A_ESCAPE_MODE_COMMAND_1_0

Offset: 0x473e
Byte Offset: 0x11cf8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_COMMAND1_A**: The received command from escape mode.

NVCSI_PHY_0_NVCSI_CIL_A_ESCAPE_MODE_DATA_1_0

Offset: 0x473f
Byte Offset: 0x11cfc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_DATA1_A**: The received data from excape mode.

NVCSI_PHY_0_NVCSI_CIL_A_DPHY_SYNC_PATTERN_0

Offset: 0x4740
Byte Offset: 0x11d00
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01)

Bit	Reset	Description
1	0x0	**ALLOW_FIRST_BIT_ON_CLK_NEGEDGE_A**: Allow shift 1/3/5/7 bit to search the sync word. Normal case only shift 0/2/4/6 bit to search sync word.
0	0x1	**DISABLE_SB_ERR_IN_SYNC_A**: Allow one single bit error in sync word 6MSB

NVCSI_PHY_0_NVCSI_CIL_A_DPHY_DESKEW_SYNC_PATTERN_0

Offset: 0x4741
Byte Offset: 0x11d04
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**INVERT_DESKEW_PATTERN**: Invert the de-skew pattern.

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_SYNC_SEARCH_0

Offset: 0x4742
Byte Offset: 0x11d08
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01080108 (0bxxxx,xxx1,0000,1000,xxxx,xxx1,0000,1000)

Bit	Reset	Description
24	0x1	**COUNT_EN_TRIO1_A**: Check if the symbol is pre-amble before sync word.
23:16	0x8	**PERIOD_TRIO1_A**: For CPHY, how many cycles allowed for non pre-amble symbol before sync word.

Bit	Reset	Description
8	ENABLE	**COUNT_EN_TRIO0_A**: Check if the symbol is pre-amble before sync word. **0** = DISABLE **1** = ENABLE
7:0	0x8	**PERIOD_TRIO0_A**: For CPHY, how many cycles allowed for non pre-amble symbol before sync word.

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_START_0

Offset: 0x4743
Byte Offset: 0x11d0c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**EDGE_DELAY_CAL_TRIGGER_TRIO1_A**: Trigger the CPHY clock recover calibration for lane A1 by write 1'b1, when the calibration complete, HW will clear this bit to 1'b0. Write 1'b0 to this be take no effect.
0	0x0	**EDGE_DELAY_CAL_TRIGGER_TRIO0_A**: Trigger the CPHY clock recover calibration for lane A0 by write 1'b1, when the calibration complete, HW will clear this bit to 1'b0. Write 1'b0 to this be take no effect.

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_CTRL_0

Offset: 0x4744
Byte Offset: 0x11d10
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x12011210 (0b0001,0010,0000,0001,0001,0010,0001,0000)
PROD: 0x08000800 (0bxxx0,10xx,xxxx,xxxx,xxx0,10xx,xxxx,xxxx)

Bit	Reset	PROD	Description
31	0x0	_NONE_	**TRIM_CAL_E_NONSTOP_TRIO1_A**: Reserved. Trimmer calibration background monitoring control. 1: enable. The accumulator keeps updating.
30:29	0x0	_NONE_	**TRIM_CAL_GAIN_SEL_TRIO1_A**: Trimmer calibration loop gain selection [0]: gain control 0: coarse tune (counter1) gain = 1, fine tune (counter2) gain = 1/4. For example: coarse tune: 1 UP -> trim_cal_code + 1; fine tune: 4 UPs -> trim_cal_code + 1. 1: coarse tune (counter1) gain = 1/2, fine tune (counter2) gain = 1/8 [1]: reserved
28:26	0x4	0x2	**TRIM_CAL_CNT_SEL_TRIO1_A**: Trimmer calibration counter selection [0]: counter1 (coarse tune) clock cycle. 0: 64T (128UI) 1: 128T (256UI) [2:1]: counter2 (fine tune) clock cycle. 00: 32T (64UI) 01: 64T (128UI) 10: 96T (192UI) 11: 128T (256UI) For high data rate (>3.5GS/s): 64T(cnt1)+96T(cnt2)=160T(320ui) -> cnt_sel=3'b100 For low data rate (<=3.5GS/s): 128T(cnt1)+32T(cnt2)=160T(320ui) -> cnt_se=3'b001
25	0x1	_NONE_	**TRIM_CAL_CPHY_ED_LOOPMODE_TRIO1_A**: Trimmer calibration and cphy edge detector loop mode control 0: exclude trimmer delay in 2nd round of reset loop. 1: include trimmer delay in 2nd round of reset loop. For low data rate.
24:16	0x1	_NONE_	**CALIB_LENGTH_TRIO1_A**: //The length of the peramble which use to do caliration. Count by nvcsi core clock. It's used to make sure the calibration is doing inside the preamble phase. If this length set to a small value, the calibration may can't complete within one packet and need several packets to complete.
15	0x0	_NONE_	**TRIM_CAL_E_NONSTOP_TRIO0_A**: Reserved. Trimmer calibration background monitoring control. 1: enable. The accumulator keeps updating.
14:13	0x0	_NONE_	**TRIM_CAL_GAIN_SEL_TRIO0_A**: Trimmer calibration loop gain selection [0]: gain control 0: coarse tune (counter1) gain = 1, fine tune (counter2) gain = 1/4. For example: coarse tune: 1 UP -> trim_cal_code + 1; fine tune: 4 UPs -> trim_cal_code + 1. 1: coarse tune (counter1) gain = 1/2, fine tune (counter2) gain = 1/8 [1]: reserved

Bit	Reset	PROD	Description
12:10	0x4	0x2	**TRIM_CAL_CNT_SEL_TRIO0_A**: Trimmer calibration counter selection [0]: counter1 (coarse tune) clock cycle. 0: 64T (128UI) 1: 128T (256UI) [2:1]: counter2 (fine tune) clock cycle. 00: 32T (64UI) 01: 64T (128UI) 10: 96T (192UI) 11: 128T (256UI) For high data rate (>3.5GS/s): 64T(cnt1)+96T(cnt2)=160T(320ui) -> cnt_sel=3'b100 For low data rate (<=3.5GS/s): 128T(cnt1)+32T(cnt2)=160T(320ui) -> cnt_se=3'b001
9	0x1	_NONE_	**TRIM_CAL_CPHY_ED_LOOPMODE_TRIO0_A**: Trimmer calibration and cphy edge detector loop mode control 0: exclude trimmer delay in 2nd round of reset loop. 1: include trimmer delay in 2nd round of reset loop. For low data rate.
8:0	0x10	_NONE_	**CALIB_LENGTH_TRIO0_A**: The length of the peramble which use to do caliration. Count by nvcsi core clock. It's used to make sure the calibration is doing inside the preamble phase. If this length set to a small value, the calibration may can't complete within one packet and need several packets to complete.

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_CTRL1_0

Offset: 0x4745
Byte Offset: 0x11d14
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0c0e0c0e (0bx000,1100,0000,1110,x000,1100,0000,1110)

Bit	Reset	Description
30	0x0	**PERIODIC_CALIB_TRIO1_A**: Enable the periodic calibration.
29	0x0	**CALIB_APPLY_IN_POST_TRIO1_A**: 1: Apply the calibration result when detect post symbol. 0: Apply the calibration result immediately when calibration done.

Bit	Reset	Description
28:27	0x1	**CALIB_APPLY_SELECT_TRIO1_A**: Which calibration result is applied to pad for dual calibration. 2'b11: Reserved 2'b10: Apply the middle value of two calibration to pad if the difference within threshold. 2'b01: Apply the 2nd calibration result to pad if it's difference with previous latch result within threshold. 2'b00: Apply the 1st calibration result to pad if it's difference with previous latch result within threshold.
26	0x1	**DUAL_CALIB_TRIO1_A**: Two times calibratiton. 1: Two time calibration with one trigger. 0: Single time calibration with one trigger
25:18	0x3	**TEMP_EDGE_DELAY_VALUE_TRIO1_A**: The edge delay value while doing the calibration.
17	0x1	**TEMP_EDGE_DELAY_SEL_TRIO1_A**: When the selection of edge delay value when do edge delay calibration. 1: Use the temp value from register setting. 0: Use the currently using setting. For the first calibration after reset, it's still using the temp value from register setting.
16	0x0	**BYPASS_CALIB_PACKET_TRIO1_A**: 1: Drop the packet data when do the calibration. 0: Not drop the packet data when do the calibration.
14	0x0	**PERIODIC_CALIB_TRIO0_A**: Enable the periodic calibration.
13	0x0	**CALIB_APPLY_IN_POST_TRIO0_A**: 1: Apply the calibration result when detect post symbol. 0: Apply the calibration result immediately when calibration done.
12:11	0x1	**CALIB_APPLY_SELECT_TRIO0_A**: Which calibration result is applied to pad for dual calibration. 2'b11: Reserved 2'b10: Apply the middle value of two calibration to pad if the difference within threshold. 2'b01: Apply the 2nd calibration result to pad if it's difference with previous latch result within threshold. 2'b00: Apply the 1st calibration result to pad if it's difference with previous latch result within threshold.
10	0x1	**DUAL_CALIB_TRIO0_A**: Two times calibratiton. 1: Two time calibration with one trigger. 0: Single time calibration with one trigger
9:2	0x3	**TEMP_EDGE_DELAY_VALUE_TRIO0_A**: The edge delay value apply to pad while doing the calibration.
1	0x1	**TEMP_EDGE_DELAY_SEL_TRIO0_A**: When the selection of edge delay value when do edge delay calibration. 1: Use the temp value from register setting. 0: Use the currently using setting. For the first calibration after reset, it's still using the temp value from register setting.
0	0x0	**BYPASS_CALIB_PACKET_TRIO0_A**: 1: Drop the packet data when do the calibration. 0: Not drop the packet data when do the calibration.

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_CTRL2_0

Offset: 0x4746
Byte Offset: 0x11d18
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01000100 (0bxxxx,xxx1,0000,0000,xxxx,xxx1,0000,0000)

Bit	Reset	Description
24	ENABLE	**TRIM_CAL_PD_HOLD_TRIO1_A**: Hold in PD **0** = DISABLE **1** = ENABLE
23	DISABLE	**TRIM_CAL_E_FIXEDDLY_TRIO1_A**: Enable fixed delay in trim calibration unit. **0** = DISABLE **1** = ENABLE
22:16	0x0	**TRIM_CAL_FIXEDDLY_TRIO1_A**: Adjust fixed delay. Delay = code*5ps(TT)/7ps(SS)
8	ENABLE	**TRIM_CAL_PD_HOLD_TRIO0_A**: Hold in PD **0** = DISABLE **1** = ENABLE
7	DISABLE	**TRIM_CAL_E_FIXEDDLY_TRIO0_A**: Enable fixed delay in trim calibration unit. **0** = DISABLE **1** = ENABLE
6:0	0x0	**TRIM_CAL_FIXEDDLY_TRIO0_A**: Adjust fixed delay. Delay = code*5ps(TT)/7ps(SS)

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_CTRL3_0

Offset: 0x4747
Byte Offset: 0x11d1c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffffffff	**CALIB_PERIOD_TRIO0_A**: The period of the periodic edge delay calibration, in nvcsi clock cycle number.

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_CTRL4_0

Offset: 0x4748
Byte Offset: 0x11d20
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffffffff	**CALIB_PERIOD_TRIO1_A**: The period of the periodic edge delay calibration, in nvcsi clock cycle number.

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_CTRL5_0

Offset: 0x4749
Byte Offset: 0x11d24
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000ffff (0bxxxx,xxxx,xxxx,xxxx,1111,1111,1111,1111)

Bit	Reset	Description
15:8	0xff	**ERR_THRESHOLD_TRIO1_A**: The threshold of drop the calibration result between current result and previous result. For dual calibration, check between the two calibration result in the same trigger. For signal calibration, check between the current calibration result and previous calibration result.
7:0	0xff	**ERR_THRESHOLD_TRIO0_A**: The threshold of drop the calibration result between current result and previous resulti. For dual calibration, check between the two calibration result in the same trigger. For signal calibration, check between the current calibration result and previous calibration result.

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_STATUS_0

Offset: 0x474a
Byte Offset: 0x11d28
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,xxxx)

Bit	Reset	Description
9	0x0	**EDGE_DELAY_MISMATCH_TRIO1_A**: When the CALIB_APPLY_SELECT_TRIO is 0/1 or dual time calibration is not enabled, the calibration result difference from the previous latched calibration result is larger than threshold, this time calibration result is not apply to pad. When the CALIB_APPLY_SELECT_TRIO is 2, the calibration result difference of the two times calibration result is larger than threshold, this time calibration result is not apply to pad.
8	0x0	**EDGE_DELAY_MISMATCH_TRIO0_A**: When the CALIB_APPLY_SELECT_TRIO is 0/1 or dual time calibration is not enabled, the calibration result difference from the previous latched calibration result is larger than threshold, this time calibration result is not apply to pad. When the CALIB_APPLY_SELECT_TRIO is 2, the calibration result difference of the two times calibration result is larger than threshold, this time calibration result is not apply to pad.
7	0x0	**CALIB_DONE_1_TRIO1_A**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.
6	0x0	**CALIB_DONE_1_TRIO0_A**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.
5	0x0	**CALIB_DONE_0_TRIO1_A**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.
4	0x0	**CALIB_DONE_0_TRIO0_A**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_EDGE_DELAY_CTRL_0

Offset: 0x474b
Byte Offset: 0x11d2c
Read/Write: R/W
Parity Protection: See table below
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x40000000 (0b01xx,x000,0000,0000,xxxx,x000,0000,0000)

Bit	Parity Protection	Reset	Description
31:30	Y	0x1	**CPHY_CLKGEN_TRIGWIDTH_A**: Adjust AB/CB/CA pulse width in self-clock generator.
26	N	0x0	**UPDATE_OFFSET_TRIO1_A**: Trigger to apply the new offset value
25:17	Y	0x0	**DELAY_OFFSET_TRIO1_A**: Offset value of trio A1. Represents signed 2's complement number for calibration offset. Represents unsigned number if the field used as override value.
16	Y	0x0	**OVERRIDE_CAL_VAL_TRIO1_A**: Override for trio A1 0: Apply the DELAY_OFFSET as an offset to the calibration result. 1: Apply the DELAY_OFFSET value to the pad.
10	N	0x0	**UPDATE_OFFSET_TRIO0_A**: Trigger to apply the new offset value
9:1	Y	0x0	**DELAY_OFFSET_TRIO0_A**: Offset value of trio A0. Represents signed 2's complement number for calibration offset. Represents unsigned number if the field used as override value.
0	Y	0x0	**OVERRIDE_CAL_VAL_TRIO0_A**: Override for trio A0 0: Apply the DELAY_OFFSET as an offset to the calibration result. 1: Apply the DELAY_OFFSET value to the pad.

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_EDGE_DELAY_VALUE_0

Offset: 0x474c
Byte Offset: 0x11d30
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000303 (0bxxxx,xxxx,xxxx,xxxx,0000,0011,0000,0011)

Bit	Reset	Description
15:8	0x3	**EDGE_DELAY_VALUE_TRIO1_A**: The edge delay value apply to pad finally
7:0	0x3	**EDGE_DELAY_VALUE_TRIO0_A**: The edge delay value apply to pad finally

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_RESULT_0

Offset: 0x474d
Byte Offset: 0x11d34
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**EDGE_DELAY_CALIB_RESULT_1_TRIO1_A**: The edge delay calibrate result, with offset.
23:16	0x0	**EDGE_DELAY_CALIB_RESULT_1_TRIO0_A**: The edge delay calibrate result, with offset.
15:8	0x0	**EDGE_DELAY_CALIB_RESULT_0_TRIO1_A**: The edge delay calibrate result, with offset.
7:0	0x0	**EDGE_DELAY_CALIB_RESULT_0_TRIO0_A**: The edge delay calibrate result, with offset.

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_INADJ_CALIB_CTRL_0

Offset: 0x474e
Byte Offset: 0x11d38
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
16	0x0	**INADJ_CAL_TRIGGER_TRIO1_A**: Trigger INADJ calibration for trio0
0	0x0	**INADJ_CAL_TRIGGER_TRIO0_A**: Trigger INADJ calibration for trio1

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_INADJ_CALIB_CTRL1_0

Offset: 0x474f
Byte Offset: 0x11d3c

Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00010001 (0b0000,0000,0000,0001,0000,0000,0000,0001)

Bit	Reset	Description
31:16	0x1	**PRBS9_SEED_TRIO1_A**: PRBS9 initial seed for trio0
15:0	0x1	**PRBS9_SEED_TRIO0_A**: PRBS9 initial seed for trio1

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_INADJ_CALIB_CTRL2_0

Offset: 0x4750
Byte Offset: 0x11d40
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0fc00fc0 (0bxxxx,1111,1100,0000,xxxx,1111,1100,0000)

Bit	Reset	Description
27:22	0x3f	**INADJ_LIMIT_HIGH_TRIO1_A**: The upper limit for the INADJ sweep.
21:16	0x0	**INADJ_LIMIT_LOW_TRIO1_A**: The lower limit for the INADJ sweep.
11:6	0x3f	**INADJ_LIMIT_HIGH_TRIO0_A**: The upper limit for the INADJ sweep.
5:0	0x0	**INADJ_LIMIT_LOW_TRIO0_A**: The lower limit for the INADJ sweep.

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_INADJ_CALIB_STATUS_0

Offset: 0x4751
Byte Offset: 0x11d44
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**INADJ_CALIB_DONE_TRIO1_A**: Trio a1 inadj calibration done.
0	0x0	**INADJ_CALIB_DONE_TRIO0_A**: Trio a0 inadj calibration done.

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_INADJ_CALIB_STATUS1_0

Offset: 0x4752
Byte Offset: 0x11d48
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_LOW_TRIO0_A**: Trio a0 PRBS check status for lower 32 INADJ value.

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_INADJ_CALIB_STATUS2_0

Offset: 0x4753
Byte Offset: 0x11d4c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_LOW_TRIO1_A**: Trio a1 PRBS check status for lower 32 INADJ value.

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_INADJ_CALIB_STATUS3_0

Offset: 0x4754
Byte Offset: 0x11d50

Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_HIGH_TRIO0_A**: Trio a0 PRBS check status for upper 32 INADJ value.

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_INADJ_CALIB_STATUS4_0

Offset: 0x4755
Byte Offset: 0x11d54
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_HIGH_TRIO1_A**: Trio a1 PRBS check status for upper 32 INADJ value.

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_INADJ_OVERRIDE_0

Offset: 0x4756
Byte Offset: 0x11d58
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00080008 (0bxxxx,xxx0,0000,1000,xxxx,xxx0,0000,1000)

Bit	Reset	Description
24:17	0x4	**CPHY_INADJ_TRIO1_A**: The SW override INADJ value.
16	DISABLE	**SW_SET_CPHY_INADJ_TRIO1_A**: Enable SW override of the INADJ. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
8:1	0x4	**CPHY_INADJ_TRIO0_A**: The SW override INADJ value.
0	DISABLE	**SW_SET_CPHY_INADJ_TRIO0_A**: Enable SW override of the INADJ. **0** = DISABLE **1** = ENABLE

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_BIST_CTRL_0

Offset: 0x4757
Byte Offset: 0x11d5c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	0x0	**STOP_BIST_TRIO1_A**: Manually stop the A1 BIST, write 1 to stop the RX bist.
2	0x0	**STOP_BIST_TRIO0_A**: Manually stop the A0 BIST, write 1 to stop the RX bist.
1	0x0	**TRIGGER_BIST_TRIO1_A**: Trigger the trio A1 BIST, write 1 to trigger the RX bist, HW will clear it to 0 when the BIST is done, write 0 to this field is with no effect.
0	0x0	**TRIGGER_BIST_TRIO0_A**: Trigger the trio A0 BIST, write 1 to trigger the RX bist, HW will clear it to 0 when the BIST is done, write 0 to this field is with no effect.

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_BIST_STATUS_0_0

Offset: 0x4758
Byte Offset: 0x11d60
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:8	0x0	**WORD_ERR_CNT_TRIO1_A**: The trio A1 number of word comparison failure, report for BIST mode 1.
7:0	0x0	**WORD_ERR_CNT_TRIO0_A**: The trio A0 number of word comparison failure, report for BIST mode 1.

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_BIST_STATUS_1_0

Offset: 0x4759
Byte Offset: 0x11d64
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:8	0x0	**SYM_ERR_CNT_TRIO1_A**: The trio A1 number of symbol compariosn failure, it's the sum of the symbol err count and the clock shift count, the real wire state error number is 1/2 of this field, report for BIST mode 1.
7:0	0x0	**SYM_ERR_CNT_TRIO0_A**: The trio A0 number of symbol compariosn failure, it's the sum of the symbol err count and the clock shift count, the real wire state error number is 1/2 of this field, report for BIST mode 1.

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_BIST_STATUS_2_0

Offset: 0x475a
Byte Offset: 0x11d68
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WORD_CNT_LOW_TRIO0_A**: The trio A0 low 32 bits of the compared word number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_BIST_STATUS_3_0

Offset: 0x475b
Byte Offset: 0x11d6c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WORD_CNT_HIGH_TRIO0_A**: The trio A0 high 16 bits of the compared word number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_BIST_STATUS_4_0

Offset: 0x475c
Byte Offset: 0x11d70
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WORD_CNT_LOW_TRIO1_A**: The trio A1 low 32 bits of the compared word number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_BIST_STATUS_5_0

Offset: 0x475d
Byte Offset: 0x11d74
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WORD_CNT_HIGH_TRIO1_A**: The trio A1 high 16 bits of the compared word number number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_0_NVCSI_CIL_A_CPHY_BIST_STATUS_6_0

Offset: 0x475e
Byte Offset: 0x11d78
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,x000,0000,0000,xxxx,x000,0000,0000)

Bit	Reset	Description
26	0x0	**ERR_DETECT_FAIL_TRIO1_A**: Multi symbol error in one word (7 symbol) for trio A1, BIST logic is not able to detect the accurate wire state error. With this error, if not other errors detected, the word error is accurate.
25	0x0	**SHIFT_DETECT_FAIL_TRIO1_A**: Detect symbol error in two continuous word (14 symbols) for trio A1, BIST logic is not able to determine if symbol clock is slip. (if the symbol error count is not saturating, the result is still available with this error)
24	0x0	**SHIFT_OVERFLOW_TRIO1_A**: Too many clocks slip and the BIST logic is not able to re-align for trio A1. Typically, when this error is detect, the symbol error counter will saturate.
23:20	0x0	**DOUBLE_CLK_MISSED_CNT_TRIO1_A**: Number of the times for double symbol clock lost for trio A1.
19:16	0x0	**SINGLE_CLK_MISSED_CNT_TRIO1_A**: Number of the times for single symbol clock lost for trio A1.
10	0x0	**ERR_DETECT_FAIL_TRIO0_A**: Multi symbol error in one word (7 symbol) for trio A0, BIST logic is not able to detect the accurate wire state error. With this error, if not other errors detected, the word error is accurate.
9	0x0	**SHIFT_DETECT_FAIL_TRIO0_A**: Detect symbol error in two continuous word (14 symbols) for trio A0, BIST logic is not able to determine if symbol clock is slip. (if the symbol error count is not saturating, the result is still available with this error)
8	0x0	**SHIFT_OVERFLOW_TRIO0_A**: Too many clocks slip and the BIST logic is not able to re-align for trio A0. Typically, when this error is detect, the symbol error counter will saturate.
7:4	0x0	**DOUBLE_CLK_MISSED_CNT_TRIO0_A**: Number of the times for double symbol clock lost for trio A0.

Bit	Reset	Description
3:0	0x0	**SINGLE_CLK_MISSED_CNT_TRIO0_A**: Number of the times for single symbol clock lost for trio A0.

NVCSI_PHY_0_NVCSI_CIL_B_SW_RESET_0

Offset: 0x475f
Byte Offset: 0x11d7c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**SW_RESET1_B**: SOFT RESET FOR LANE B1. Active High **1** = ENABLE **0** = DISABLE
0	0x0	**SW_RESET0_B**: SOFT RESET FOR LANE B0. Active High **1** = ENABLE **0** = DISABLE

NVCSI_PHY_0_NVCSI_CIL_B_CLKEN_OVERRIDE_CTRL_0

Offset: 0x4760
Byte Offset: 0x11d80
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**CLKEN_OVERRIDE1_B**: CLKENABLE OVERRIDE FOR LANE B1. Active High **1** = ENABLE **0** = DISABLE

Bit	Reset	Description
0	0x0	**CLKEN_OVERRIDE0_B**: CLKENABLE OVERRIDE FOR LANE B0. Active High **1** = ENABLE **0** = DISABLE

NVCSI_PHY_0_NVCSI_CIL_B_CTLE_CTRL_0

Offset: 0x4761
Byte Offset: 0x11d84
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000888 (0bxxxx,xxxx,0000,0000,0000,1000,1000,1000)

Bit	Reset	Description
23:20	0x0	**AFE_DCGAIN_IO1_B**: RX AFE DC gain control
19:16	0x0	**AFE_DCGAIN_IO0_B**: RX AFE DC gain control
15:12	0x0	**AFE_DCGAIN_CLK_B**: RX AFE DC gain control
11:8	0x8	**AFE_HFGAIN_IO1_B**: RX AFE_CTLE high frequency gain control, 16 curves, ac_gain=code*1db
7:4	0x8	**AFE_HFGAIN_IO0_B**: RX AFE_CTLE high frequency gain control, 16 curves, ac_gain=code*1db
3:0	0x8	**AFE_HFGAIN_CLK_B**: RX AFE_CTLE high frequency gain control, 16 curves, ac_gain=code*1db

NVCSI_PHY_0_NVCSI_CIL_B_CTLE_CTRL1_0

Offset: 0x4762
Byte Offset: 0x11d88
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000554 (0bxxxx,xxxx,0000,0000,0000,0101,0101,0100)

Bit	Reset	Description
23:20	0x0	**AFE_CTRL_IO1_B**
19:16	0x0	**AFE_CTRL_IO0_B**
15:12	0x0	**AFE_CTRL_CLK_B**
11:10	0x1	**AFE_CURRENT_IO1_B**
9:8	0x1	**AFE_CURRENT_IO0_B**
7:6	0x1	**AFE_CURRENT_CLK_B**
5:4	0x1	**AFE_FREQBAND_IO1_B**: RX AFE bandwidth selection. 2'b00: <1.5GS/s, 2'b01: 1.5~3GS/s, 2'b10: 3~5GS/s, 2'b11: reserved.
3:2	0x1	**AFE_FREQBAND_IO0_B**: RX AFE bandwidth selection. 2'b00: <1.5GS/s, 2'b01: 1.5~3GS/s, 2'b10: 3~5GS/s, 2'b11: reserved.
1:0	0x0	**AFE_FREQBAND_CLK_B**: RX AFE bandwidth selection. 2'b00: <1.5GS/s, 2'b01: 1.5~3GS/s, 2'b10: 3~5GS/s, 2'b11: reserved.

NVCSI_PHY_0_NVCSI_CIL_B_PAD_CONFIG_0

Offset: 0x4763
Byte Offset: 0x11d8c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00070000 (0b0000,0000,0000,x111,xxxx,0000,0000,0000)

Bit	Reset	Description
31	0x0	**LP_RX_SELECT_IO1_B**: LP mode receiver select for partition B lane 1. DPHY mode: controls IO1P_B and IO1N_B CPHY mode: controls IO1P_B, IO1N_B and IOCLKN_B.
30	0x0	**LP_RX_SELECT_IO0_B**: LP mode receiver select for partition B lane 0. DPHY mode: controls IO0P_B and IO0N_B CPHY mode: controls IO0P_B, IO0N_B and IOCLKP_B.
29	0x0	**LP_RX_SELECT_CLK_B**: LP mode receiver select. 0: Schmitt receiver. 1: self-biased receiver. DPHY mode: controls IOCLKP_B and IOCLKN_B. CPHY mode: dummy.

Bit	Reset	Description
28	0x0	**E_LPRX_HYS_N_IO1_B**: Enable LP-RX hysteresis. 0: enable hysteresis 1: disable hysteresis
27	0x0	**E_LPRX_HYS_N_IO0_B**: Enable LP-RX hysteresis. 0: enable hysteresis 1: disable hysteresis
26	0x0	**E_LPRX_HYS_N_CLK_B**: Enable LP-RX hysteresis. 0: enable hysteresis 1: disable hysteresis
25	0x0	**E_HS_TERM_N_IO1_B**: Enable the HS termination of Partition B Lane 1.
24	0x0	**E_HS_TERM_N_IO0_B**: Enable the HS termination of Partition B Lane 0.
23	0x0	**E_HS_TERM_N_CLK_B**: Enable the HS termination of Clock partition B.
22	0x0	**E_INPUT_LP_IO1_B**: Enable LP receiver of Partition B Lane 1 Applicable in both CPHY and DPHY case.
21	0x0	**E_INPUT_LP_IO0_B**: Enable LP receiver of Partition B Lane 0 Applicable in both CPHY and DPHY case.
20	0x0	**E_INPUT_LP_CLK_B**: enable LP receiver of Clock partition B Applicable in DPHY case. N/A for CPHY
18	0x1	**PD_CLK_B**: Power down for CLk of Partition B. Applicable in DPHY case. N/A for CPHY
17	0x1	**PD_IO1_B**: Power down for Trio 1 and Lane 1 of Partition B. Applicable in both CPHY and DPHY case.
16	0x1	**PD_IO0_B**: Power down for Trio 0 and Lane 0 of Partition B. Applicable in both CPHY and DPHY case.
11:8	0x0	**SPARE_CLK_B**: Spare control bits for pad control. Functionality of other bits currently not defined. Undefined register bits are reserved for future use. Pull down may be enabled when Clock Lane is unused to minimized interference.
7:4	0x0	**SPARE_IO1_B**: Spare control bits for pad control. Functionality of other bits currently not defined. Undefined register bits are reserved for future use. Pull down may be enabled when Data Lane-1 is unused to minimized interference.

Bit	Reset	Description
3:0	0x0	**SPARE_IO0_B**: Spare control bits for pad control. Functionality of other bits currently not defined. Undefined register bits are reserved for future use. Pull down may be enabled when Data Lane-0 is unused to minimized interference.

NVCSI_PHY_0_NVCSI_CIL_B_PAD_CONFIG_1_0

Offset: 0x4764
Byte Offset: 0x11d90
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000002 (0b0000,0000,0000,0000,0000,0000,xx00,0010)

Bit	Reset	Description
31:24	0x0	**OUTADJ_IO1_B**: Trio 1 output delay trimmer, each tap delays 9ps
23:16	0x0	**OUTADJ_IO0_B**: Trio 0 output delay trimmer, each tap delays 9ps
15:8	0x0	**OUTADJ_CLK_B**: Clock bit output delay trimmer, each tap delays 9ps
5	0x0	**HS_BSO_B**: 1= power-on 0.4v-TX-regulator during LP-mode; 0= power-off 0.4v-TX-regulator during LP-mode;
4	0x0	**CPHY_MID_STRENGTH_TRIO1_B**: 0: when bump is driven to mid-voltage-level, both 100-ohm-pull-up and 100-ohm-pull-down is turned on to drive mid-voltage-level; 1: when bump is driven to mid-voltage-level, both pull-up and pull-down are turned off to become tri-stated;
3	0x0	**CPHY_MID_STRENGTH_TRIO0_B**: 0: when bump is driven to mid-voltage-level, both 100-ohm-pull-up and 100-ohm-pull-down is turned on to drive mid-voltage-level; 1: when bump is driven to mid-voltage-level, both pull-up and pull-down are turned off to become tri-stated;
2	DISABLE	**REV_CLK_B**: Reverse clock polarity **0** = DISABLE **1** = ENABLE
1	ENABLE	**E_PREDRV_FLOP_RESET_B**: HS predriver flop reset **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
0	DISABLE	**FCZERO_B**: Force clk bit to drive differential 0 **0** = DISABLE **1** = ENABLE

NVCSI_PHY_0_NVCSI_CIL_B_PAD_CONFIG_2_0

Offset: 0x4765
Byte Offset: 0x11d94
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,0000,0000,0000,xxxx,0000,0000,0000)

Bit	Reset	Description
27:24	0x0	**PEMPD_IO1_B**: Enable trio 1 HS driver pull down pre-emphasis.
23:20	0x0	**PEMPD_IO0_B**: Enable trio 0 HS driver pull down pre-emphasis.
19:16	0x0	**PEMPD_CLK_B**: Enable clock bit HS driver pull down pre-emphasis.
11:8	0x0	**PEMPU_IO1_B**: Enable trio 1 HS driver pull up pre-emphasis.
7:4	0x0	**PEMPU_IO0_B**: Enable trio 0 HS driver pull up pre-emphasis.
3:0	0x0	**PEMPU_CLK_B**: Enable clock bit HS driver pull up pre-emphasis.

NVCSI_PHY_0_NVCSI_CIL_B_PAD_CONFIG_3_0

Offset: 0x4766
Byte Offset: 0x11d98
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x06660666 (0bxxxx,0110,0110,0110,xxxx,0110,0110,0110)

Bit	Reset	Description
27:24	0x6	**LPDNADJ_IO1_B**: Driver pull down impedance control
23:20	0x6	**LPDNADJ_IO0_B**: Driver pull down impedance control
19:16	0x6	**LPDNADJ_CLK_B**: Driver pull down impedance control
11:8	0x6	**LPUPADJ_IO1_B**: Driver pull up impedance control
7:4	0x6	**LPUPADJ_IO0_B**: Driver pull up impedance control
3:0	0x6	**LPUPADJ_CLK_B**: Driver pull up impedance control

NVCSI_PHY_0_NVCSI_CIL_B_PAD_CONFIG_4_0

Offset: 0x4767
Byte Offset: 0x11d9c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,0000,0000,0000,xxxx,0000,0000,0000)

Bit	Reset	Description
27:24	0x0	**SLEWDNADJ_IO1_B**: Pull down slew rate adjust
23:20	0x0	**SLEWDNADJ_IO0_B**: Pull down slew rate adjust
19:16	0x0	**SLEWDNADJ_CLK_B**: Pull down slew rate adjust
11:8	0x0	**SLEWUPADJ_IO1_B**: Pull up slew rate adjust
7:4	0x0	**SLEWUPADJ_IO0_B**: Pull up slew rate adjust
3:0	0x0	**SLEWUPADJ_CLK_B**: Pull up slew rate adjust

NVCSI_PHY_0_NVCSI_CIL_B_PAD_CONFIG_5_0

Offset: 0x4768

Byte Offset: 0x11da0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000110 (0bxxxx,xxxx,xxxx,xxxx,xxxx,x001,0001,0000)

Bit	Reset	Description
10:7	0x2	**CDDNADJ_IO1_B**: Level adjust on low limit of detection
6:3	0x2	**CDDNADJ_IO0_B**: Level adjust on low limit of detection
2	DISABLE	**E_CD_IO1_B**: Data bits contention detector enable **0** = DISABLE **1** = ENABLE
1	DISABLE	**E_CD_IO0_B**: Data bits contention detector enable **0** = DISABLE **1** = ENABLE
0	DISABLE	**E_CD_CLK_B**: Clock bit contention detector enable **0** = DISABLE **1** = ENABLE

NVCSI_PHY_0_NVCSI_CIL_B_PAD_CONFIG_6_0

Offset: 0x4769
Byte Offset: 0x11da4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**E_DEEP_LPBK_B**: Enable deep loopback
0	0x0	**E_SHALLOW_LPBK_B**: Enalbe shallow loopback

NVCSI_PHY_0_NVCSI_CIL_B_PAD_CD_STATUS_0

Offset: 0x476a
Byte Offset: 0x11da8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**CDN_IO1_B**: trio1 n output, 1=contention detected.
4	0x0	**CDN_IO0_B**: trio0 n output, 1=contention detected.
3	0x0	**CDN_CLK_B**: Clock bit n output, 1=contention detected.
2	0x0	**CDP_IO1_B**: trio1 p output, 1=contention detected.
1	0x0	**CDP_IO0_B**: trio0 p output, 1=contention detected.
0	0x0	**CDP_CLK_B**: Clock bit p output, 1=contention detected.

NVCSI_PHY_0_NVCSI_CIL_B_DPHY_INADJ_CTRL_0

Offset: 0x476b
Byte Offset: 0x11dac
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,x000,0000,x000,0000,x000,0000)

Bit	Reset	Description
22	0x0	**SW_SET_DPHY_INADJ_CLK_B**: Software set for clock input delay trimmer
21:16	0x0	**DPHY_INADJ_CLK_B**: Programmable value for CLK input delay trimmer,

Bit	Reset	Description
14	0x0	**SW_SET_DPHY_INADJ_IO1_B**: Software override for bit 1 input delay trimmer 1 - SW programmed value is applied to the pads. 2 - Trimmer value from HW calibration state machine is applied to the pads
13:8	0x0	**DPHY_INADJ_IO1_B**: Programmable value for bit 1 input delay trimmer, each tap delay of 9 ps
6	0x0	**SW_SET_DPHY_INADJ_IO0_B**: Software override for bit 0 input delay trimmer 1 - SW programmed value is applied to the pads. 2 - Trimmer value from HW calibration state machine is applied to the pads
5:0	0x0	**DPHY_INADJ_IO0_B**: Programmable value for bit 0 input delay trimmer,

NVCSI_PHY_0_NVCSI_CIL_B_CLK_DESKEW_CTRL_0

Offset: 0x476c
Byte Offset: 0x11db0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000bf00 (0bxxxx,xxxx,xxxx,xxxx,1x11,1111,xx00,0000)

Bit	Reset	Description
15	ENABLE	**CLK_INADJ_SWEEP_CTRL_B**: Enable the clock trimmer sweep for D-PHY de-skew. **0** = DISABLE **1** = ENABLE
13:8	0x3f	**CLK_INADJ_LIMIT_HIGH_B**: The upper limit of the sweep range for trimmer sweep.
5:0	0x0	**CLK_INADJ_LIMIT_LOW_B**: The lower limit of the sweep range for trimmer sweep.

NVCSI_PHY_0_NVCSI_CIL_B_DATA_DESKEW_CTRL_0

Offset: 0x476d
Byte Offset: 0x11db4
Read/Write: R/W
Parity Protection: Y
Shadow: N

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x9f80bf00 (0b1xx1,1111,1x00,0000,1x11,1111,xx00,0000)

Bit	Reset	Description
31	0x1	**DATA_INADJ_SWEEP_CTRL1_B**: Enable the data lane B1 trimmer sweep for D-PHY de-skew.
28:23	0x3f	**DATA_INADJ_LIMIT_HIGH1_B**: The upper limit of the sweep range for trimmer sweep.
21:16	0x0	**DATA_INADJ_LIMIT_LOW1_B**: The lower limit of the sweep range for trimmer sweep.
15	0x1	**DATA_INADJ_SWEEP_CTRL0_B**: Enable the data lane B0 trimmer sweep for D-PHY de-skew.
13:8	0x3f	**DATA_INADJ_LIMIT_HIGH0_B**: The upper limit of the sweep range for trimmer sweep.
5:0	0x0	**DATA_INADJ_LIMIT_LOW0_B**: The lower limit of the sweep range for trimmer sweep.

NVCSI_PHY_0_NVCSI_CIL_B_DPHY_DESKEW_STATUS_0

Offset: 0x476e
Byte Offset: 0x11db8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000X0XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
15	X	**DPHY_CALIB_ERR_IO1_B**: calib error status
14	X	**DPHY_CALIB_DONE_IO1_B**: calib done status
7	X	**DPHY_CALIB_ERR_IO0_B**: calib error status
6	X	**DPHY_CALIB_DONE_IO0_B**: calib done status
1	X	**DPHY_CALIB_ERR_CTRL_B**: calib error status
0	X	**DPHY_CALIB_DONE_CTRL_B**: calib done status

NVCSI_PHY_0_NVCSI_CIL_B_DPHY_DESKEW_DATA_CALIB_STATUS_LOW_0_0

Offset: 0x476f
Byte Offset: 0x11dbc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_B**

NVCSI_PHY_0_NVCSI_CIL_B_DPHY_DESKEW_DATA_CALIB_STATUS_HIGH_0_0

Offset: 0x4770
Byte Offset: 0x11dc0
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_B**

NVCSI_PHY_0_NVCSI_CIL_B_DPHY_DESKEW_DATA_CALIB_STATUS_LOW_1_0

Offset: 0x4771
Byte Offset: 0x11dc4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_B**

NVCSI_PHY_0_NVCSI_CIL_B_DPHY_DESKEW_DATA_CALIB_STATUS_HIGH_1_0

Offset: 0x4772
Byte Offset: 0x11dc8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_B**

NVCSI_PHY_0_NVCSI_CIL_B_DPHY_DESKEW_CLK_CALIB_STATUS_LOW_0_0

Offset: 0x4773
Byte Offset: 0x11dcc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_B**

NVCSI_PHY_0_NVCSI_CIL_B_DPHY_DESKEW_CLK_CALIB_STATUS_HIGH_0_0

Offset: 0x4774
Byte Offset: 0x11dd0
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_B**

NVCSI_PHY_0_NVCSI_CIL_B_DPHY_DESKEW_CLK_CALIB_STATUS_LOW_1_0

Offset: 0x4775
Byte Offset: 0x11dd4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_B**

NVCSI_PHY_0_NVCSI_CIL_B_DPHY_DESKEW_CLK_CALIB_STATUS_HIGH_1_0

Offset: 0x4776
Byte Offset: 0x11dd8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_B**

NVCSI_PHY_0_NVCSI_CIL_B_DPHY_DESKEW_RESULT_STATUS_0

Offset: 0x4777
Byte Offset: 0x11ddc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	DONE	**DESKEW_RESULT_STATUS1_B**: For lane B1. The result status of the deskew calibration, HW check this field before start the trimmer sweep, if it's 1'b1, the trimmer sweep will not perform. HW set this bit to 1 when the calibration is done. SW should clear this field after it read out the 2 64 bits calibration status. **0** = DONE **1** = PENDING
0	DONE	**DESKEW_RESULT_STATUS0_B**: For lane B0. The result status of the deskew calibration, HW check this field before start the trimmer sweep, if it's 1'b1, the trimmer sweep will not perform. HW set this bit to 1 when the calibration is done. SW should clear this field after it read out the 2 64 bits calibration status. **0** = DONE **1** = PENDING

NVCSI_PHY_0_NVCSI_CIL_B_POLARITY_SWIZZLE_CTRL_0

Offset: 0x4778
Byte Offset: 0x11de0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xx00,0000,xxxx,x000)

Bit	Reset	Description
13:11	0x0	**POLARITY_SWIZZLE_CPHY1_B**: Polarity Swizle control for Lane B1 in CPHY mode. valid only in CPHY mode 000 - A B C --> A B C 001 - A B C --> A C B 010 - A B C --> B C A 011 - A B C --> B A C 100 - A B C --> C A B 101 - A B C --> C B A
10:8	0x0	**POLARITY_SWIZZLE_CPHY0_B**: Polarity Swizle control for Lane B0 in CPHY mode. valid only in CPHY mode 000 - A B C --> A B C 001 - A B C --> A C B 010 - A B C --> B C A 011 - A B C --> B A C 100 - A B C --> C A B 101 - A B C --> C B A
2	0x0	**POLARITY_SWIZZLE_CLK_B**: Polarity Swizzle control for clock lane B. Valid only in DPHY mode, need to work with ALLOW_FIRST_BIT_ON_CLK_NEGEDGE if enabled.

Bit	Reset	Description
1	0x0	**POLARITY_SWIZZLE_DPHY1_B**: Polarity Swizzle control for data lane B1. register bit valid only in DPHY mode.
0	0x0	**POLARITY_SWIZZLE_DPHY0_B**: Polarity Swizzle control for data lane B0. register bit valid only in DPHY mode.

NVCSI_PHY_0_NVCSI_CIL_B_DESKEW_CONTROL_0

Offset: 0x4779
Byte Offset: 0x11de4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00007411 (0bxxxx,xxxx,xxxx,xxxx,0111,0100,0001,0001)

Bit	Reset	Description
15:13	0x3	**DESKEW_DATA_DELAY**: How many cycles delay introduced by re-timing flops
12	0x1	**DESKEW_LANE_SKEW_CHECK_EN**: Check the skew between the lanes. this is used to handle the error case that error happen in the CSI link caused some data lanes detect the HS sync word (0xb8) while other data lanes detect the deskew sync word (0xff). If this bit is not set with the error case, the lane FIFO will overflow and reset is required. With this bit set, the FIFO will be auto flused and the FIFO overflow will not happen.
11:8	0x4	**DESKEW_LANE_SKEW_TOLERANCE**: The allowed skew between the data lane, only used in sync word search
7:4	0x1	**DESKEW_COMPARE**: Regsiter select to control the number of comparisons to be done for each trimmer setting during deskew calibration.
3:0	0x1	**DESKEW_SETTLE**: Regsiter select to control the number of byte clocks to wait before INADJ value settles. This is used during deskew calibration

NVCSI_PHY_0_NVCSI_CIL_B_CONTROL_0

Offset: 0x477a
Byte Offset: 0x11de8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**BYPASS_LP_FILTER**: Bypass the glitch filter.
30:24	0x0	**CLK_SETTLE1_B**: Used only in CPHY mode, Settle time for clk start when moving B1 data lane from LP to HS (LP11->LP01->LP00).
23:17	0x0	**CLK_SETTLE0_B**: Used for both CPHY/DPY mode. For DPHY: Settle time for clk lane when moving from LP to HS (LP11->LP01->LP00), this setting determines how many csicil clock cycles to wait after LP00. If the settle set to 0 or 8, the real setting is 1, otherwise, the real setting is (4+settle) For CPHY: Settle time for clk start when moving B0 data lane from LP to HS (LP11->LP01->LP00).
16:9	0x0	**THS_SETTLE1_B**: settle time for B1 data lane when moving from LP to HS (LP11->LP01->LP00), this setting determines how many LP clock cycles to wait, after LP00, before starting to look at the data.
8:1	0x0	**THS_SETTLE0_B**: settle time for B0 data lane when moving from LP to HS (LP11->LP01->LP00), this setting determines how many LP clock cycles to wait, after LP00, before starting to look at the data.
0	0x0	**BYPASS_LP_SEQ**: For DPHY: The clock lane LP signals should sequence through LP11->LP01->LP00 state, to indicate to CLOCK CIL about the mode switching to HS Rx mode. In case Camera is enabled earlier than CIL , it is highly likely that camera sends this control sequence sooner than cil can detect it. Enabling this bit allows the CLOCK CIL to overlook the LP control sequence and step in HS Rx mode directly looking at LP00 only. For CPHY: it's only used for debug purpose as it's embedded clock for CPHY, clock will always presents when the data lane toggle.

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_ERR_STATUS_0

Offset: 0x477b
Byte Offset: 0x11dec
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
29:24	X	**LANE_DEMAPPER_ERR_MUX1_B**: Debug register, error detected in CPHY de-mapper mux.
23	X	**LANE_DEMAPPER_ERR_RXCTRL1_B**: Debug register, error detected in CPHY de-mapper rx ctrl.
22:16	X	**LANE_DECODER_ERR1_B**: Debug register, error detected in CPHY de-coder.
13:8	X	**LANE_DEMAPPER_ERR_MUX0_B**: Debug register, error detected in CPHY de-mapper mux.
7	X	**LANE_DEMAPPER_ERR_RXCTRL0_B**: Debug register, error detected in CPHY de-mapper rx ctrl.
6:0	X	**LANE_DECODER_ERR0_B**: Debug register, error detected in CPHY de-coder.

NVCSI_PHY_0_NVCSI_CIL_B_ESCAPE_MODE_COMMAND_0_0

Offset: 0x477c
Byte Offset: 0x11df0
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_COMMAND0_B**: The received command from escape mode.

NVCSI_PHY_0_NVCSI_CIL_B_ESCAPE_MODE_DATA_0_0

Offset: 0x477d
Byte Offset: 0x11df4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_DATA0_B**: The received data from excape mode.

NVCSI_PHY_0_NVCSI_CIL_B_ESCAPE_MODE_COMMAND_1_0

Offset: 0x477e
Byte Offset: 0x11df8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_COMMAND1_B**: The received command from escape mode.

NVCSI_PHY_0_NVCSI_CIL_B_ESCAPE_MODE_DATA_1_0

Offset: 0x477f
Byte Offset: 0x11dfc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_DATA1_B**: The received data from excape mode.

NVCSI_PHY_0_NVCSI_CIL_B_DPHY_SYNC_PATTERN_0

Offset: 0x4780
Byte Offset: 0x11e00
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01)

Bit	Reset	Description
1	0x0	**ALLOW_FIRST_BIT_ON_CLK_NEGEDGE_B**: Allow shift 1/3/5/7 bit to search the sync word. Normal case only shift 0/2/4/6 bit to search sync word.
0	0x1	**DISABLE_SB_ERR_IN_SYNC_B**: Allow one single bit error in sync word 6MSB.

NVCSI_PHY_0_NVCSI_CIL_B_DPHY_DESKEW_SYNC_PATTERN_0

Offset: 0x4781
Byte Offset: 0x11e04
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**INVERT_DESKEW_PATTERN**: Invert the de-skew pattern.

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_SYNC_SEARCH_0

Offset: 0x4782
Byte Offset: 0x11e08
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01080108 (0bxxxx,xxx1,0000,1000,xxxx,xxx1,0000,1000)

Bit	Reset	Description
24	0x1	**COUNT_EN_TRIO1_B**: Check if the symbol is pre-amble before sync word.
23:16	0x8	**PERIOD_TRIO1_B**: For CPHY, how many cycles allowed for non pre-amble symbol before sync word.
8	0x1	**COUNT_EN_TRIO0_B**: Check if the symbol is pre-amble before sync word.

Bit	Reset	Description
7:0	0x8	**PERIOD_TRIO0_B**: For CPHY, how many cycles allowed for non pre-amble symbol before sync word.

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_START_0

Offset: 0x4783
Byte Offset: 0x11e0c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**EDGE_DELAY_CAL_TRIGGER_TRIO1_B**: Trigger the CPHY clock recover calibration for lane B1 by write 1'b1, when the calibration complete, HW will clear this bit to 1'b0. Write 1'b0 to this be take no effect.
0	0x0	**EDGE_DELAY_CAL_TRIGGER_TRIO0_B**: Trigger the CPHY clock recover calibration for lane B0 by write 1'b1, when the calibration complete, HW will clear this bit to 1'b0. Write 1'b0 to this be take no effect.

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_CTRL_0

Offset: 0x4784
Byte Offset: 0x11e10
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x12011201 (0b0001,0010,0000,0001,0001,0010,0000,0001)
PROD: 0x08000800 (0bxxx0,10xx,xxxx,xxxx,xxx0,10xx,xxxx,xxxx)

Bit	Reset	PROD	Description
31	0x0	_NONE_	**TRIM_CAL_E_NONSTOP_TRIO1_B**: Reserved. Trimmer calibration background monitoring control. 1: enable. The accumulator keeps updating.

Bit	Reset	PROD	Description
30:29	0x0	_NONE_	**TRIM_CAL_GAIN_SEL_TRIO1_B**: Trimmer calibration loop gain selection [0]: gain control 0: coarse tune (counter1) gain = 1, fine tune (counter2) gain = 1/4. For example: coarse tune: 1 UP -> trim_cal_code + 1; fine tune: 4 UPs -> trim_cal_code + 1. 1: coarse tune (counter1) gain = 1/2, fine tune (counter2) gain = 1/8 [1]: reserved
28:26	0x4	0x2	**TRIM_CAL_CNT_SEL_TRIO1_B**: Trimmer calibration counter selection [0]: counter1 (coarse tune) clock cycle. 0: 64T (128UI) 1: 128T (256UI) [2:1]: counter2 (fine tune) clock cycle. 00: 32T (64UI) 01: 64T (128UI) 10: 96T (192UI) 11: 128T (256UI) For high data rate (>3.5GS/s): 64T(cnt1)+96T(cnt2)=160T(320ui) -> cnt_sel=3'b100 For low data rate (<=3.5GS/s): 128T(cnt1)+32T(cnt2)=160T(320ui) -> cnt_se=3'b001
25	0x1	_NONE_	**TRIM_CAL_CPHY_ED_LOOPMODE_TRIO1_B**: Trimmer calibration and cphy edge detector loop mode control 0: exclude trimmer delay in 2nd round of reset loop. 1: include trimmer delay in 2nd round of reset loop. For low data rate.
24:16	0x1	_NONE_	**CALIB_LENGTH_TRIO1_B**: The length of the peramble which use to do caliration. Count by nvcsi core clock. It's used to make sure the calibration is doing inside the preamble phase. If this length set to a small value, the calibration may can't complete within one packet and need several packets to complete.
15	0x0	_NONE_	**TRIM_CAL_E_NONSTOP_TRIO0_B**: Reserved. Trimmer calibration background monitoring control. 1: enable. The accumulator keeps updating.
14:13	0x0	_NONE_	**TRIM_CAL_GAIN_SEL_TRIO0_B**: Trimmer calibration loop gain selection [0]: gain control 0: coarse tune (counter1) gain = 1, fine tune (counter2) gain = 1/4. For example: coarse tune: 1 UP -> trim_cal_code + 1; fine tune: 4 UPs -> trim_cal_code + 1. 1: coarse tune (counter1) gain = 1/2, fine tune (counter2) gain = 1/8 [1]: reserved

Bit	Reset	PROD	Description
12:10	0x4	0x2	**TRIM_CAL_CNT_SEL_TRIO0_B**: Trimmer calibration counter selection [0]: counter1 (coarse tune) clock cycle. 0: 64T (128UI) 1: 128T (256UI) [2:1]: counter2 (fine tune) clock cycle. 00: 32T (64UI) 01: 64T (128UI) 10: 96T (192UI) 11: 128T (256UI) For high data rate (>3.5GS/s): 64T(cnt1)+96T(cnt2)=160T(320ui) -> cnt_sel=3'b100 For low data rate (<=3.5GS/s): 128T(cnt1)+32T(cnt2)=160T(320ui) -> cnt_se=3'b001
9	0x1	_NONE_	**TRIM_CAL_CPHY_ED_LOOPMODE_TRIO0_B**: Trimmer calibration and cphy edge detector loop mode control 0: exclude trimmer delay in 2nd round of reset loop. 1: include trimmer delay in 2nd round of reset loop. For low data rate.
8:0	0x1	_NONE_	**CALIB_LENGTH_TRIO0_B**: The length of the peramble which use to do caliration. Count by nvcsi core clock. It's used to make sure the calibration is doing inside the preamble phase. If this length set to a small value, the calibration may can't complete within one packet and need several packets to complete.

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_CTRL1_0

Offset: 0x4785
Byte Offset: 0x11e14
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0c0e0c0e (0bx000,1100,0000,1110,x000,1100,0000,1110)

Bit	Reset	Description
30	0x0	**PERIODIC_CALIB_TRIO1_B**: Enable the periodic calibration.
29	0x0	**CALIB_APPLY_IN_POST_TRIO1_B**: 1: Apply the calibration result when detect post symbol. 0: Apply the calibration result immediately when the calibration done.

Bit	Reset	Description
28:27	0x1	**CALIB_APPLY_SELECT_TRIO1_B**: Which calibration result is applied to pad for dual calibration. 2'b11: Reserved 2'b10: Apply the middle value of two calibration to pad if the difference within threshold. 2'b01: Apply the 2nd calibration result to pad if it's difference with previous latch result within threshold. 2'b00: Apply the 1st calibration result to pad if it's difference with previous latch result within threshold.
26	0x1	**DUAL_CALIB_TRIO1_B**: Two times calibratiton. 1: Two time calibration with one trigger. 0: Single time calibration with one trigger
25:18	0x3	**TEMP_EDGE_DELAY_VALUE_TRIO1_B**: The edge delay value while doing the calibration.
17	0x1	**TEMP_EDGE_DELAY_SEL_TRIO1_B**: When the selection of edge delay value when do edge delay calibration. 1: Use the temp value from register setting. 0: Use the currently using setting. For the first calibration after reset, it's still using the temp value from register setting.
16	0x0	**BYPASS_CALIB_PACKET_TRIO1_B**: 1: Drop the packet data when do the calibration. 0: Not drop the packet data when do the calibration.
14	0x0	**PERIODIC_CALIB_TRIO0_B**: Enable the periodic calibration.
13	0x0	**CALIB_APPLY_IN_POST_TRIO0_B**: 1: Apply the calibration result when detect post symbol. 0: Apply the calibration result immediately when the calibration done.
12:11	0x1	**CALIB_APPLY_SELECT_TRIO0_B**: Which calibration result is applied to pad for dual calibration. 2'b11: Reserved 2'b10: Apply the middle value of two calibration to pad if the difference within threshold. 2'b01: Apply the 2nd calibration result to pad if it's difference with previous latch result within threshold. 2'b00: Apply the 1st calibration result to pad if it's difference with previous latch result within threshold.
10	0x1	**DUAL_CALIB_TRIO0_B**: Two times calibratiton. 1: Two time calibration with one trigger. 0: Single time calibration with one trigger
9:2	0x3	**TEMP_EDGE_DELAY_VALUE_TRIO0_B**: The edge delay value apply to pad while doing the calibration.
1	0x1	**TEMP_EDGE_DELAY_SEL_TRIO0_B**: When the selection of edge delay value when do edge delay calibration. 1: Use the temp value from register setting. 0: Use the currently using setting. For the first calibration after reset, it's still using the temp value from register setting.
0	0x0	**BYPASS_CALIB_PACKET_TRIO0_B**: 1: Drop the packet data when do the calibration. 0: Not drop the packet data when do the calibration.

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_CTRL2_0

Offset: 0x4786
Byte Offset: 0x11e18
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01000100 (0bxxxx,xxx1,0000,0000,xxxx,xxx1,0000,0000)

Bit	Reset	Description
24	ENABLE	**TRIM_CAL_PD_HOLD_TRIO1_B**: Hold in PD **0** = DISABLE **1** = ENABLE
23	DISABLE	**TRIM_CAL_E_FIXEDDLY_TRIO1_B**: Enable fixed delay in trim calibration unit. **0** = DISABLE **1** = ENABLE
22:16	0x0	**TRIM_CAL_FIXEDDLY_TRIO1_B**: Adjust fixed delay. Delay = code*5ps(TT)/7ps(SS)
8	ENABLE	**TRIM_CAL_PD_HOLD_TRIO0_B**: Hold in PD **0** = DISABLE **1** = ENABLE
7	DISABLE	**TRIM_CAL_E_FIXEDDLY_TRIO0_B**: Enable fixed delay in trim calibration unit. **0** = DISABLE **1** = ENABLE
6:0	0x0	**TRIM_CAL_FIXEDDLY_TRIO0_B**: Adjust fixed delay. Delay = code*5ps(TT)/7ps(SS)

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_CTRL3_0

Offset: 0x4787
Byte Offset: 0x11e1c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffffffff	**CALIB_PERIOD_TRIO0_B**: The period of the periodic edge delay calibration, in nvcsi clock cycle number.

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_CTRL4_0

Offset: 0x4788
Byte Offset: 0x11e20
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffffffff	**CALIB_PERIOD_TRIO1_B**: The period of the periodic edge delay calibration, in nvcsi clock cycle number.

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_CTRL5_0

Offset: 0x4789
Byte Offset: 0x11e24
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000ffff (0bxxxx,xxxx,xxxx,xxxx,1111,1111,1111,1111)

Bit	Reset	Description
15:8	0xff	**ERR_THRESHOLD_TRIO1_B**: The threshold of drop the calibration result between current result and previous result. For dual calibration, check between the two calibration result in the same trigger. For signal calibration, check between the current calibration result and previous calibration result.
7:0	0xff	**ERR_THRESHOLD_TRIO0_B**: The threshold of drop the calibration result between current result and previous result. For dual calibration, check between the two calibration result in the same trigger. For signal calibration, check between the current calibration result and previous calibration result.

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_STATUS_0

Offset: 0x478a
Byte Offset: 0x11e28
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,xxxx)

Bit	Reset	Description
9	0x0	**EDGE_DELAY_MISMATCH_TRIO1_B**: When the CALIB_APPLY_SELECT_TRIO is 0/1 or dual time calibration is not enabled, the calibration result difference from the previous latched calibration result is larger than threshold, this time calibration result is not apply to pad. When the CALIB_APPLY_SELECT_TRIO is 2, the calibration result difference of the two times calibration result is larger than threshold, this time calibration result is not apply to pad.
8	0x0	**EDGE_DELAY_MISMATCH_TRIO0_B**: When the CALIB_APPLY_SELECT_TRIO is 0/1 or dual time calibration is not enabled, the calibration result difference from the previous latched calibration result is larger than threshold, this time calibration result is not apply to pad. When the CALIB_APPLY_SELECT_TRIO is 2, the calibration result difference of the two times calibration result is larger than threshold, this time calibration result is not apply to pad.
7	0x0	**CALIB_DONE_1_TRIO1_B**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.
6	0x0	**CALIB_DONE_1_TRIO0_B**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.
5	0x0	**CALIB_DONE_0_TRIO1_B**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.
4	0x0	**CALIB_DONE_0_TRIO0_B**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_EDGE_DELAY_CTRL_0

Offset: 0x478b
Byte Offset: 0x11e2c
Read/Write: R/W
Parity Protection: See table below
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x40000000 (0b01xx,x000,0000,0000,xxxx,x000,0000,0000)

Bit	Parity Protection	Reset	Description
31:30	Y	0x1	**CPHY_CLKGEN_TRIGWIDTH_B**: Adjust AB/CB/CA pulse width in self-clock generator
26	N	0x0	**UPDATE_OFFSET_TRIO1_B**: Trigger to apply the a offset value. Used to update offset directly without trigger a new calibration.
25:17	Y	0x0	**DELAY_OFFSET_TRIO1_B**: Offset value of trio B1. Represents signed 2's complement number for calibration offset. Represents unsigned number if the field used as override value.
16	Y	0x0	**OVERRIDE_CAL_VAL_TRIO1_B**: Override for trio B1 0: Apply the DELAY_OFFSET as an offset to the calibration result. 1: Apply the DELAY_OFFSET value to the pad.
10	N	0x0	**UPDATE_OFFSET_TRIO0_B**: Trigger to apply the a offset value. Used to update offset directly without trigger a new calibration.
9:1	Y	0x0	**DELAY_OFFSET_TRIO0_B**: Offset value of trio B0. Represents signed 2's complement number for calibration offset. Represents unsigned number if the field used as override value.
0	Y	0x0	**OVERRIDE_CAL_VAL_TRIO0_B**: Override for trio B0 0: Apply the DELAY_OFFSET as an offset to the calibration result. 1: Apply the DELAY_OFFSET value to the pad.

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_EDGE_DELAY_VALUE_0

Offset: 0x478c
Byte Offset: 0x11e30
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000303 (0bxxxx,xxxx,xxxx,xxxx,0000,0011,0000,0011)

Bit	Reset	Description
15:8	0x3	**EDGE_DELAY_VALUE_TRIO1_B**: The edge delay value apply to pad finally

Bit	Reset	Description
7:0	0x3	**EDGE_DELAY_VALUE_TRIO0_B**: The edge delay value apply to pad finally

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_RESULT_0

Offset: 0x478d
Byte Offset: 0x11e34
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**EDGE_DELAY_CALIB_RESULT_1_TRIO1_B**: The edge delay calibrate result, with offset.
23:16	0x0	**EDGE_DELAY_CALIB_RESULT_1_TRIO0_B**: The edge delay calibrate result, with offset.
15:8	0x0	**EDGE_DELAY_CALIB_RESULT_0_TRIO1_B**: The edge delay calibrate result, with offset.
7:0	0x0	**EDGE_DELAY_CALIB_RESULT_0_TRIO0_B**: The edge delay calibrate result, with offset.

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_INADJ_CALIB_CTRL_0

Offset: 0x478e
Byte Offset: 0x11e38
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
16	0x0	**INADJ_CAL_TRIGGER_TRIO1_B**: Trigger INADJ calibration for trio0
0	0x0	**INADJ_CAL_TRIGGER_TRIO0_B**: Trigger INADJ calibration for trio1

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_INADJ_CALIB_CTRL1_0

Offset: 0x478f
Byte Offset: 0x11e3c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00010001 (0b0000,0000,0000,0001,0000,0000,0000,0001)

Bit	Reset	Description
31:16	0x1	**PRBS9_SEED_TRIO1_B**: PRBS9 initial seed for trio0
15:0	0x1	**PRBS9_SEED_TRIO0_B**: PRBS9 initial seed for trio1

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_INADJ_CALIB_CTRL2_0

Offset: 0x4790
Byte Offset: 0x11e40
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0fc00fc0 (0bxxxx,1111,1100,0000,xxxx,1111,1100,0000)

Bit	Reset	Description
27:22	0x3f	**INADJ_LIMIT_HIGH_TRIO1_B**: The upper limit for the INADJ sweep.
21:16	0x0	**INADJ_LIMIT_LOW_TRIO1_B**: The lower limit for the INADJ sweep.
11:6	0x3f	**INADJ_LIMIT_HIGH_TRIO0_B**: The upper limit for the INADJ sweep.
5:0	0x0	**INADJ_LIMIT_LOW_TRIO0_B**: The lower limit for the INADJ sweep.

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_INADJ_CALIB_STATUS_0

Offset: 0x4791
Byte Offset: 0x11e44

Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**INADJ_CALIB_DONE_TRIO1_B**: Trio b1 inadj calibration done.
0	0x0	**INADJ_CALIB_DONE_TRIO0_B**: Trio b0 inadj calibration done.

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_INADJ_CALIB_STATUS1_0

Offset: 0x4792
Byte Offset: 0x11e48
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_LOW_TRIO0_B**: Trio b0 PRBS check status for lower 32 INADJ value.

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_INADJ_CALIB_STATUS2_0

Offset: 0x4793
Byte Offset: 0x11e4c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_LOW_TRIO1_B**: Trio b1 PRBS check status for lower 32 INADJ value.

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_INADJ_CALIB_STATUS3_0

Offset: 0x4794
Byte Offset: 0x11e50
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_HIGH_TRIO0_B**: Trio b0 PRBS check status for upper 32 INADJ value.

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_INADJ_CALIB_STATUS4_0

Offset: 0x4795
Byte Offset: 0x11e54
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_HIGH_TRIO1_B**: Trio b1 PRBS check status for upper 32 INADJ value.

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_INADJ_OVERRIDE_0

Offset: 0x4796
Byte Offset: 0x11e58
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00080008 (0bxxxx,xxx0,0000,1000,xxxx,xxx0,0000,1000)

Bit	Reset	Description
24:17	0x4	**CPHY_INADJ_TRIO1_B**: The SW override INADJ value.
16	DISABLE	**SW_SET_CPHY_INADJ_TRIO1_B**: Enable SW override of the INADJ. **0** = DISABLE **1** = ENABLE
8:1	0x4	**CPHY_INADJ_TRIO0_B**: The SW override INADJ value.
0	DISABLE	**SW_SET_CPHY_INADJ_TRIO0_B**: Enable SW override of the INADJ. **0** = DISABLE **1** = ENABLE

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_BIST_CTRL_0

Offset: 0x4797
Byte Offset: 0x11e5c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	0x0	**STOP_BIST_TRIO1_B**: Manually stop the B1 BIST, write 1 to stop the RX bist.
2	0x0	**STOP_BIST_TRIO0_B**: Manually stop the B0 BIST, write 1 to stop the RX bist.
1	0x0	**TRIGGER_BIST_TRIO1_B**: Trigger the trio B1 BIST, write 1 to trigger the RX bist, HW will clear it to 0 when the BIST is done, write 0 to this field is with no effect.
0	0x0	**TRIGGER_BIST_TRIO0_B**: Trigger the trio B0 BIST, write 1 to trigger the RX bist, HW will clear it to 0 when the BIST is done, write 0 to this field is with no effect.

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_BIST_STATUS_0_0

Offset: 0x4798
Byte Offset: 0x11e60
Read/Write: RO
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:8	0x0	**WORD_ERR_CNT_TRIO1_B**: The trio B1 number of word comparison failure, report for BIST mode 1.
7:0	0x0	**WORD_ERR_CNT_TRIO0_B**: The trio B0 number of word comparison failure, report for BIST mode 1.

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_BIST_STATUS_1_0

Offset: 0x4799
Byte Offset: 0x11e64
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:8	0x0	**SYM_ERR_CNT_TRIO1_B**: The trio B1 number of symbol compariosn failure, it's the sum of the symbol err count and the clock shift count, the real wire state error number is 1/2 of this field, report for BIST mode 1.
7:0	0x0	**SYM_ERR_CNT_TRIO0_B**: The trio B0 number of symbol compariosn failure, it's the sum of the symbol err count and the clock shift count, the real wire state error number is 1/2 of this field, report for BIST mode 1.

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_BIST_STATUS_2_0

Offset: 0x479a
Byte Offset: 0x11e68
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WORD_CNT_LOW_TRIO0_B**: The trio B0 low 32 bits of the compared word number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_BIST_STATUS_3_0

Offset: 0x479b
Byte Offset: 0x11e6c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WORD_CNT_HIGH_TRIO0_B**: The trio B0 high 16 bits of the compared word number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_BIST_STATUS_4_0

Offset: 0x479c
Byte Offset: 0x11e70
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WORD_CNT_LOW_TRIO1_B**: The trio B1 low 32 bits of the compared word number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_BIST_STATUS_5_0

Offset: 0x479d
Byte Offset: 0x11e74
Read/Write: RO
Parity Protection: N
Shadow: N

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WORD_CNT_HIGH_TRIO1_B**: The trio B1 high 16 bits of the compared word number number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_0_NVCSI_CIL_B_CPHY_BIST_STATUS_6_0

Offset: 0x479e
Byte Offset: 0x11e78
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,x000,0000,0000,xxxx,x000,0000,0000)

Bit	Reset	Description
26	0x0	**ERR_DETECT_FAIL_TRIO1_B**: Multi symbol error in one word (7 symbol) for trio B1, BIST logic is not able to detect the accurate wire state error. With this error, if not other errors detected, the word error is accurate.
25	0x0	**SHIFT_DETECT_FAIL_TRIO1_B**: Detect symbol error in two continuous word (14 symbols) for trio B1, BIST logic is not able to determine if symbol clock is slip. (if the symbol error count is not saturating, the result is still available with this error)
24	0x0	**SHIFT_OVERFLOW_TRIO1_B**: Too many clocks slip and the BIST logic is not able to re-align for trio B1. Typically, when this error is detect, the symbol error counter will saturate.
23:20	0x0	**DOUBLE_CLK_MISSED_CNT_TRIO1_B**: Number of the times for double symbol clock lost for trio B1.
19:16	0x0	**SINGLE_CLK_MISSED_CNT_TRIO1_B**: Number of the times for single symbol clock lost for trio B1.
10	0x0	**ERR_DETECT_FAIL_TRIO0_B**: Multi symbol error in one word (7 symbol) for trio B0, BIST logic is not able to detect the accurate wire state error. With this error, if not other errors detected, the word error is accurate.
9	0x0	**SHIFT_DETECT_FAIL_TRIO0_B**: Detect symbol error in two continuous word (14 symbols) for trio B0, BIST logic is not able to determine if symbol clock is slip. (if the symbol error count is not saturating, the result is still available with this error)
8	0x0	**SHIFT_OVERFLOW_TRIO0_B**: Too many clocks slip and the BIST logic is not able to re-align for trio B0. Typically, when this error is detect, the symbol error counter will saturate.

Bit	Reset	Description
7:4	0x0	**DOUBLE_CLK_MISSED_CNT_TRIO0_B**: Number of the times for double symbol clock lost for trio B0.
3:0	0x0	**SINGLE_CLK_MISSED_CNT_TRIO0_B**: Number of the times for single symbol clock lost for trio B0.

```
base NVCSI_PHY_1_CILA 0x021400;
```

NVCSI_PHY_1_CILA_INTR_0_STATUS_CILA_0

Offset: 0x8500
Byte Offset: 0x21400
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,00x0,0xxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
30	0x0	**intr_cphy_edge_delay_cal_done_trio1_a**
29	0x0	**intr_cphy_edge_delay_cal_done_trio0_a**
28	0x0	**intr_dphy_cil_lane_align_err_a**
27	0x0	**intr_dphy_cil_deskew_calib_err_ctrl_a**
26	0x0	**intr_dphy_cil_deskew_calib_err_lane1_a**
25	0x0	**intr_dphy_cil_deskew_calib_err_lane0_a**
24	0x0	**intr_dphy_cil_deskew_calib_done_ctrl_a**
23	0x0	**intr_dphy_cil_deskew_calib_done_lane1_a**
22	0x0	**intr_dphy_cil_deskew_calib_done_lane0_a**
20	0x0	**intr_cil_data_lane_esc_mode_sync_err1_a**
19	0x0	**intr_cil_data_lane_esc_mode_sync_err0_a**
8	0x0	**intr_cil_data_lane_rxfifo_full_err1_a**
7	0x0	**intr_cil_data_lane_ctrl_err1_a**
6	0x0	**intr_cil_data_lane_sot_mb_err1_a**
5	0x0	**intr_cil_data_lane_sot_sb_err1_a**
4	0x0	**intr_cil_data_lane_rxfifo_full_err0_a**

Bit	Reset	Description
3	0x0	**intr_cil_data_lane_ctrl_err0_a**
2	0x0	**intr_cil_data_lane_sot_mb_err0_a**
1	0x0	**intr_cil_data_lane_sot_sb_err0_a**
0	0x0	**intr_dphy_cil_clk_lane_ctrl_err_a**

NVCSI_PHY_1_CILA_INTR_1_STATUS_CILA_0

Offset: 0x8501
Byte Offset: 0x21404
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**intr_cphy_inadj_cal_done_trio1_a**
13	0x0	**intr_cphy_inadj_cal_done_trio0_a**
12	0x0	**intr_idac_cal_done_trio1_a**
11	0x0	**intr_idac_cal_done_trio0_a**
10	0x0	**intr_dphy_cil_clk_lane_ulpm_req_a**
9	0x0	**intr_cil_lpdt_int1_a**
8	0x0	**intr_cil_ulps_trigger_int1_a**
7	0x0	**intr_cil_remoterst_trigger_int1_a**
6	0x0	**intr_cil_lpdt_int0_a**
5	0x0	**intr_cil_ulps_trigger_int0_a**
4	0x0	**intr_cil_remoterst_trigger_int0_a**
3	0x0	**intr_cil_data_lane_esc_data_rec1_a**
2	0x0	**intr_cil_data_lane_esc_cmd_rec1_a**
1	0x0	**intr_cil_data_lane_esc_data_rec0_a**
0	0x0	**intr_cil_data_lane_esc_cmd_rec0_a**

NVCSI_PHY_1_CILA_INTR_0_MASK_CILA_0
Offset: 0x8502
Byte Offset: 0x21408
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,00x0,0xxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
30	0x0	**intr_mask_cphy_edge_delay_cal_done_trio1_a**
29	0x0	**intr_mask_cphy_edge_delay_cal_done_trio0_a**
28	0x0	**intr_mask_dphy_cil_lane_align_err_a**
27	0x0	**intr_mask_dphy_cil_deskew_calib_err_ctrl_a**
26	0x0	**intr_mask_dphy_cil_deskew_calib_err_lane1_a**
25	0x0	**intr_mask_dphy_cil_deskew_calib_err_lane0_a**
24	0x0	**intr_mask_dphy_cil_deskew_calib_done_ctrl_a**
23	0x0	**intr_mask_dphy_cil_deskew_calib_done_lane1_a**
22	0x0	**intr_mask_dphy_cil_deskew_calib_done_lane0_a**
20	0x0	**intr_mask_cil_data_lane_esc_mode_sync_err1_a**
19	0x0	**intr_mask_cil_data_lane_esc_mode_sync_err0_a**
8	0x0	**intr_mask_cil_data_lane_rxfifo_full_err1_a**
7	0x0	**intr_mask_cil_data_lane_ctrl_err1_a**
6	0x0	**intr_mask_cil_data_lane_sot_mb_err1_a**
5	0x0	**intr_mask_cil_data_lane_sot_sb_err1_a**
4	0x0	**intr_mask_cil_data_lane_rxfifo_full_err0_a**
3	0x0	**intr_mask_cil_data_lane_ctrl_err0_a**
2	0x0	**intr_mask_cil_data_lane_sot_mb_err0_a**
1	0x0	**intr_mask_cil_data_lane_sot_sb_err0_a**
0	0x0	**intr_mask_dphy_cil_clk_lane_ctrl_err_a**

NVCSI_PHY_1_CILA_INTR_1_MASK_CILA_0

Offset: 0x8503
Byte Offset: 0x2140c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**intr_mask_cphy_inadj_cal_done_trio1_a**
13	0x0	**intr_mask_cphy_inadj_cal_done_trio0_a**
12	0x0	**intr_mask_idac_cal_done_trio1_a**
11	0x0	**intr_mask_idac_cal_done_trio0_a**
10	0x0	**intr_mask_dphy_cil_clk_lane_ulpm_req_a**
9	0x0	**intr_mask_cil_lpdt_int1_a**
8	0x0	**intr_mask_cil_ulps_trigger_int1_a**
7	0x0	**intr_mask_cil_remoterst_trigger_int1_a**
6	0x0	**intr_mask_cil_lpdt_int0_a**
5	0x0	**intr_mask_cil_ulps_trigger_int0_a**
4	0x0	**intr_mask_cil_remoterst_trigger_int0_a**
3	0x0	**intr_mask_cil_data_lane_esc_data_rec1_a**
2	0x0	**intr_mask_cil_data_lane_esc_cmd_rec1_a**
1	0x0	**intr_mask_cil_data_lane_esc_data_rec0_a**
0	0x0	**intr_mask_cil_data_lane_esc_cmd_rec0_a**

NVCSI_PHY_1_CILA_CORRECTABLE_ERR_INTR_STATUS_CILA_0

Offset: 0x8504
Byte Offset: 0x21410
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**err_intr_cil_data_lane_esc_mode_sync_err1_a**
13	0x0	**err_intr_cil_data_lane_esc_mode_sync_err0_a**
12	0x0	**err_intr_dphy_cil_lane_align_err_a**
11	0x0	**err_intr_dphy_cil_deskew_calib_err_ctrl_a**
10	0x0	**err_intr_dphy_cil_deskew_calib_err_lane1_a**
9	0x0	**err_intr_dphy_cil_deskew_calib_err_lane0_a**
8	0x0	**err_intr_cil_data_lane_rxfifo_full_err1_a**
7	0x0	**err_intr_cil_data_lane_ctrl_err1_a**
6	0x0	**err_intr_cil_data_lane_sot_mb_err1_a**
5	0x0	**err_intr_cil_data_lane_sot_sb_err1_a**
4	0x0	**err_intr_cil_data_lane_rxfifo_full_err0_a**
3	0x0	**err_intr_cil_data_lane_ctrl_err0_a**
2	0x0	**err_intr_cil_data_lane_sot_mb_err0_a**
1	0x0	**err_intr_cil_data_lane_sot_sb_err0_a**
0	0x0	**err_intr_dphy_cil_clk_lane_ctrl_err_a**

NVCSI_PHY_1_CILA_UNCORRECTABLE_ERR_INTR_STATUS_CILA_0

Offset: 0x8505
Byte Offset: 0x21414
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**err_intr_cil_data_lane_esc_mode_sync_err1_a**
13	0x0	**err_intr_cil_data_lane_esc_mode_sync_err0_a**
12	0x0	**err_intr_dphy_cil_lane_align_err_a**
11	0x0	**err_intr_dphy_cil_deskew_calib_err_ctrl_a**

Bit	Reset	Description
10	0x0	**err_intr_dphy_cil_deskew_calib_err_lane1_a**
9	0x0	**err_intr_dphy_cil_deskew_calib_err_lane0_a**
8	0x0	**err_intr_cil_data_lane_rxfifo_full_err1_a**
7	0x0	**err_intr_cil_data_lane_ctrl_err1_a**
6	0x0	**err_intr_cil_data_lane_sot_mb_err1_a**
5	0x0	**err_intr_cil_data_lane_sot_sb_err1_a**
4	0x0	**err_intr_cil_data_lane_rxfifo_full_err0_a**
3	0x0	**err_intr_cil_data_lane_ctrl_err0_a**
2	0x0	**err_intr_cil_data_lane_sot_mb_err0_a**
1	0x0	**err_intr_cil_data_lane_sot_sb_err0_a**
0	0x0	**err_intr_dphy_cil_clk_lane_ctrl_err_a**

NVCSI_PHY_1_CILA_ERR_INTR_MASK_CILA_0

Offset: 0x8506
Byte Offset: 0x21418
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00007fff (0bxxxx,xxxx,xxxx,xxxx,x111,1111,1111,1111)

Bit	Reset	Description
14	0x1	**err_intr_mask_cil_data_lane_esc_mode_sync_err1_a**
13	0x1	**err_intr_mask_cil_data_lane_esc_mode_sync_err0_a**
12	0x1	**err_intr_mask_dphy_cil_lane_align_err_a**
11	0x1	**err_intr_mask_dphy_cil_deskew_calib_err_ctrl_a**
10	0x1	**err_intr_mask_dphy_cil_deskew_calib_err_lane1_a**
9	0x1	**err_intr_mask_dphy_cil_deskew_calib_err_lane0_a**
8	0x1	**err_intr_mask_cil_data_lane_rxfifo_full_err1_a**
7	0x1	**err_intr_mask_cil_data_lane_ctrl_err1_a**
6	0x1	**err_intr_mask_cil_data_lane_sot_mb_err1_a**

Bit	Reset	Description
5	0x1	**err_intr_mask_cil_data_lane_sot_sb_err1_a**
4	0x1	**err_intr_mask_cil_data_lane_rxfifo_full_err0_a**
3	0x1	**err_intr_mask_cil_data_lane_ctrl_err0_a**
2	0x1	**err_intr_mask_cil_data_lane_sot_mb_err0_a**
1	0x1	**err_intr_mask_cil_data_lane_sot_sb_err0_a**
0	0x1	**err_intr_mask_dphy_cil_clk_lane_ctrl_err_a**

NVCSI_PHY_1_CILA_ERR_INTR_TYPE_CILA_0

Offset: 0x8507
Byte Offset: 0x2141c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**err_intr_type_cil_data_lane_esc_mode_sync_err1_a**
13	0x0	**err_intr_type_cil_data_lane_esc_mode_sync_err0_a**
12	0x0	**err_intr_type_dphy_cil_lane_align_err_a**
11	0x0	**err_intr_type_dphy_cil_deskew_calib_err_ctrl_a**
10	0x0	**err_intr_type_dphy_cil_deskew_calib_err_lane1_a**
9	0x0	**err_intr_type_dphy_cil_deskew_calib_err_lane0_a**
8	0x0	**err_intr_type_cil_data_lane_rxfifo_full_err1_a**
7	0x0	**err_intr_type_cil_data_lane_ctrl_err1_a**
6	0x0	**err_intr_type_cil_data_lane_sot_mb_err1_a**
5	0x0	**err_intr_type_cil_data_lane_sot_sb_err1_a**
4	0x0	**err_intr_type_cil_data_lane_rxfifo_full_err0_a**
3	0x0	**err_intr_type_cil_data_lane_ctrl_err0_a**
2	0x0	**err_intr_type_cil_data_lane_sot_mb_err0_a**
1	0x0	**err_intr_type_cil_data_lane_sot_sb_err0_a**

Bit	Reset	Description
0	0x0	**err_intr_type_dphy_cil_clk_lane_ctrl_err_a**

NVCSI_PHY_1_CILB_INTR_0_STATUS_CILB_0

Offset: 0x8600
Byte Offset: 0x21800
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,00x0,0xxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
30	0x0	**intr_cphy_edge_delay_cal_done_trio1_b**
29	0x0	**intr_cphy_edge_delay_cal_done_trio0_b**
28	0x0	**intr_dphy_cil_lane_align_err_b**
27	0x0	**intr_dphy_cil_deskew_calib_err_ctrl_b**
26	0x0	**intr_dphy_cil_deskew_calib_err_lane1_b**
25	0x0	**intr_dphy_cil_deskew_calib_err_lane0_b**
24	0x0	**intr_dphy_cil_deskew_calib_done_ctrl_b**
23	0x0	**intr_dphy_cil_deskew_calib_done_lane1_b**
22	0x0	**intr_dphy_cil_deskew_calib_done_lane0_b**
20	0x0	**intr_cil_data_lane_esc_mode_sync_err1_b**
19	0x0	**intr_cil_data_lane_esc_mode_sync_err0_b**
8	0x0	**intr_cil_data_lane_rxfifo_full_err1_b**
7	0x0	**intr_cil_data_lane_ctrl_err1_b**
6	0x0	**intr_cil_data_lane_sot_mb_err1_b**
5	0x0	**intr_cil_data_lane_sot_sb_err1_b**
4	0x0	**intr_cil_data_lane_rxfifo_full_err0_b**
3	0x0	**intr_cil_data_lane_ctrl_err0_b**
2	0x0	**intr_cil_data_lane_sot_mb_err0_b**

Bit	Reset	Description
1	0x0	**intr_cil_data_lane_sot_sb_err0_b**
0	0x0	**intr_dphy_cil_clk_lane_ctrl_err_b**

NVCSI_PHY_1_CILB_INTR_1_STATUS_CILB_0

Offset: 0x8601
Byte Offset: 0x21804
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**intr_cphy_inadj_cal_done_trio1_b**
13	0x0	**intr_cphy_inadj_cal_done_trio0_b**
12	0x0	**intr_idac_cal_done_trio1_b**
11	0x0	**intr_idac_cal_done_trio0_b**
10	0x0	**intr_dphy_cil_clk_lane_ulpm_req_b**
9	0x0	**intr_cil_lpdt_int1_b**
8	0x0	**intr_cil_ulps_trigger_int1_b**
7	0x0	**intr_cil_remoterst_trigger_int1_b**
6	0x0	**intr_cil_lpdt_int0_b**
5	0x0	**intr_cil_ulps_trigger_int0_b**
4	0x0	**intr_cil_remoterst_trigger_int0_b**
3	0x0	**intr_cil_data_lane_esc_data_rec1_b**
2	0x0	**intr_cil_data_lane_esc_cmd_rec1_b**
1	0x0	**intr_cil_data_lane_esc_data_rec0_b**
0	0x0	**intr_cil_data_lane_esc_cmd_rec0_b**

NVCSI_PHY_1_CILB_INTR_0_MASK_CILB_0

Offset: 0x8602

Byte Offset: 0x21808
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,00x0,0xxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
30	0x0	**intr_mask_cphy_edge_delay_cal_done_trio1_b**
29	0x0	**intr_mask_cphy_edge_delay_cal_done_trio0_b**
28	0x0	**intr_mask_dphy_cil_lane_align_err_b**
27	0x0	**intr_mask_dphy_cil_deskew_calib_err_ctrl_b**
26	0x0	**intr_mask_dphy_cil_deskew_calib_err_lane1_b**
25	0x0	**intr_mask_dphy_cil_deskew_calib_err_lane0_b**
24	0x0	**intr_mask_dphy_cil_deskew_calib_done_ctrl_b**
23	0x0	**intr_mask_dphy_cil_deskew_calib_done_lane1_b**
22	0x0	**intr_mask_dphy_cil_deskew_calib_done_lane0_b**
20	0x0	**intr_mask_cil_data_lane_esc_mode_sync_err1_b**
19	0x0	**intr_mask_cil_data_lane_esc_mode_sync_err0_b**
8	0x0	**intr_mask_cil_data_lane_rxfifo_full_err1_b**
7	0x0	**intr_mask_cil_data_lane_ctrl_err1_b**
6	0x0	**intr_mask_cil_data_lane_sot_mb_err1_b**
5	0x0	**intr_mask_cil_data_lane_sot_sb_err1_b**
4	0x0	**intr_mask_cil_data_lane_rxfifo_full_err0_b**
3	0x0	**intr_mask_cil_data_lane_ctrl_err0_b**
2	0x0	**intr_mask_cil_data_lane_sot_mb_err0_b**
1	0x0	**intr_mask_cil_data_lane_sot_sb_err0_b**
0	0x0	**intr_mask_dphy_cil_clk_lane_ctrl_err_b**

NVCSI_PHY_1_CILB_INTR_1_MASK_CILB_0

Offset: 0x8603
Byte Offset: 0x2180c
Read/Write: R/W

Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**intr_mask_cphy_inadj_cal_done_trio1_b**
13	0x0	**intr_mask_cphy_inadj_cal_done_trio0_b**
12	0x0	**intr_mask_idac_cal_done_trio1_b**
11	0x0	**intr_mask_idac_cal_done_trio0_b**
10	0x0	**intr_mask_dphy_cil_clk_lane_ulpm_req_b**
9	0x0	**intr_mask_cil_lpdt_int1_b**
8	0x0	**intr_mask_cil_ulps_trigger_int1_b**
7	0x0	**intr_mask_cil_remoterst_trigger_int1_b**
6	0x0	**intr_mask_cil_lpdt_int0_b**
5	0x0	**intr_mask_cil_ulps_trigger_int0_b**
4	0x0	**intr_mask_cil_remoterst_trigger_int0_b**
3	0x0	**intr_mask_cil_data_lane_esc_data_rec1_b**
2	0x0	**intr_mask_cil_data_lane_esc_cmd_rec1_b**
1	0x0	**intr_mask_cil_data_lane_esc_data_rec0_b**
0	0x0	**intr_mask_cil_data_lane_esc_cmd_rec0_b**

NVCSI_PHY_1_CILB_CORRECTABLE_ERR_INTR_STATUS_CILB_0

Offset: 0x8604
Byte Offset: 0x21810
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**err_intr_cil_data_lane_esc_mode_sync_err1_b**
13	0x0	**err_intr_cil_data_lane_esc_mode_sync_err0_b**

Bit	Reset	Description
12	0x0	**err_intr_dphy_cil_lane_align_err_b**
11	0x0	**err_intr_dphy_cil_deskew_calib_err_ctrl_b**
10	0x0	**err_intr_dphy_cil_deskew_calib_err_lane1_b**
9	0x0	**err_intr_dphy_cil_deskew_calib_err_lane0_b**
8	0x0	**err_intr_cil_data_lane_rxfifo_full_err1_b**
7	0x0	**err_intr_cil_data_lane_ctrl_err1_b**
6	0x0	**err_intr_cil_data_lane_sot_mb_err1_b**
5	0x0	**err_intr_cil_data_lane_sot_sb_err1_b**
4	0x0	**err_intr_cil_data_lane_rxfifo_full_err0_b**
3	0x0	**err_intr_cil_data_lane_ctrl_err0_b**
2	0x0	**err_intr_cil_data_lane_sot_mb_err0_b**
1	0x0	**err_intr_cil_data_lane_sot_sb_err0_b**
0	0x0	**err_intr_dphy_cil_clk_lane_ctrl_err_b**

NVCSI_PHY_1_CILB_UNCORRECTABLE_ERR_INTR_STATUS_CILB_0

Offset: 0x8605
Byte Offset: 0x21814
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**err_intr_cil_data_lane_esc_mode_sync_err1_b**
13	0x0	**err_intr_cil_data_lane_esc_mode_sync_err0_b**
12	0x0	**err_intr_dphy_cil_lane_align_err_b**
11	0x0	**err_intr_dphy_cil_deskew_calib_err_ctrl_b**
10	0x0	**err_intr_dphy_cil_deskew_calib_err_lane1_b**
9	0x0	**err_intr_dphy_cil_deskew_calib_err_lane0_b**
8	0x0	**err_intr_cil_data_lane_rxfifo_full_err1_b**

Bit	Reset	Description
7	0x0	**err_intr_cil_data_lane_ctrl_err1_b**
6	0x0	**err_intr_cil_data_lane_sot_mb_err1_b**
5	0x0	**err_intr_cil_data_lane_sot_sb_err1_b**
4	0x0	**err_intr_cil_data_lane_rxfifo_full_err0_b**
3	0x0	**err_intr_cil_data_lane_ctrl_err0_b**
2	0x0	**err_intr_cil_data_lane_sot_mb_err0_b**
1	0x0	**err_intr_cil_data_lane_sot_sb_err0_b**
0	0x0	**err_intr_dphy_cil_clk_lane_ctrl_err_b**

NVCSI_PHY_1_CILB_ERR_INTR_MASK_CILB_0

Offset: 0x8606
Byte Offset: 0x21818
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00007fff (0bxxxx,xxxx,xxxx,xxxx,x111,1111,1111,1111)

Bit	Reset	Description
14	0x1	**err_intr_mask_cil_data_lane_esc_mode_sync_err1_b**
13	0x1	**err_intr_mask_cil_data_lane_esc_mode_sync_err0_b**
12	0x1	**err_intr_mask_dphy_cil_lane_align_err_b**
11	0x1	**err_intr_mask_dphy_cil_deskew_calib_err_ctrl_b**
10	0x1	**err_intr_mask_dphy_cil_deskew_calib_err_lane1_b**
9	0x1	**err_intr_mask_dphy_cil_deskew_calib_err_lane0_b**
8	0x1	**err_intr_mask_cil_data_lane_rxfifo_full_err1_b**
7	0x1	**err_intr_mask_cil_data_lane_ctrl_err1_b**
6	0x1	**err_intr_mask_cil_data_lane_sot_mb_err1_b**
5	0x1	**err_intr_mask_cil_data_lane_sot_sb_err1_b**
4	0x1	**err_intr_mask_cil_data_lane_rxfifo_full_err0_b**
3	0x1	**err_intr_mask_cil_data_lane_ctrl_err0_b**

Bit	Reset	Description
2	0x1	**err_intr_mask_cil_data_lane_sot_mb_err0_b**
1	0x1	**err_intr_mask_cil_data_lane_sot_sb_err0_b**
0	0x1	**err_intr_mask_dphy_cil_clk_lane_ctrl_err_b**

NVCSI_PHY_1_CILB_ERR_INTR_TYPE_CILB_0

Offset: 0x8607
Byte Offset: 0x2181c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**err_intr_type_cil_data_lane_esc_mode_sync_err1_b**
13	0x0	**err_intr_type_cil_data_lane_esc_mode_sync_err0_b**
12	0x0	**err_intr_type_dphy_cil_lane_align_err_b**
11	0x0	**err_intr_type_dphy_cil_deskew_calib_err_ctrl_b**
10	0x0	**err_intr_type_dphy_cil_deskew_calib_err_lane1_b**
9	0x0	**err_intr_type_dphy_cil_deskew_calib_err_lane0_b**
8	0x0	**err_intr_type_cil_data_lane_rxfifo_full_err1_b**
7	0x0	**err_intr_type_cil_data_lane_ctrl_err1_b**
6	0x0	**err_intr_type_cil_data_lane_sot_mb_err1_b**
5	0x0	**err_intr_type_cil_data_lane_sot_sb_err1_b**
4	0x0	**err_intr_type_cil_data_lane_rxfifo_full_err0_b**
3	0x0	**err_intr_type_cil_data_lane_ctrl_err0_b**
2	0x0	**err_intr_type_cil_data_lane_sot_mb_err0_b**
1	0x0	**err_intr_type_cil_data_lane_sot_sb_err0_b**
0	0x0	**err_intr_type_dphy_cil_clk_lane_ctrl_err_b**

NVCSI_PHY_1_NVCSI_CIL_PHY_CTRL_0

Offset: 0x8700
Byte Offset: 0x21c00
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DPHY	**CFG_PHY_MODE**: The CPHY/DPHY type mode. **0** = DPHY **1** = CPHY

NVCSI_PHY_1_LM_SLCG_CTRL_0

Offset: 0x8701
Byte Offset: 0x21c04
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	DISABLE	**cfg_lm1_slcg_override**: Enable the SLCG override for lane merger 1. **0** = DISABLE **1** = ENABLE
0	DISABLE	**cfg_lm0_slcg_override**: Enable the SLCG override for lane merger 0. **0** = DISABLE **1** = ENABLE

NVCSI_PHY_1_LM_EN_CTRL_0

Offset: 0x8702
Byte Offset: 0x21c08
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x0000060c (0bxxxx,xxxx,xxxx,xxxx,xxx0,0110,xx00,11x0)

Bit	Reset	Description
12:9	0x3	**cfg_lm1_water_mark**: Water mark for LM0 detect stuck
8	DISABLE	**cfg_lm1_en**: Lane Merger 1 enable **0** = DISABLE **1** = ENABLE
5:2	0x3	**cfg_lm0_water_mark**: Water mark for LM0 detect stuck
0	DISABLE	**cfg_lm0_en**: Lane Merger 0 enable **0** = DISABLE **1** = ENABLE

NVCSI_PHY_1_LM_SW_RESET_0

Offset: 0x8703
Byte Offset: 0x21c0c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	DISABLE	**cfg_lm1_swreset**: Reset the lane merger 1. **0** = DISABLE **1** = ENABLE
0	DISABLE	**cfg_lm0_swreset**: Reset the lane merger 0. **0** = DISABLE **1** = ENABLE

NVCSI_PHY_1_NVCSI_CIL_CONFIG_0

Offset: 0x8704
Byte Offset: 0x21c10
Read/Write: R/W

Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,x000,xxxx,x000)

Bit	Reset	Description
10:8	0x0	**DATA_LANE_B**: Lane number for lane merger B. 000: 0 lanes active 001: 1 lanes active 010: 2 lanes active Others: illegal
2:0	0x0	**DATA_LANE_A**: Lane number for lane merger A. 000: 0 lanes active 001: 1 lanes active 010: 2 lanes active 011: 3 lanes active (valid only in CPHY mode. NA in DPHY mode) 100: 4 lanes active, CLK from Partition A is used. 101: 4 lanes active, CLK from Partition B is used. Others: illegal

NVCSI_PHY_1_NVCSI_CIL_CLKEN_OVERRIDE_CTRL_0

Offset: 0x8705
Byte Offset: 0x21c14
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**CLKEN_OVERRIDE**: CLKENABLE OVERRIDE FOR CIL. **1** = ENABLE **0** = DISABLE

NVCSI_PHY_1_NVCSI_CIL_PAD_CONFIG_0

Offset: 0x8706
Byte Offset: 0x21c18
Read/Write: R/W
Parity Protection: Y
Shadow: N

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00040100 (0bxxxx,x000,0000,0100,0000,xxx1,0000,0000)

Bit	Reset	Description
26:25	0x0	**VD09REG_LEVEL**: 2-bit regulator level control. control voltage level output of 0.9v regulator. 00:0.9v, 01:0.85v, 10: 0.95v, 11: 1v
24:23	0x0	**VD09REG_LEAKER**: 2-bit regulator leaker control. 11 will give best performance, but consumers more DC quiescent current. (control 0.9v internal regulator for all partitions)
22:21	0x0	**VD04REG_LEVEL**: 2-bit regulator level control control voltage level output of 0.4V regulator 00: 0.4v 01: 0.38V 10: 0.42V 11: 0.44V
20:19	0x0	**VD04REG_LEAKER**: 2-bit regulator leaker control. 11 will give best performance, but consumers more DC quiescent current. (control 0.4v internal tx regulator for all partitions)
18:16	0x4	**LPRX_LEVEL_SEL**: Internal vref adjustment for LP self-biased receiver(when LP_RX_SELECT_IO1_[A/B]=1)
15:12	0x0	**LOADADJ**: LOAD ADJ value to be connected to the pad not used for CSI functional mode. used only for IO bist
8	0x1	**PDVCLAMP**: Power down regular which supplies current to de-serializer logic. Active High.
7:5	0x0	**SEL_CKTEST**
4	0x0	**E_CKTEST**: Enable clock test output.
3:1	0x0	**SEL_ATEST**
0	0x0	**E_ATEST**: Enable analog test voltage output.

NVCSI_PHY_1_NVCSI_CIL_PAD_CONFIG_1_0

Offset: 0x8707
Byte Offset: 0x21c1c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE_TOP**: spare register bits for top level control

NVCSI_PHY_1_NVCSI_CIL_LANE_SWIZZLE_CTRL_0

Offset: 0x8708
Byte Offset: 0x21c20
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4:0	0x0	**LANE_SWIZZLE_CTRL**: Lane Swizzle control for Brick. Valid in both CPHY and DPHY modes. For CPHY, all modes are supported. But for DPHY, some mode are not supported with different lane count. Only for 4 lanes: 00011 A0 A1 B0 B1 --> A0 B0 A1 B1 00100 A0 A1 B0 B1 --> A0 B1 A1 B0 00101 A0 A1 B0 B1 --> A0 B1 B0 A1 00010 A0 A1 B0 B1 --> A0 B0 B1 A1 01000 A0 A1 B0 B1 --> A1 B0 B1 A0 01001 A0 A1 B0 B1 --> A1 B0 A0 B1 01010 A0 A1 B0 B1 --> A1 B1 A0 B0 01011 A0 A1 B0 B1 --> A1 B1 B0 A0 01100 A0 A1 B0 B1 --> B0 A1 A0 B1 01101 A0 A1 B0 B1 --> B0 A1 B1 A0 01110 A0 A1 B0 B1 --> B0 A0 B1 A1 01111 A0 A1 B0 B1 --> B0 A0 A1 B1 10000 A0 A1 B0 B1 --> B0 B1 A1 A0 10001 A0 A1 B0 B1 --> B0 B1 A0 A1 10010 A0 A1 B0 B1 --> B1 A1 B0 A0 10011 A0 A1 B0 B1 --> B1 A1 A0 B0 10100 A0 A1 B0 B1 --> B1 B0 A0 A1 10101 A0 A1 B0 B1 --> B1 B0 A1 A0 10110 A0 A1 B0 B1 --> B1 A0 A1 B0 10111 A0 A1 B0 B1 --> B1 A0 B0 A1 Support for 1/2/4 lanes 00000 A0 A1 B0 B1 --> A0 A1 B0 B1 00001 A0 A1 B0 B1 --> A0 A1 B1 B0 00110 A0 A1 B0 B1 --> A1 A0 B0 B1 00111 A0 A1 B0 B1 --> A1 A0 B1 B0

NVCSI_PHY_1_NVCSI_CIL_BK_MODE_STATUS_0

Offset: 0x8709

Byte Offset: 0x21c24
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
1:0	X	**BK_MODE**: 00: two independent 2x bricks 01: one 4x brick, received clock from partition A is used. Clock from partition B is not used 10: one 4x brick, received clock from partition B is used. Clock from partition A is not used 11: illegal

NVCSI_PHY_1_NVCSI_CIL_TX_TIMING_0_0

Offset: 0x870a
Byte Offset: 0x21c28
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00006030 (0bxxxx,xxxx,xxxx,xxxx,0110,0000,0011,0000)

Bit	Reset	Description
15:8	0x60	**THSEXIT**: THSEXIT length, in slow clock cycle number
7:0	0x30	**TLPX**: TLPX length, in slow clock cycle number

NVCSI_PHY_1_NVCSI_CIL_TX_TIMING_1_0

Offset: 0x870b
Byte Offset: 0x21c2c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00101030 (0bxxxx,xxxx,0001,0000,0001,0000,0011,0000)

Bit	Reset	Description
23:16	0x10	**T3POST**: T3POST length, in slow clock cycle number
15:8	0x10	**T3PREBEGIN**: T3PREBEGIN length, in slow clock cycle number
7:0	0x30	**T3PREPARE**: T3PREPARE length, in slow clock cycle number

NVCSI_PHY_1_NVCSI_CIL_TX_TIMING_2_0

Offset: 0x870c
Byte Offset: 0x21c30
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:8	0xfff	**T3CALALTSEQ**: T3CALALTSEQ length, in slow clock cycle number
7:0	0xff	**T3CALPREAMBLE**: T3CALPREAMBLE length, in slow clock cycle number

NVCSI_PHY_1_NVCSI_CIL_TX_CALIB_CTRL_0

Offset: 0x870d
Byte Offset: 0x21c34
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000004 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0100)

Bit	Reset	Description
17:2	0x1	**CALIB_SEED**: Calibration sequence format 2 PRBS seed. Should set to a non-zero value.

Bit	Reset	Description
1	FMT_1	**CALIB_SEQ**: Calibration sequence format. **0** = FMT_1 **1** = FMT_2
0	DISABLE	**CALIB_EN**: Calibration sequence enable **0** = DISABLE **1** = ENABLE

NVCSI_PHY_1_NVCSI_CIL_TX_CTRL_0

Offset: 0x870e
Byte Offset: 0x21c38
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000004 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0100)

Bit	Reset	Description
4	DISABLE	**TX_ENABLE**: Enable the TX SCIL. **0** = DISABLE **1** = ENABLE
3	SINGLE	**PAD_BK_MODE**: Set the pad to one stream or two independent stream usage. **0** = SINGLE **1** = DUAL
2:0	0x4	**LANE_NUM**: Valid value: 1/2/4

NVCSI_PHY_1_NVCSI_CIL_TX_SW_RESET_0_0

Offset: 0x870f
Byte Offset: 0x21c3c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	ENABLE	**SW_RESET**: Reset the TX lane **0** = DISABLE **1** = ENABLE

NVCSI_PHY_1_NVCSI_CIL_CPHY_BIST_CONFIG_0_0

Offset: 0x8710
Byte Offset: 0x21c40
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000009 (0bxxxx,xxxx,xx00,0000,0000,0000,0000,1001)

Bit	Reset	Description
21	DISABLE	**REALIGN_EN**: If set to enable, the BIST logic will re-align the reference symbol to match the received symbol when there is clock slip by wire state error. **0** = DISABLE **1** = ENABLE
20:3	0x1	**PRBS_SEED**: Seed of the PRBS pattern. Should set to a non-zero value.
2:1	0x0	**PRBS_PATTERN**: PRBS pattern PRBS9: X0+X5+X9 PRBS11: X0+X9+X11 PRBS18: X0+X11+X18 **0** = PRBS9 **1** = PRBS11 **2** = PRBS18
0	MODE1	**BIST_MODE**: BIST control mode MODE0: Set the expected error number, BIST logic will keep comparing the received symbols and the reference PRBS generated symbols. When the expected error number got, BIST logic stop. And report how many word are totally compared. MODE11: Set the expected word number, BIST logic will keep comparing the received symbols and the reference PRBS generated symbols. When the expected words(symbols) are compared, BIST logic stop and report how many errors are found. **0** = MODE0 **1** = MODE1

NVCSI_PHY_1_NVCSI_CIL_CPHY_BIST_CONFIG_1_0

Offset: 0x8711
Byte Offset: 0x21c44
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000100 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx1,0000,0000)

Bit	Reset	Description
8	0x1	**ERR_TYPE**: The expecte error number type. SYMBOL_NUM: symbol error number. **0** = WORD_NUM :WORD_NUM: word error number. **1** = SYMBOL_NUM
7:0	0x0	**ERR_NUM**: The expected error number, used for BIST MODE 0.

NVCSI_PHY_1_NVCSI_CIL_CPHY_BIST_CONFIG_2_0

Offset: 0x8712
Byte Offset: 0x21c48
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WORD_NUM_LOW**: The low 32 bits of the expect word number, used for BIST MODE 1.

NVCSI_PHY_1_NVCSI_CIL_CPHY_BIST_CONFIG_3_0

Offset: 0x8713
Byte Offset: 0x21c4c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WORD_NUM_HIGH**: The higher 16 bits of the expected word number, used for BIST MODE 1.

NVCSI_PHY_1_NVCSI_CIL_IDAC_CALIB_START_0

Offset: 0x8714
Byte Offset: 0x21c50
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**IDAC_CAL_TRIGGER**: Trigger the IDAC calibration.

NVCSI_PHY_1_NVCSI_CIL_IDAC_CALIB_CTRL_0

Offset: 0x8715
Byte Offset: 0x21c54
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,0000,xxxx,x000,0000,0101)

Bit	Reset	Description
19	0x0	**IDAC_CAL_EN_TRIO1_B**: Enable IDAC calibration for trio B1
18	0x0	**IDAC_CAL_EN_TRIO0_B**: Enable IDAC calibration for trio B0
17	0x0	**IDAC_CAL_EN_TRIO1_A**: Enable IDAC calibration for trio A1
16	0x0	**IDAC_CAL_EN_TRIO0_A**: Enable IDAC calibration for trio A0

Bit	Reset	Description
10:0	0x5	**IDAC_SETTLE_TIME**: How many cycle for settle to sample the done from pad.

NVCSI_PHY_1_NVCSI_CIL_IDAC_CALIB_STATUS_0

Offset: 0x8716
Byte Offset: 0x21c58
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,0000,0000,0000,xxxx,0000,0000,0000)

Bit	Reset	Description
27	0x0	**IDAC_CALIB_MAXOUT_RXCA_TRIO1_B**: IDAC calibration maxout for trio B1 RXCA.
26	0x0	**IDAC_CALIB_MAXOUT_RXBC_TRIO1_B**: IDAC calibration maxout for trio B1 RXBC.
25	0x0	**IDAC_CALIB_MAXOUT_RXAB_TRIO1_B**: IDAC calibration maxout for trio B1 RXAB.
24	0x0	**IDAC_CALIB_MAXOUT_RXCA_TRIO0_B**: IDAC calibration maxout for trio B0 RXCA.
23	0x0	**IDAC_CALIB_MAXOUT_RXBC_TRIO0_B**: IDAC calibration maxout for trio B0 RXBC.
22	0x0	**IDAC_CALIB_MAXOUT_RXAB_TRIO0_B**: IDAC calibration maxout for trio B0 RXAB.
21	0x0	**IDAC_CALIB_MAXOUT_RXCA_TRIO1_A**: IDAC calibration maxout for trio A1 RXCA.
20	0x0	**IDAC_CALIB_MAXOUT_RXBC_TRIO1_A**: IDAC calibration maxout for trio A1 RXBC.
19	0x0	**IDAC_CALIB_MAXOUT_RXAB_TRIO1_A**: IDAC calibration maxout for trio A1 RXAB.
18	0x0	**IDAC_CALIB_MAXOUT_RXCA_TRIO0_A**: IDAC calibration maxout for trio A0 RXCA.
17	0x0	**IDAC_CALIB_MAXOUT_RXBC_TRIO0_A**: IDAC calibration maxout for trio A0 RXBC.
16	0x0	**IDAC_CALIB_MAXOUT_RXAB_TRIO0_A**: IDAC calibration maxout for trio A0 RXAB.
11	0x0	**IDAC_CALIB_DONE_RXCA_TRIO1_B**: IDAC calibration done for trio B1 RXCA.

Bit	Reset	Description
10	0x0	**IDAC_CALIB_DONE_RXBC_TRIO1_B**: IDAC calibration done for trio B1 RXBC.
9	0x0	**IDAC_CALIB_DONE_RXAB_TRIO1_B**: IDAC calibration done for trio B1 RXAB.
8	0x0	**IDAC_CALIB_DONE_RXCA_TRIO0_B**: IDAC calibration done for trio B0 RXCA.
7	0x0	**IDAC_CALIB_DONE_RXBC_TRIO0_B**: IDAC calibration done for trio B0 RXBC.
6	0x0	**IDAC_CALIB_DONE_RXAB_TRIO0_B**: IDAC calibration done for trio B0 RXAB.
5	0x0	**IDAC_CALIB_DONE_RXCA_TRIO1_A**: IDAC calibration done for trio A1 RXCA.
4	0x0	**IDAC_CALIB_DONE_RXBC_TRIO1_A**: IDAC calibration done for trio A1 RXBC.
3	0x0	**IDAC_CALIB_DONE_RXAB_TRIO1_A**: IDAC calibration done for trio A1 RXAB.
2	0x0	**IDAC_CALIB_DONE_RXCA_TRIO0_A**: IDAC calibration done for trio A0 RXCA.
1	0x0	**IDAC_CALIB_DONE_RXBC_TRIO0_A**: IDAC calibration done for trio A0 RXBC.
0	0x0	**IDAC_CALIB_DONE_RXAB_TRIO0_A**: IDAC calibration done for trio A0 RXAB.

NVCSI_PHY_1_NVCSI_CIL_IDAC_CALIB_STATUS1_0

Offset: 0x8717
Byte Offset: 0x21c5c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x21084210 (0bxx10,0001,0000,1000,0100,0010,0001,0000)

Bit	Reset	Description
29:25	0x10	**RXCA_IDACADJ_TRIO1_A**: IDAC calibration result for trio A1 RXCA
24:20	0x10	**RXBC_IDACADJ_TRIO1_A**: IDAC calibration result for trio A1 RXBC
19:15	0x10	**RXAB_IDACADJ_TRIO1_A**: IDAC calibration result for trio A1 RXAB

Bit	Reset	Description
14:10	0x10	**RXCA_IDACADJ_TRIO0_A**: IDAC calibration result for trio A0 RXCA
9:5	0x10	**RXBC_IDACADJ_TRIO0_A**: IDAC calibration result for trio A0 RXBC
4:0	0x10	**RXAB_IDACADJ_TRIO0_A**: IDAC calibration result for trio A0 RXAB

NVCSI_PHY_1_NVCSI_CIL_IDAC_CALIB_STATUS2_0

Offset: 0x8718
Byte Offset: 0x21c60
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x21084210 (0bxx10,0001,0000,1000,0100,0010,0001,0000)

Bit	Reset	Description
29:25	0x10	**RXCA_IDACADJ_TRIO1_B**: IDAC calibration result for trio B1 RXCA
24:20	0x10	**RXBC_IDACADJ_TRIO1_B**: IDAC calibration result for trio B1 RXBC
19:15	0x10	**RXAB_IDACADJ_TRIO1_B**: IDAC calibration result for trio B1 RXAB
14:10	0x10	**RXCA_IDACADJ_TRIO0_B**: IDAC calibration result for trio B0 RXCA
9:5	0x10	**RXBC_IDACADJ_TRIO0_B**: IDAC calibration result for trio B0 RXBC
4:0	0x10	**RXAB_IDACADJ_TRIO0_B**: IDAC calibration result for trio B0 RXAB

NVCSI_PHY_1_NVCSI_CIL_IDAC_OVERRIDE_A_0

Offset: 0x8719
Byte Offset: 0x21c64
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x84218421 (0b1000,0100,0010,0001,1000,0100,0010,0001)

Bit	Reset	Description
31:27	0x10	**IDAC_VALUE_RXCA_TRIO1_A**: The override value
26:22	0x10	**IDAC_VALUE_RXBC_TRIO1_A**: The override value
21:17	0x10	**IDAC_VALUE_RXAB_TRIO1_A**: The override value
16	0x1	**SW_SET_IDAC_TRIO1_A**: Enable the SW override of the IDAC value
15:11	0x10	**IDAC_VALUE_RXCA_TRIO0_A**: The override value
10:6	0x10	**IDAC_VALUE_RXBC_TRIO0_A**: The override value
5:1	0x10	**IDAC_VALUE_RXAB_TRIO0_A**: The override value
0	0x1	**SW_SET_IDAC_TRIO0_A**: Enable the SW override of the IDAC value

NVCSI_PHY_1_NVCSI_CIL_IDAC_OVERRIDE_B_0

Offset: 0x871a
Byte Offset: 0x21c68
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x84218421 (0b1000,0100,0010,0001,1000,0100,0010,0001)

Bit	Reset	Description
31:27	0x10	**IDAC_VALUE_RXCA_TRIO1_B**: The override value
26:22	0x10	**IDAC_VALUE_RXBC_TRIO1_B**: The override value
21:17	0x10	**IDAC_VALUE_RXAB_TRIO1_B**: The override value
16	0x1	**SW_SET_IDAC_TRIO1_B**: Enable the SW override of the IDAC value
15:11	0x10	**IDAC_VALUE_RXCA_TRIO0_B**: The override value

Bit	Reset	Description
10:6	0x10	**IDAC_VALUE_RXBC_TRIO0_B**: The override value
5:1	0x10	**IDAC_VALUE_RXAB_TRIO0_B**: The override value
0	0x1	**SW_SET_IDAC_TRIO0_B**: Enable the SW override of the IDAC value

NVCSI_PHY_1_NVCSI_CIL_IDAC_CALIB_DEBUG_0

Offset: 0x871b
Byte Offset: 0x21c6c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000XXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
11	X	**ZI_HSRX_RXCA_TRIO1_B**
10	X	**ZI_HSRX_RXBC_TRIO1_B**
9	X	**ZI_HSRX_RXAB_TRIO1_B**
8	X	**ZI_HSRX_RXCA_TRIO0_B**
7	X	**ZI_HSRX_RXBC_TRIO0_B**
6	X	**ZI_HSRX_RXAB_TRIO0_B**
5	X	**ZI_HSRX_RXCA_TRIO1_A**
4	X	**ZI_HSRX_RXBC_TRIO1_A**
3	X	**ZI_HSRX_RXAB_TRIO1_A**
2	X	**ZI_HSRX_RXCA_TRIO0_A**
1	X	**ZI_HSRX_RXBC_TRIO0_A**
0	X	**ZI_HSRX_RXAB_TRIO0_A**

NVCSI_PHY_1_NVCSI_CIL_TEST_CONTROL_0

Offset: 0x871c
Byte Offset: 0x21c70

Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**HS_EN_DATA_LANE1_B**: Force to enable the e_input_hs of the data lane b1, used for impedance measurements of HS mode
4	0x0	**HS_EN_DATA_LANE0_B**: Force to enable the e_input_hs of the data lane b0, used for impedance measurements of HS mode
3	0x0	**HS_EN_DATA_LANE1_A**: Force to enable the e_input_hs of the data lane a1, used for impedance measurements of HS mode
2	0x0	**HS_EN_DATA_LANE0_A**: Force to enable the e_input_hs of the data lane a0, used for impedance measurements of HS mode
1	0x0	**HS_EN_CLK_LANE_B**: Force to enable the e_input_hs of the clock lane b, used for impedance measurements of HS mode
0	0x0	**HS_EN_CLK_LANE_A**: Force to enable the e_input_hs of the clock lane a, used for impedance measurements of HS mode

NVCSI_PHY_1_NVCSI_CIL_PULLDN_CONTROL_0

Offset: 0x871d
Byte Offset: 0x21c74
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**E_PULLDN_IO1_B**
4	0x0	**E_PULLDN_IO0_B**
3	0x0	**E_PULLDN_IO1_A**
2	0x0	**E_PULLDN_IO0_A**
1	0x0	**E_PULLDN_CLK_B**

Bit	Reset	Description
0	0x0	**E_PULLDN_CLK_A**

NVCSI_PHY_1_NVCSI_CIL_SPARE_0

Spare register
Offset: 0x871e
Byte Offset: 0x21c78
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**phy_reg**

NVCSI_PHY_1_NVCSI_CIL_A_SW_RESET_0

Offset: 0x871f
Byte Offset: 0x21c7c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**SW_RESET1_A**: SOFT RESET FOR LANE A1. Active High **1** = ENABLE **0** = DISABLE
0	0x0	**SW_RESET0_A**: SOFT RESET FOR LANE A0. Active High **1** = ENABLE **0** = DISABLE

NVCSI_PHY_1_NVCSI_CIL_A_CLKEN_OVERRIDE_CTRL_0

Offset: 0x8720
Byte Offset: 0x21c80
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**CLKEN_OVERRIDE1_A**: CLKENABLE OVERRIDE FOR LANE A1. Active High **1** = ENABLE **0** = DISABLE
0	0x0	**CLKEN_OVERRIDE0_A**: CLKENABLE OVERRIDE FOR LANE A0. Active High **1** = ENABLE **0** = DISABLE

NVCSI_PHY_1_NVCSI_CIL_A_CTLE_CTRL_0

Offset: 0x8721
Byte Offset: 0x21c84
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000888 (0bxxxx,xxxx,0000,0000,0000,1000,1000,1000)

Bit	Reset	Description
23:20	0x0	**AFE_DCGAIN_IO1_A**: RX AFE DC gain control
19:16	0x0	**AFE_DCGAIN_IO0_A**: RX AFE DC gain control
15:12	0x0	**AFE_DCGAIN_CLK_A**: RX AFE DC gain control
11:8	0x8	**AFE_HFGAIN_IO1_A**: RX AFE_CTLE high frequency gain control, 16 curves, ac_gain=code*1db
7:4	0x8	**AFE_HFGAIN_IO0_A**: RX AFE_CTLE high frequency gain control, 16 curves, ac_gain=code*1db

Bit	Reset	Description
3:0	0x8	**AFE_HFGAIN_CLK_A**: RX AFE_CTLE high frequency gain control, 16 curves, ac_gain=code*1db

NVCSI_PHY_1_NVCSI_CIL_A_CTLE_CTRL1_0

Offset: 0x8722
Byte Offset: 0x21c88
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000554 (0bxxxx,xxxx,0000,0000,0000,0101,0101,0100)

Bit	Reset	Description
23:20	0x0	**AFE_CTRL_IO1_A**
19:16	0x0	**AFE_CTRL_IO0_A**
15:12	0x0	**AFE_CTRL_CLK_A**
11:10	0x1	**AFE_CURRENT_IO1_A**
9:8	0x1	**AFE_CURRENT_IO0_A**
7:6	0x1	**AFE_CURRENT_CLK_A**
5:4	0x1	**AFE_FREQBAND_IO1_A**: RX AFE bandwidth selection. 2'b00: <1.5GS/s, 2'b01: 1.5~3GS/s, 2'b10: 3~5GS/s, 2'b11: reserved.
3:2	0x1	**AFE_FREQBAND_IO0_A**: RX AFE bandwidth selection. 2'b00: <1.5GS/s, 2'b01: 1.5~3GS/s, 2'b10: 3~5GS/s, 2'b11: reserved.
1:0	0x0	**AFE_FREQBAND_CLK_A**: RX AFE bandwidth selection. 2'b00: <1.5GS/s, 2'b01: 1.5~3GS/s, 2'b10: 3~5GS/s, 2'b11: reserved.

NVCSI_PHY_1_NVCSI_CIL_A_PAD_CONFIG_0

Offset: 0x8723
Byte Offset: 0x21c8c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00070000 (0b0000,0000,0000,x111,xxxx,0000,0000,0000)

Bit	Reset	Description
31	0x0	**LP_RX_SELECT_IO1_A**: LP mode receiver select for partition A lane 1. DPHY mode: controls IO1P_A and IO1N_A CPHY mode: controls IO1P_A, IO1N_A and IOCLKN_A.
30	0x0	**LP_RX_SELECT_IO0_A**: LP mode receiver select for partition A lane 0. DPHY mode: controls IO0P_A and IO0N_A CPHY mode: controls IO0P_A, IO0N_A and IOCLKP_A.
29	0x0	**LP_RX_SELECT_CLK_A**: LP mode receiver select. 0: Schmitt receiver. 1: self-biased receiver. DPHY mode: controls IOCLKP_A and IOCLKN_A. CPHY mode: dummy.
28	0x0	**E_LPRX_HYS_N_IO1_A**: Enable LP-RX hysteresis. 0: enable hysteresis 1: disable hysteresis
27	0x0	**E_LPRX_HYS_N_IO0_A**: Enable LP-RX hysteresis. 0: enable hysteresis 1: disable hysteresis
26	0x0	**E_LPRX_HYS_N_CLK_A**: Enable LP-RX hysteresis. 0: enable hysteresis 1: disable hysteresis
25	0x0	**E_HS_TERM_N_IO1_A**: Enable the HS termination of Partition A Lane 1.
24	0x0	**E_HS_TERM_N_IO0_A**: Enable the HS termination of Partition A Lane 0.
23	0x0	**E_HS_TERM_N_CLK_A**: Enable the HS termination of Clock partition A.
22	0x0	**E_INPUT_LP_IO1_A**: Enable LP receiver of Partition A Lane 1 Applicable in both CPHY and DPHY case.
21	0x0	**E_INPUT_LP_IO0_A**: Enable LP receiver of Partition A Lane 0 Applicable in both CPHY and DPHY case.
20	0x0	**E_INPUT_LP_CLK_A**: enable LP receiver of Clock partition A Applicable in DPHY case. N/A for CPHY
18	0x1	**PD_CLK_A**: Power down for CLk of Partition A. Applicable in DPHY case. N/A for CPHY
17	0x1	**PD_IO1_A**: Power down for Trio 1 and Lane 1 of Partition A. Applicable in both CPHY and DPHY case.

Bit	Reset	Description
16	0x1	**PD_IO0_A**: Power down for Trio 0 and Lane 0 of Partition A. Applicable in both CPHY and DPHY case.
11:8	0x0	**SPARE_CLK_A**: Spare control bits for pad control. Bit 11 is used as E_PULLDN_CLK_A control (0 - Pull down of Clock Lane enabled, 1 - Pull down of Clock Lane enabled). Functionality of other bits currently not defined. Undefined register bits are reserved for future use. Pull down may be enabled when Clock Lane is unused to minimized interference.
7:4	0x0	**SPARE_IO1_A**: Spare control bits for pad control. Functionality of other bits currently not defined. Undefined register bits are reserved for future use. Pull down may be enabled when Data Lane-1 is unused to minimized interference.
3:0	0x0	**SPARE_IO0_A**: Spare control bits for pad control. Functionality of other bits currently not defined. Undefined register bits are reserved for future use. Pull down may be enabled when Data Lane-0 is unused to minimized interference.

NVCSI_PHY_1_NVCSI_CIL_A_PAD_CONFIG_1_0

Offset: 0x8724
Byte Offset: 0x21c90
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000002 (0b0000,0000,0000,0000,0000,0000,xx00,0010)

Bit	Reset	Description
31:24	0x0	**OUTADJ_IO1_A**: Trio 1 output delay trimmer, each tap delays 9ps
23:16	0x0	**OUTADJ_IO0_A**: Trio 0 output delay trimmer, each tap delays 9ps
15:8	0x0	**OUTADJ_CLK_A**: Clock bit output delay trimmer, each tap delays 9ps
5	0x0	**HS_BSO_A**: 1= power-on 0.4v-TX-regulator during LP-mode; 0= power-off 0.4v-TX-regulator during LP-mode;
4	0x0	**CPHY_MID_STRENGTH_TRIO1_A**: 0: when bump is driven to mid-voltage-level, both 100-ohm-pull-up and 100-ohm-pull-down is turned on to drive mid-voltage-level; 1: when bump is driven to mid-voltage-level, both pull-up and pull-down are turned off to become tri-stated;

Bit	Reset	Description
3	0x0	**CPHY_MID_STRENGTH_TRIO0_A**: 0: when bump is driven to mid-voltage-level, both 100-ohm-pull-up and 100-ohm-pull-down is turned on to drive mid-voltage-level; 1: when bump is driven to mid-voltage-level, both pull-up and pull-down are turned off to become tri-stated;
2	DISABLE	**REV_CLK_A**: Reverse clock polarity **0** = DISABLE **1** = ENABLE
1	ENABLE	**E_PREDRV_FLOP_RESET_A**: HS predriver flop reset **0** = DISABLE **1** = ENABLE
0	DISABLE	**FCZERO_A**: Force clk bit to drive differential 0 **0** = DISABLE **1** = ENABLE

NVCSI_PHY_1_NVCSI_CIL_A_PAD_CONFIG_2_0

Offset: 0x8725
Byte Offset: 0x21c94
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,0000,0000,0000,xxxx,0000,0000,0000)

Bit	Reset	Description
27:24	0x0	**PEMPD_IO1_A**: Enable trio 1 HS driver pull down pre-emphasis.
23:20	0x0	**PEMPD_IO0_A**: Enable trio 0 HS driver pull down pre-emphasis.
19:16	0x0	**PEMPD_CLK_A**: Enable clock bit HS driver pull down pre-emphasis.
11:8	0x0	**PEMPU_IO1_A**: Enable trio 1 HS driver pull up pre-emphasis.
7:4	0x0	**PEMPU_IO0_A**: Enable trio 0 HS driver pull up pre-emphasis.
3:0	0x0	**PEMPU_CLK_A**: Enable clock bit HS driver pull up pre-emphasis.

NVCSI_PHY_1_NVCSI_CIL_A_PAD_CONFIG_3_0

Offset: 0x8726
Byte Offset: 0x21c98
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x06660666 (0bxxxx,0110,0110,0110,xxxx,0110,0110,0110)

Bit	Reset	Description
27:24	0x6	**LPDNADJ_IO1_A**: Driver pull down impedance control
23:20	0x6	**LPDNADJ_IO0_A**: Driver pull down impedance control
19:16	0x6	**LPDNADJ_CLK_A**: Driver pull down impedance control
11:8	0x6	**LPUPADJ_IO1_A**: Driver pull up impedance control
7:4	0x6	**LPUPADJ_IO0_A**: Driver pull up impedance control
3:0	0x6	**LPUPADJ_CLK_A**: Driver pull up impedance control

NVCSI_PHY_1_NVCSI_CIL_A_PAD_CONFIG_4_0

Offset: 0x8727
Byte Offset: 0x21c9c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,0000,0000,0000,xxxx,0000,0000,0000)

Bit	Reset	Description
27:24	0x0	**SLEWDNADJ_IO1_A**: Pull down slew rate adjust
23:20	0x0	**SLEWDNADJ_IO0_A**: Pull down slew rate adjust
19:16	0x0	**SLEWDNADJ_CLK_A**: Pull down slew rate adjust

Bit	Reset	Description
11:8	0x0	**SLEWUPADJ_IO1_A**: Pull up slew rate adjust
7:4	0x0	**SLEWUPADJ_IO0_A**: Pull up slew rate adjust
3:0	0x0	**SLEWUPADJ_CLK_A**: Pull up slew rate adjust

NVCSI_PHY_1_NVCSI_CIL_A_PAD_CONFIG_5_0

Offset: 0x8728
Byte Offset: 0x21ca0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000110 (0bxxxx,xxxx,xxxx,xxxx,xxxx,x001,0001,0000)

Bit	Reset	Description
10:7	0x2	**CDDNADJ_IO1_A**: Level adjust on low limit of detection
6:3	0x2	**CDDNADJ_IO0_A**: Level adjust on low limit of detection
2	DISABLE	**E_CD_IO1_A**: Data bits contention detector enable **0** = DISABLE **1** = ENABLE
1	DISABLE	**E_CD_IO0_A**: Data bits contention detector enable **0** = DISABLE **1** = ENABLE
0	DISABLE	**E_CD_CLK_A**: Clock bit contention detector enable **0** = DISABLE **1** = ENABLE

NVCSI_PHY_1_NVCSI_CIL_A_PAD_CONFIG_6_0

Offset: 0x8729
Byte Offset: 0x21ca4
Read/Write: R/W

Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**E_DEEP_LPBK_A**: Enable deep loopback
0	0x0	**E_SHALLOW_LPBK_A**: Enalbe shallow loopback

NVCSI_PHY_1_NVCSI_CIL_A_PAD_CD_STATUS_0

Offset: 0x872a
Byte Offset: 0x21ca8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**CDN_IO1_A**: trio1 n output, 1=contention detected.
4	0x0	**CDN_IO0_A**: trio0 n output, 1=contention detected.
3	0x0	**CDN_CLK_A**: Clock bit n output, 1=contention detected.
2	0x0	**CDP_IO1_A**: trio1 p output, 1=contention detected.
1	0x0	**CDP_IO0_A**: trio0 p output, 1=contention detected.
0	0x0	**CDP_CLK_A**: Clock bit p output, 1=contention detected.

NVCSI_PHY_1_NVCSI_CIL_A_DPHY_INADJ_CTRL_0

Offset: 0x872b
Byte Offset: 0x21cac
Read/Write: R/W
Parity Protection: Y

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,x000,0000,x000,0000,x000,0000)

Bit	Reset	Description
22	0x0	**SW_SET_DPHY_INADJ_CLK_A**: Software set for clock input delay trimmer
21:16	0x0	**DPHY_INADJ_CLK_A**: Programmable value for CLK input delay trimmer,
14	0x0	**SW_SET_DPHY_INADJ_IO1_A**: Software override for bit 1 input delay trimmer 1 - SW programmed value is applied to the pads. 2 - Trimmer value from HW calibration state machine is applied to the pads
13:8	0x0	**DPHY_INADJ_IO1_A**: Programmable value for bit 1 input delay trimmer, each tap delay of 9 ps
6	0x0	**SW_SET_DPHY_INADJ_IO0_A**: Software override for bit 0 input delay trimmer 1 - SW programmed value is applied to the pads. 2 - Trimmer value from HW calibration state machine is applied to the pads
5:0	0x0	**DPHY_INADJ_IO0_A**: Programmable value for bit 0 input delay trimmer,

NVCSI_PHY_1_NVCSI_CIL_A_CLK_DESKEW_CTRL_0

Offset: 0x872c
Byte Offset: 0x21cb0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000bf00 (0bxxxx,xxxx,xxxx,xxxx,1x11,1111,xx00,0000)

Bit	Reset	Description
15	ENABLE	**CLK_INADJ_SWEEP_CTRL_A**: Enable the clock trimmer sweep for D-PHY de-skew. **0** = DISABLE **1** = ENABLE
13:8	0x3f	**CLK_INADJ_LIMIT_HIGH_A**: The upper limit of the sweep range for trimmer sweep.
5:0	0x0	**CLK_INADJ_LIMIT_LOW_A**: The lower limit of the sweep range for trimmer sweep.

NVCSI_PHY_1_NVCSI_CIL_A_DATA_DESKEW_CTRL_0

Offset: 0x872d
Byte Offset: 0x21cb4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x9f80bf00 (0b1xx1,1111,1x00,0000,1x11,1111,xx00,0000)

Bit	Reset	Description
31	0x1	**DATA_INADJ_SWEEP_CTRL1_A**: Enable the data lane A1 trimmer sweep for D-PHY de-skew.
28:23	0x3f	**DATA_INADJ_LIMIT_HIGH1_A**: The upper limit of the sweep range for trimmer sweep.
21:16	0x0	**DATA_INADJ_LIMIT_LOW1_A**: The lower limit of the sweep range for trimmer sweep.
15	0x1	**DATA_INADJ_SWEEP_CTRL0_A**: Enable the data lane A0 trimmer sweep for D-PHY de-skew.
13:8	0x3f	**DATA_INADJ_LIMIT_HIGH0_A**: The upper limit of the sweep range for trimmer sweep.
5:0	0x0	**DATA_INADJ_LIMIT_LOW0_A**: The lower limit of the sweep range for trimmer sweep.

NVCSI_PHY_1_NVCSI_CIL_A_DPHY_DESKEW_STATUS_0

Offset: 0x872e
Byte Offset: 0x21cb8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000X0XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
15	X	**DPHY_CALIB_ERR_IO1_A**: calib error status
14	X	**DPHY_CALIB_DONE_IO1_A**: calib done status
7	X	**DPHY_CALIB_ERR_IO0_A**: calib error status

Bit	Reset	Description
6	X	**DPHY_CALIB_DONE_IO0_A**: calib done status
1	X	**DPHY_CALIB_ERR_CTRL_A**: calib error status
0	X	**DPHY_CALIB_DONE_CTRL_A**: calib done status

NVCSI_PHY_1_NVCSI_CIL_A_DPHY_DESKEW_DATA_CALIB_STATUS_LOW_0_0

Offset: 0x872f
Byte Offset: 0x21cbc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_A**

NVCSI_PHY_1_NVCSI_CIL_A_DPHY_DESKEW_DATA_CALIB_STATUS_HIGH_0_0

Offset: 0x8730
Byte Offset: 0x21cc0
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_A**

NVCSI_PHY_1_NVCSI_CIL_A_DPHY_DESKEW_DATA_CALIB_STATUS_LOW_1_0

Offset: 0x8731
Byte Offset: 0x21cc4
Read/Write: RO
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_A**

NVCSI_PHY_1_NVCSI_CIL_A_DPHY_DESKEW_DATA_CALIB_STATUS_HIGH_1_0

Offset: 0x8732
Byte Offset: 0x21cc8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_A**

NVCSI_PHY_1_NVCSI_CIL_A_DPHY_DESKEW_CLK_CALIB_STATUS_LOW_0_0

Offset: 0x8733
Byte Offset: 0x21ccc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_A**

NVCSI_PHY_1_NVCSI_CIL_A_DPHY_DESKEW_CLK_CALIB_STATUS_HIGH_0_0

Offset: 0x8734
Byte Offset: 0x21cd0
Read/Write: RO
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_A**

NVCSI_PHY_1_NVCSI_CIL_A_DPHY_DESKEW_CLK_CALIB_STATUS_LOW_1_0

Offset: 0x8735
Byte Offset: 0x21cd4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_A**

NVCSI_PHY_1_NVCSI_CIL_A_DPHY_DESKEW_CLK_CALIB_STATUS_HIGH_1_0

Offset: 0x8736
Byte Offset: 0x21cd8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_A**

NVCSI_PHY_1_NVCSI_CIL_A_DPHY_DESKEW_RESULT_STATUS_0

Offset: 0x8737
Byte Offset: 0x21cdc
Read/Write: R/W
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	DONE	**DESKEW_RESULT_STATUS1_A**: For lane A1. The result status of the deskew calibration, HW check this field before start the trimmer sweep, if it's 1'b1, the trimmer sweep will not perform. HW set this bit to 1 when the calibration is done. SW should clear this field after it read out the 2 64 bits calibration status. **0** = DONE **1** = PENDING
0	DONE	**DESKEW_RESULT_STATUS0_A**: For lane A0. The result status of the deskew calibration, HW check this field before start the trimmer sweep, if it's 1'b1, the trimmer sweep will not perform. HW set this bit to 1 when the calibration is done. SW should clear this field after it read out the 2 64 bits calibration status. **0** = DONE **1** = PENDING

NVCSI_PHY_1_NVCSI_CIL_A_POLARITY_SWIZZLE_CTRL_0

Offset: 0x8738
Byte Offset: 0x21ce0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xx00,0000,xxxx,x000)

Bit	Reset	Description
13:11	0x0	**POLARITY_SWIZZLE_CPHY1_A**: Polarity Swizle control for Lane A1 in CPHY mode. valid only in CPHY mode 000 - A B C --> A B C 001 - A B C --> A C B 010 - A B C --> B C A 011 - A B C --> B A C 100 - A B C --> C A B 101 - A B C --> C B A
10:8	0x0	**POLARITY_SWIZZLE_CPHY0_A**: Polarity Swizle control for Lane A0 in CPHY mode. valid only in CPHY mode 000 - A B C --> A B C 001 - A B C --> A C B 010 - A B C --> B C A 011 - A B C --> B A C 100 - A B C --> C A B 101 - A B C --> C B A

Bit	Reset	Description
2	0x0	**POLARITY_SWIZZLE_CLK_A**: Polarity Swizzle control for clock lane A. Valid only in DPHY mode, need to work with ALLOW_FIRST_BIT_ON_CLK_NEGEDGE if enabled.
1	0x0	**POLARITY_SWIZZLE_DPHY1_A**: Polarity Swizzle control for data lane A1. Valid only in DPHY mode.
0	0x0	**POLARITY_SWIZZLE_DPHY0_A**: Polarity Swizzle control for data lane A0. Valid only in DPHY mode.

NVCSI_PHY_1_NVCSI_CIL_A_DESKEW_CONTROL_0

Offset: 0x8739
Byte Offset: 0x21ce4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00007411 (0bxxxx,xxxx,xxxx,xxxx,0111,0100,0001,0001)

Bit	Reset	Description
15:13	0x3	**DESKEW_DATA_DELAY**: How many cycles delay introduced by re-timing flops
12	0x1	**DESKEW_LANE_SKEW_CHECK_EN**: Check the skew between the lanes, this is used to handle the error case that error happen in the CSI link caused some data lanes detect the HS sync word (0xb8) while other data lanes detect the deskew sync word (0xff). If this bit is not set with the error case, the lane FIFO will overflow and reset is required. With this bit set, the FIFO will be auto flused and the FIFO overflow will not happen.
11:8	0x4	**DESKEW_LANE_SKEW_TOLERANCE**: The allowed skew between the data lane, only used in sync word search
7:4	0x1	**DESKEW_COMPARE**: Regsiter select to control the number of comparisons to be done for each trimmer setting during deskew calibration.
3:0	0x1	**DESKEW_SETTLE**: Regsiter select to control the number of byte clocks to wait before INADJ value settles. This is used during deskew calibration

NVCSI_PHY_1_NVCSI_CIL_A_CONTROL_0

Offset: 0x873a
Byte Offset: 0x21ce8
Read/Write: R/W
Parity Protection: Y

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**BYPASS_LP_FILTER**: Bypass the glitch filter.
30:24	0x0	**CLK_SETTLE1_A**: Used only in CPHY mode, Settle time for clk start when moving A1 data lane from LP to HS (LP11->LP01->LP00).
23:17	0x0	**CLK_SETTLE0_A**: Used for both CPHY/DPY mode. For DPHY: Settle time for clk lane A0 when moving from LP to HS (LP11->LP01->LP00), this setting determines how many csicil clock cycles to wait after LP00. If the settle set to 0 or 8, the real setting is 1, otherwise, the real setting is (4+settle) For CPHY: Settle time for clk start when moving A0 data lane from LP to HS (LP11->LP01->LP00).
16:9	0x0	**THS_SETTLE1_A**: settle time for A1 data lane when moving from LP to HS (LP11->LP01->LP00), this setting determines how many LP clock cycles to wait, after LP00,before starting to look at the data.
8:1	0x0	**THS_SETTLE0_A**: settle time for A0 data lane when moving from LP to HS (LP11->LP01->LP00), this setting determines how many LP clock cycles to wait, after LP00,before starting to look at the data.
0	0x0	**BYPASS_LP_SEQ**: For DPHY: The clock lane LP signals should sequence through LP11->LP01->LP00 state, to indicate to CLOCK CIL about the mode switching to HS Rx mode. In case Camera is enabled earlier than CIL , it is highly likely that camera sends this control sequence sooner than cil can detect it. Enabling this bit allows the CLOCK CIL to overlook the LP control sequence and step in HS Rx mode directly looking at LP00 only. For CPHY: it's only used for debug purpose as it's embedded clock for CPHY, clock will always presents when the data lane toggle.

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_ERR_STATUS_0

Offset: 0x873b
Byte Offset: 0x21cec
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
29:24	X	**LANE_DEMAPPER_ERR_MUX1_A**: Debug register, error detected in CPHY de-mapper mux.
23	X	**LANE_DEMAPPER_ERR_RXCTRL1_A**: Debug register, Error detected in CPHY de-mapper rx ctrl.
22:16	X	**LANE_DECODER_ERR1_A**: Debug register, Error detected in CPHY de-coder.
13:8	X	**LANE_DEMAPPER_ERR_MUX0_A**: Debug register, Error detected in CPHY de-mapper mux.
7	X	**LANE_DEMAPPER_ERR_RXCTRL0_A**: Debug register, Error detected in CPHY de-mapper rx ctrl.
6:0	X	**LANE_DECODER_ERR0_A**: Debug register, Error detected in CPHY de-coder.

NVCSI_PHY_1_NVCSI_CIL_A_ESCAPE_MODE_COMMAND_0_0

Offset: 0x873c
Byte Offset: 0x21cf0
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_COMMAND0_A**: The received command from escape mode.

NVCSI_PHY_1_NVCSI_CIL_A_ESCAPE_MODE_DATA_0_0

Offset: 0x873d
Byte Offset: 0x21cf4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_DATA0_A**: The received data from excape mode.

NVCSI_PHY_1_NVCSI_CIL_A_ESCAPE_MODE_COMMAND_1_0

Offset: 0x873e
Byte Offset: 0x21cf8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_COMMAND1_A**: The received command from escape mode.

NVCSI_PHY_1_NVCSI_CIL_A_ESCAPE_MODE_DATA_1_0

Offset: 0x873f
Byte Offset: 0x21cfc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_DATA1_A**: The received data from excape mode.

NVCSI_PHY_1_NVCSI_CIL_A_DPHY_SYNC_PATTERN_0

Offset: 0x8740
Byte Offset: 0x21d00
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01)

Bit	Reset	Description
1	0x0	**ALLOW_FIRST_BIT_ON_CLK_NEGEDGE_A**: Allow shift 1/3/5/7 bit to search the sync word. Normal case only shift 0/2/4/6 bit to search sync word.
0	0x1	**DISABLE_SB_ERR_IN_SYNC_A**: Allow one single bit error in sync word 6MSB

NVCSI_PHY_1_NVCSI_CIL_A_DPHY_DESKEW_SYNC_PATTERN_0

Offset: 0x8741
Byte Offset: 0x21d04
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**INVERT_DESKEW_PATTERN**: Invert the de-skew pattern.

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_SYNC_SEARCH_0

Offset: 0x8742
Byte Offset: 0x21d08
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01080108 (0bxxxx,xxx1,0000,1000,xxxx,xxx1,0000,1000)

Bit	Reset	Description
24	0x1	**COUNT_EN_TRIO1_A**: Check if the symbol is pre-amble before sync word.
23:16	0x8	**PERIOD_TRIO1_A**: For CPHY, how many cycles allowed for non pre-amble symbol before sync word.

Bit	Reset	Description
8	ENABLE	**COUNT_EN_TRIO0_A**: Check if the symbol is pre-amble before sync word. **0** = DISABLE **1** = ENABLE
7:0	0x8	**PERIOD_TRIO0_A**: For CPHY, how many cycles allowed for non pre-amble symbol before sync word.

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_START_0

Offset: 0x8743
Byte Offset: 0x21d0c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**EDGE_DELAY_CAL_TRIGGER_TRIO1_A**: Trigger the CPHY clock recover calibration for lane A1 by write 1'b1, when the calibration complete, HW will clear this bit to 1'b0. Write 1'b0 to this be take no effect.
0	0x0	**EDGE_DELAY_CAL_TRIGGER_TRIO0_A**: Trigger the CPHY clock recover calibration for lane A0 by write 1'b1, when the calibration complete, HW will clear this bit to 1'b0. Write 1'b0 to this be take no effect.

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_CTRL_0

Offset: 0x8744
Byte Offset: 0x21d10
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x12011210 (0b0001,0010,0000,0001,0001,0010,0001,0000)
PROD: 0x08000800 (0bxxx0,10xx,xxxx,xxxx,xxx0,10xx,xxxx,xxxx)

Bit	Reset	PROD	Description
31	0x0	_NONE_	**TRIM_CAL_E_NONSTOP_TRIO1_A**: Reserved. Trimmer calibration background monitoring control. 1: enable. The accumulator keeps updating.
30:29	0x0	_NONE_	**TRIM_CAL_GAIN_SEL_TRIO1_A**: Trimmer calibration loop gain selection [0]: gain control 0: coarse tune (counter1) gain = 1, fine tune (counter2) gain = 1/4. For example: coarse tune: 1 UP -> trim_cal_code + 1; fine tune: 4 UPs -> trim_cal_code + 1. 1: coarse tune (counter1) gain = 1/2, fine tune (counter2) gain = 1/8 [1]: reserved
28:26	0x4	0x2	**TRIM_CAL_CNT_SEL_TRIO1_A**: Trimmer calibration counter selection [0]: counter1 (coarse tune) clock cycle. 0: 64T (128UI) 1: 128T (256UI) [2:1]: counter2 (fine tune) clock cycle. 00: 32T (64UI) 01: 64T (128UI) 10: 96T (192UI) 11: 128T (256UI) For high data rate (>3.5GS/s): 64T(cnt1)+96T(cnt2)=160T(320ui) -> cnt_sel=3'b100 For low data rate (<=3.5GS/s): 128T(cnt1)+32T(cnt2)=160T(320ui) -> cnt_se=3'b001
25	0x1	_NONE_	**TRIM_CAL_CPHY_ED_LOOPMODE_TRIO1_A**: Trimmer calibration and cphy edge detector loop mode control 0: exclude trimmer delay in 2nd round of reset loop. 1: include trimmer delay in 2nd round of reset loop. For low data rate.
24:16	0x1	_NONE_	**CALIB_LENGTH_TRIO1_A**: //The length of the peramble which use to do caliration. Count by nvcsi core clock. It's used to make sure the calibration is doing inside the preamble phase. If this length set to a small value, the calibration may can't complete within one packet and need several packets to complete.
15	0x0	_NONE_	**TRIM_CAL_E_NONSTOP_TRIO0_A**: Reserved. Trimmer calibration background monitoring control. 1: enable. The accumulator keeps updating.
14:13	0x0	_NONE_	**TRIM_CAL_GAIN_SEL_TRIO0_A**: Trimmer calibration loop gain selection [0]: gain control 0: coarse tune (counter1) gain = 1, fine tune (counter2) gain = 1/4. For example: coarse tune: 1 UP -> trim_cal_code + 1; fine tune: 4 UPs -> trim_cal_code + 1. 1: coarse tune (counter1) gain = 1/2, fine tune (counter2) gain = 1/8 [1]: reserved

Bit	Reset	PROD	Description
12:10	0x4	0x2	**TRIM_CAL_CNT_SEL_TRIO0_A**: Trimmer calibration counter selection [0]: counter1 (coarse tune) clock cycle. 0: 64T (128UI) 1: 128T (256UI) [2:1]: counter2 (fine tune) clock cycle. 00: 32T (64UI) 01: 64T (128UI) 10: 96T (192UI) 11: 128T (256UI) For high data rate (>3.5GS/s): 64T(cnt1)+96T(cnt2)=160T(320ui) -> cnt_sel=3'b100 For low data rate (<=3.5GS/s): 128T(cnt1)+32T(cnt2)=160T(320ui) -> cnt_se=3'b001
9	0x1	_NONE_	**TRIM_CAL_CPHY_ED_LOOPMODE_TRIO0_A**: Trimmer calibration and cphy edge detector loop mode control 0: exclude trimmer delay in 2nd round of reset loop. 1: include trimmer delay in 2nd round of reset loop. For low data rate.
8:0	0x10	_NONE_	**CALIB_LENGTH_TRIO0_A**: The length of the peramble which use to do caliration. Count by nvcsi core clock. It's used to make sure the calibration is doing inside the preamble phase. If this length set to a small value, the calibration may can't complete within one packet and need several packets to complete.

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_CTRL1_0

Offset: 0x8745
Byte Offset: 0x21d14
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0c0e0c0e (0bx000,1100,0000,1110,x000,1100,0000,1110)

Bit	Reset	Description
30	0x0	**PERIODIC_CALIB_TRIO1_A**: Enable the periodic calibration.
29	0x0	**CALIB_APPLY_IN_POST_TRIO1_A**: 1: Apply the calibration result when detect post symbol. 0: Apply the calibration result immediately when calibration done.

Bit	Reset	Description
28:27	0x1	**CALIB_APPLY_SELECT_TRIO1_A**: Which calibration result is applied to pad for dual calibration. 2'b11: Reserved 2'b10: Apply the middle value of two calibration to pad if the difference within threshold. 2'b01: Apply the 2nd calibration result to pad if it's difference with previous latch result within threshold. 2'b00: Apply the 1st calibration result to pad if it's difference with previous latch result within threshold.
26	0x1	**DUAL_CALIB_TRIO1_A**: Two times calibratiton. 1: Two time calibration with one trigger. 0: Single time calibration with one trigger
25:18	0x3	**TEMP_EDGE_DELAY_VALUE_TRIO1_A**: The edge delay value while doing the calibration.
17	0x1	**TEMP_EDGE_DELAY_SEL_TRIO1_A**: When the selection of edge delay value when do edge delay calibration. 1: Use the temp value from register setting. 0: Use the currently using setting. For the first calibration after reset, it's still using the temp value from register setting.
16	0x0	**BYPASS_CALIB_PACKET_TRIO1_A**: 1: Drop the packet data when do the calibration. 0: Not drop the packet data when do the calibration.
14	0x0	**PERIODIC_CALIB_TRIO0_A**: Enable the periodic calibration.
13	0x0	**CALIB_APPLY_IN_POST_TRIO0_A**: 1: Apply the calibration result when detect post symbol. 0: Apply the calibration result immediately when calibration done.
12:11	0x1	**CALIB_APPLY_SELECT_TRIO0_A**: Which calibration result is applied to pad for dual calibration. 2'b11: Reserved 2'b10: Apply the middle value of two calibration to pad if the difference within threshold. 2'b01: Apply the 2nd calibration result to pad if it's difference with previous latch result within threshold. 2'b00: Apply the 1st calibration result to pad if it's difference with previous latch result within threshold.
10	0x1	**DUAL_CALIB_TRIO0_A**: Two times calibratiton. 1: Two time calibration with one trigger. 0: Single time calibration with one trigger
9:2	0x3	**TEMP_EDGE_DELAY_VALUE_TRIO0_A**: The edge delay value apply to pad while doing the calibration.
1	0x1	**TEMP_EDGE_DELAY_SEL_TRIO0_A**: When the selection of edge delay value when do edge delay calibration. 1: Use the temp value from register setting. 0: Use the currently using setting. For the first calibration after reset, it's still using the temp value from register setting.
0	0x0	**BYPASS_CALIB_PACKET_TRIO0_A**: 1: Drop the packet data when do the calibration. 0: Not drop the packet data when do the calibration.

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_CTRL2_0

Offset: 0x8746
Byte Offset: 0x21d18
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01000100 (0bxxxx,xxx1,0000,0000,xxxx,xxx1,0000,0000)

Bit	Reset	Description
24	ENABLE	**TRIM_CAL_PD_HOLD_TRIO1_A**: Hold in PD **0** = DISABLE **1** = ENABLE
23	DISABLE	**TRIM_CAL_E_FIXEDDLY_TRIO1_A**: Enable fixed delay in trim calibration unit. **0** = DISABLE **1** = ENABLE
22:16	0x0	**TRIM_CAL_FIXEDDLY_TRIO1_A**: Adjust fixed delay. Delay = code*5ps(TT)/7ps(SS)
8	ENABLE	**TRIM_CAL_PD_HOLD_TRIO0_A**: Hold in PD **0** = DISABLE **1** = ENABLE
7	DISABLE	**TRIM_CAL_E_FIXEDDLY_TRIO0_A**: Enable fixed delay in trim calibration unit. **0** = DISABLE **1** = ENABLE
6:0	0x0	**TRIM_CAL_FIXEDDLY_TRIO0_A**: Adjust fixed delay. Delay = code*5ps(TT)/7ps(SS)

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_CTRL3_0

Offset: 0x8747
Byte Offset: 0x21d1c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffffffff	**CALIB_PERIOD_TRIO0_A**: The period of the periodic edge delay calibration, in nvcsi clock cycle number.

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_CTRL4_0

Offset: 0x8748
Byte Offset: 0x21d20
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffffffff	**CALIB_PERIOD_TRIO1_A**: The period of the periodic edge delay calibration, in nvcsi clock cycle number.

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_CTRL5_0

Offset: 0x8749
Byte Offset: 0x21d24
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000ffff (0bxxxx,xxxx,xxxx,xxxx,1111,1111,1111,1111)

Bit	Reset	Description
15:8	0xff	**ERR_THRESHOLD_TRIO1_A**: The threshold of drop the calibration result between current result and previous result. For dual calibration, check between the two calibration result in the same trigger. For signal calibration, check between the current calibration result and previous calibration result.
7:0	0xff	**ERR_THRESHOLD_TRIO0_A**: The threshold of drop the calibration result between current result and previous resulti. For dual calibration, check between the two calibration result in the same trigger. For signal calibration, check between the current calibration result and previous calibration result.

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_STATUS_0

Offset: 0x874a
Byte Offset: 0x21d28
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,xxxx)

Bit	Reset	Description
9	0x0	**EDGE_DELAY_MISMATCH_TRIO1_A**: When the CALIB_APPLY_SELECT_TRIO is 0/1 or dual time calibration is not enabled, the calibration result difference from the previous latched calibration result is larger than threshold, this time calibration result is not apply to pad. When the CALIB_APPLY_SELECT_TRIO is 2, the calibration result difference of the two times calibration result is larger than threshold, this time calibration result is not apply to pad.
8	0x0	**EDGE_DELAY_MISMATCH_TRIO0_A**: When the CALIB_APPLY_SELECT_TRIO is 0/1 or dual time calibration is not enabled, the calibration result difference from the previous latched calibration result is larger than threshold, this time calibration result is not apply to pad. When the CALIB_APPLY_SELECT_TRIO is 2, the calibration result difference of the two times calibration result is larger than threshold, this time calibration result is not apply to pad.
7	0x0	**CALIB_DONE_1_TRIO1_A**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.
6	0x0	**CALIB_DONE_1_TRIO0_A**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.
5	0x0	**CALIB_DONE_0_TRIO1_A**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.
4	0x0	**CALIB_DONE_0_TRIO0_A**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_EDGE_DELAY_CTRL_0

Offset: 0x874b
Byte Offset: 0x21d2c
Read/Write: R/W
Parity Protection: See table below
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x40000000 (0b01xx,x000,0000,0000,xxxx,x000,0000,0000)

Bit	Parity Protection	Reset	Description
31:30	Y	0x1	**CPHY_CLKGEN_TRIGWIDTH_A**: Adjust AB/CB/CA pulse width in self-clock generator.
26	N	0x0	**UPDATE_OFFSET_TRIO1_A**: Trigger to apply the new offset value
25:17	Y	0x0	**DELAY_OFFSET_TRIO1_A**: Offset value of trio A1. Represents signed 2's complement number for calibration offset. Represents unsigned number if the field used as override value.
16	Y	0x0	**OVERRIDE_CAL_VAL_TRIO1_A**: Override for trio A1 0: Apply the DELAY_OFFSET as an offset to the calibration result. 1: Apply the DELAY_OFFSET value to the pad.
10	N	0x0	**UPDATE_OFFSET_TRIO0_A**: Trigger to apply the new offset value
9:1	Y	0x0	**DELAY_OFFSET_TRIO0_A**: Offset value of trio A0. Represents signed 2's complement number for calibration offset. Represents unsigned number if the field used as override value.
0	Y	0x0	**OVERRIDE_CAL_VAL_TRIO0_A**: Override for trio A0 0: Apply the DELAY_OFFSET as an offset to the calibration result. 1: Apply the DELAY_OFFSET value to the pad.

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_EDGE_DELAY_VALUE_0

Offset: 0x874c
Byte Offset: 0x21d30
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000303 (0bxxxx,xxxx,xxxx,xxxx,0000,0011,0000,0011)

Bit	Reset	Description
15:8	0x3	**EDGE_DELAY_VALUE_TRIO1_A**: The edge delay value apply to pad finally
7:0	0x3	**EDGE_DELAY_VALUE_TRIO0_A**: The edge delay value apply to pad finally

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_RESULT_0

Offset: 0x874d
Byte Offset: 0x21d34
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**EDGE_DELAY_CALIB_RESULT_1_TRIO1_A**: The edge delay calibrate result, with offset.
23:16	0x0	**EDGE_DELAY_CALIB_RESULT_1_TRIO0_A**: The edge delay calibrate result, with offset.
15:8	0x0	**EDGE_DELAY_CALIB_RESULT_0_TRIO1_A**: The edge delay calibrate result, with offset.
7:0	0x0	**EDGE_DELAY_CALIB_RESULT_0_TRIO0_A**: The edge delay calibrate result, with offset.

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_INADJ_CALIB_CTRL_0

Offset: 0x874e
Byte Offset: 0x21d38
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
16	0x0	**INADJ_CAL_TRIGGER_TRIO1_A**: Trigger INADJ calibration for trio0
0	0x0	**INADJ_CAL_TRIGGER_TRIO0_A**: Trigger INADJ calibration for trio1

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_INADJ_CALIB_CTRL1_0

Offset: 0x874f
Byte Offset: 0x21d3c

Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00010001 (0b0000,0000,0000,0001,0000,0000,0000,0001)

Bit	Reset	Description
31:16	0x1	**PRBS9_SEED_TRIO1_A**: PRBS9 initial seed for trio0
15:0	0x1	**PRBS9_SEED_TRIO0_A**: PRBS9 initial seed for trio1

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_INADJ_CALIB_CTRL2_0

Offset: 0x8750
Byte Offset: 0x21d40
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0fc00fc0 (0bxxxx,1111,1100,0000,xxxx,1111,1100,0000)

Bit	Reset	Description
27:22	0x3f	**INADJ_LIMIT_HIGH_TRIO1_A**: The upper limit for the INADJ sweep.
21:16	0x0	**INADJ_LIMIT_LOW_TRIO1_A**: The lower limit for the INADJ sweep.
11:6	0x3f	**INADJ_LIMIT_HIGH_TRIO0_A**: The upper limit for the INADJ sweep.
5:0	0x0	**INADJ_LIMIT_LOW_TRIO0_A**: The lower limit for the INADJ sweep.

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_INADJ_CALIB_STATUS_0

Offset: 0x8751
Byte Offset: 0x21d44
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**INADJ_CALIB_DONE_TRIO1_A**: Trio a1 inadj calibration done.
0	0x0	**INADJ_CALIB_DONE_TRIO0_A**: Trio a0 inadj calibration done.

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_INADJ_CALIB_STATUS1_0

Offset: 0x8752
Byte Offset: 0x21d48
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_LOW_TRIO0_A**: Trio a0 PRBS check status for lower 32 INADJ value.

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_INADJ_CALIB_STATUS2_0

Offset: 0x8753
Byte Offset: 0x21d4c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_LOW_TRIO1_A**: Trio a1 PRBS check status for lower 32 INADJ value.

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_INADJ_CALIB_STATUS3_0

Offset: 0x8754
Byte Offset: 0x21d50

Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_HIGH_TRIO0_A**: Trio a0 PRBS check status for upper 32 INADJ value.

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_INADJ_CALIB_STATUS4_0

Offset: 0x8755
Byte Offset: 0x21d54
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_HIGH_TRIO1_A**: Trio a1 PRBS check status for upper 32 INADJ value.

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_INADJ_OVERRIDE_0

Offset: 0x8756
Byte Offset: 0x21d58
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00080008 (0bxxxx,xxx0,0000,1000,xxxx,xxx0,0000,1000)

Bit	Reset	Description
24:17	0x4	**CPHY_INADJ_TRIO1_A**: The SW override INADJ value.
16	DISABLE	**SW_SET_CPHY_INADJ_TRIO1_A**: Enable SW override of the INADJ. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
8:1	0x4	**CPHY_INADJ_TRIO0_A**: The SW override INADJ value.
0	DISABLE	**SW_SET_CPHY_INADJ_TRIO0_A**: Enable SW override of the INADJ. **0** = DISABLE **1** = ENABLE

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_BIST_CTRL_0

Offset: 0x8757
Byte Offset: 0x21d5c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	0x0	**STOP_BIST_TRIO1_A**: Manually stop the A1 BIST, write 1 to stop the RX bist.
2	0x0	**STOP_BIST_TRIO0_A**: Manually stop the A0 BIST, write 1 to stop the RX bist.
1	0x0	**TRIGGER_BIST_TRIO1_A**: Trigger the trio A1 BIST, write 1 to trigger the RX bist, HW will clear it to 0 when the BIST is done, write 0 to this field is with no effect.
0	0x0	**TRIGGER_BIST_TRIO0_A**: Trigger the trio A0 BIST, write 1 to trigger the RX bist, HW will clear it to 0 when the BIST is done, write 0 to this field is with no effect.

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_BIST_STATUS_0_0

Offset: 0x8758
Byte Offset: 0x21d60
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:8	0x0	**WORD_ERR_CNT_TRIO1_A**: The trio A1 number of word comparison failure, report for BIST mode 1.
7:0	0x0	**WORD_ERR_CNT_TRIO0_A**: The trio A0 number of word comparison failure, report for BIST mode 1.

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_BIST_STATUS_1_0

Offset: 0x8759
Byte Offset: 0x21d64
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:8	0x0	**SYM_ERR_CNT_TRIO1_A**: The trio A1 number of symbol compariosn failure, it's the sum of the symbol err count and the clock shift count, the real wire state error number is 1/2 of this field, report for BIST mode 1.
7:0	0x0	**SYM_ERR_CNT_TRIO0_A**: The trio A0 number of symbol compariosn failure, it's the sum of the symbol err count and the clock shift count, the real wire state error number is 1/2 of this field, report for BIST mode 1.

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_BIST_STATUS_2_0

Offset: 0x875a
Byte Offset: 0x21d68
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WORD_CNT_LOW_TRIO0_A**: The trio A0 low 32 bits of the compared word number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_BIST_STATUS_3_0

Offset: 0x875b
Byte Offset: 0x21d6c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WORD_CNT_HIGH_TRIO0_A**: The trio A0 high 16 bits of the compared word number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_BIST_STATUS_4_0

Offset: 0x875c
Byte Offset: 0x21d70
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WORD_CNT_LOW_TRIO1_A**: The trio A1 low 32 bits of the compared word number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_BIST_STATUS_5_0

Offset: 0x875d
Byte Offset: 0x21d74
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WORD_CNT_HIGH_TRIO1_A**: The trio A1 high 16 bits of the compared word number number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_1_NVCSI_CIL_A_CPHY_BIST_STATUS_6_0

Offset: 0x875e
Byte Offset: 0x21d78
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,x000,0000,0000,xxxx,x000,0000,0000)

Bit	Reset	Description
26	0x0	**ERR_DETECT_FAIL_TRIO1_A**: Multi symbol error in one word (7 symbol) for trio A1, BIST logic is not able to detect the accurate wire state error. With this error, if not other errors detected, the word error is accurate.
25	0x0	**SHIFT_DETECT_FAIL_TRIO1_A**: Detect symbol error in two continuous word (14 symbols) for trio A1, BIST logic is not able to determine if symbol clock is slip. (if the symbol error count is not saturating, the result is still available with this error)
24	0x0	**SHIFT_OVERFLOW_TRIO1_A**: Too many clocks slip and the BIST logic is not able to re-align for trio A1. Typically, when this error is detect, the symbol error counter will saturate.
23:20	0x0	**DOUBLE_CLK_MISSED_CNT_TRIO1_A**: Number of the times for double symbol clock lost for trio A1.
19:16	0x0	**SINGLE_CLK_MISSED_CNT_TRIO1_A**: Number of the times for single symbol clock lost for trio A1.
10	0x0	**ERR_DETECT_FAIL_TRIO0_A**: Multi symbol error in one word (7 symbol) for trio A0, BIST logic is not able to detect the accurate wire state error. With this error, if not other errors detected, the word error is accurate.
9	0x0	**SHIFT_DETECT_FAIL_TRIO0_A**: Detect symbol error in two continuous word (14 symbols) for trio A0, BIST logic is not able to determine if symbol clock is slip. (if the symbol error count is not saturating, the result is still available with this error)
8	0x0	**SHIFT_OVERFLOW_TRIO0_A**: Too many clocks slip and the BIST logic is not able to re-align for trio A0. Typically, when this error is detect, the symbol error counter will saturate.
7:4	0x0	**DOUBLE_CLK_MISSED_CNT_TRIO0_A**: Number of the times for double symbol clock lost for trio A0.

Bit	Reset	Description
3:0	0x0	**SINGLE_CLK_MISSED_CNT_TRIO0_A**: Number of the times for single symbol clock lost for trio A0.

NVCSI_PHY_1_NVCSI_CIL_B_SW_RESET_0

Offset: 0x875f
Byte Offset: 0x21d7c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**SW_RESET1_B**: SOFT RESET FOR LANE B1. Active High **1** = ENABLE **0** = DISABLE
0	0x0	**SW_RESET0_B**: SOFT RESET FOR LANE B0. Active High **1** = ENABLE **0** = DISABLE

NVCSI_PHY_1_NVCSI_CIL_B_CLKEN_OVERRIDE_CTRL_0

Offset: 0x8760
Byte Offset: 0x21d80
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**CLKEN_OVERRIDE1_B**: CLKENABLE OVERRIDE FOR LANE B1. Active High **1** = ENABLE **0** = DISABLE

Bit	Reset	Description
0	0x0	**CLKEN_OVERRIDE0_B**: CLKENABLE OVERRIDE FOR LANE B0. Active High **1** = ENABLE **0** = DISABLE

NVCSI_PHY_1_NVCSI_CIL_B_CTLE_CTRL_0

Offset: 0x8761
Byte Offset: 0x21d84
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000888 (0bxxxx,xxxx,0000,0000,0000,1000,1000,1000)

Bit	Reset	Description
23:20	0x0	**AFE_DCGAIN_IO1_B**: RX AFE DC gain control
19:16	0x0	**AFE_DCGAIN_IO0_B**: RX AFE DC gain control
15:12	0x0	**AFE_DCGAIN_CLK_B**: RX AFE DC gain control
11:8	0x8	**AFE_HFGAIN_IO1_B**: RX AFE_CTLE high frequency gain control, 16 curves, ac_gain=code*1db
7:4	0x8	**AFE_HFGAIN_IO0_B**: RX AFE_CTLE high frequency gain control, 16 curves, ac_gain=code*1db
3:0	0x8	**AFE_HFGAIN_CLK_B**: RX AFE_CTLE high frequency gain control, 16 curves, ac_gain=code*1db

NVCSI_PHY_1_NVCSI_CIL_B_CTLE_CTRL1_0

Offset: 0x8762
Byte Offset: 0x21d88
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000554 (0bxxxx,xxxx,0000,0000,0000,0101,0101,0100)

Bit	Reset	Description
23:20	0x0	**AFE_CTRL_IO1_B**
19:16	0x0	**AFE_CTRL_IO0_B**
15:12	0x0	**AFE_CTRL_CLK_B**
11:10	0x1	**AFE_CURRENT_IO1_B**
9:8	0x1	**AFE_CURRENT_IO0_B**
7:6	0x1	**AFE_CURRENT_CLK_B**
5:4	0x1	**AFE_FREQBAND_IO1_B**: RX AFE bandwidth selection. 2'b00: <1.5GS/s, 2'b01: 1.5~3GS/s, 2'b10: 3~5GS/s, 2'b11: reserved.
3:2	0x1	**AFE_FREQBAND_IO0_B**: RX AFE bandwidth selection. 2'b00: <1.5GS/s, 2'b01: 1.5~3GS/s, 2'b10: 3~5GS/s, 2'b11: reserved.
1:0	0x0	**AFE_FREQBAND_CLK_B**: RX AFE bandwidth selection. 2'b00: <1.5GS/s, 2'b01: 1.5~3GS/s, 2'b10: 3~5GS/s, 2'b11: reserved.

NVCSI_PHY_1_NVCSI_CIL_B_PAD_CONFIG_0

Offset: 0x8763
Byte Offset: 0x21d8c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00070000 (0b0000,0000,0000,x111,xxxx,0000,0000,0000)

Bit	Reset	Description
31	0x0	**LP_RX_SELECT_IO1_B**: LP mode receiver select for partition B lane 1. DPHY mode: controls IO1P_B and IO1N_B CPHY mode: controls IO1P_B, IO1N_B and IOCLKN_B.
30	0x0	**LP_RX_SELECT_IO0_B**: LP mode receiver select for partition B lane 0. DPHY mode: controls IO0P_B and IO0N_B CPHY mode: controls IO0P_B, IO0N_B and IOCLKP_B.
29	0x0	**LP_RX_SELECT_CLK_B**: LP mode receiver select. 0: Schmitt receiver. 1: self-biased receiver. DPHY mode: controls IOCLKP_B and IOCLKN_B. CPHY mode: dummy.

Bit	Reset	Description
28	0x0	**E_LPRX_HYS_N_IO1_B**: Enable LP-RX hysteresis. 0: enable hysteresis 1: disable hysteresis
27	0x0	**E_LPRX_HYS_N_IO0_B**: Enable LP-RX hysteresis. 0: enable hysteresis 1: disable hysteresis
26	0x0	**E_LPRX_HYS_N_CLK_B**: Enable LP-RX hysteresis. 0: enable hysteresis 1: disable hysteresis
25	0x0	**E_HS_TERM_N_IO1_B**: Enable the HS termination of Partition B Lane 1.
24	0x0	**E_HS_TERM_N_IO0_B**: Enable the HS termination of Partition B Lane 0.
23	0x0	**E_HS_TERM_N_CLK_B**: Enable the HS termination of Clock partition B.
22	0x0	**E_INPUT_LP_IO1_B**: Enable LP receiver of Partition B Lane 1 Applicable in both CPHY and DPHY case.
21	0x0	**E_INPUT_LP_IO0_B**: Enable LP receiver of Partition B Lane 0 Applicable in both CPHY and DPHY case.
20	0x0	**E_INPUT_LP_CLK_B**: enable LP receiver of Clock partition B Applicable in DPHY case. N/A for CPHY
18	0x1	**PD_CLK_B**: Power down for CLk of Partition B. Applicable in DPHY case. N/A for CPHY
17	0x1	**PD_IO1_B**: Power down for Trio 1 and Lane 1 of Partition B. Applicable in both CPHY and DPHY case.
16	0x1	**PD_IO0_B**: Power down for Trio 0 and Lane 0 of Partition B. Applicable in both CPHY and DPHY case.
11:8	0x0	**SPARE_CLK_B**: Spare control bits for pad control. Functionality of other bits currently not defined. Undefined register bits are reserved for future use. Pull down may be enabled when Clock Lane is unused to minimized interference.
7:4	0x0	**SPARE_IO1_B**: Spare control bits for pad control. Functionality of other bits currently not defined. Undefined register bits are reserved for future use. Pull down may be enabled when Data Lane-1 is unused to minimized interference.

Bit	Reset	Description
3:0	0x0	**SPARE_IO0_B**: Spare control bits for pad control. Functionality of other bits currently not defined. Undefined register bits are reserved for future use. Pull down may be enabled when Data Lane-0 is unused to minimized interference.

NVCSI_PHY_1_NVCSI_CIL_B_PAD_CONFIG_1_0

Offset: 0x8764
Byte Offset: 0x21d90
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000002 (0b0000,0000,0000,0000,0000,0000,xx00,0010)

Bit	Reset	Description
31:24	0x0	**OUTADJ_IO1_B**: Trio 1 output delay trimmer, each tap delays 9ps
23:16	0x0	**OUTADJ_IO0_B**: Trio 0 output delay trimmer, each tap delays 9ps
15:8	0x0	**OUTADJ_CLK_B**: Clock bit output delay trimmer, each tap delays 9ps
5	0x0	**HS_BSO_B**: 1= power-on 0.4v-TX-regulator during LP-mode; 0= power-off 0.4v-TX-regulator during LP-mode;
4	0x0	**CPHY_MID_STRENGTH_TRIO1_B**: 0: when bump is driven to mid-voltage-level, both 100-ohm-pull-up and 100-ohm-pull-down is turned on to drive mid-voltage-level; 1: when bump is driven to mid-voltage-level, both pull-up and pull-down are turned off to become tri-stated;
3	0x0	**CPHY_MID_STRENGTH_TRIO0_B**: 0: when bump is driven to mid-voltage-level, both 100-ohm-pull-up and 100-ohm-pull-down is turned on to drive mid-voltage-level; 1: when bump is driven to mid-voltage-level, both pull-up and pull-down are turned off to become tri-stated;
2	DISABLE	**REV_CLK_B**: Reverse clock polarity **0** = DISABLE **1** = ENABLE
1	ENABLE	**E_PREDRV_FLOP_RESET_B**: HS predriver flop reset **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
0	DISABLE	**FCZERO_B**: Force clk bit to drive differential 0 **0** = DISABLE **1** = ENABLE

NVCSI_PHY_1_NVCSI_CIL_B_PAD_CONFIG_2_0

Offset: 0x8765
Byte Offset: 0x21d94
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,0000,0000,0000,xxxx,0000,0000,0000)

Bit	Reset	Description
27:24	0x0	**PEMPD_IO1_B**: Enable trio 1 HS driver pull down pre-emphasis.
23:20	0x0	**PEMPD_IO0_B**: Enable trio 0 HS driver pull down pre-emphasis.
19:16	0x0	**PEMPD_CLK_B**: Enable clock bit HS driver pull down pre-emphasis.
11:8	0x0	**PEMPU_IO1_B**: Enable trio 1 HS driver pull up pre-emphasis.
7:4	0x0	**PEMPU_IO0_B**: Enable trio 0 HS driver pull up pre-emphasis.
3:0	0x0	**PEMPU_CLK_B**: Enable clock bit HS driver pull up pre-emphasis.

NVCSI_PHY_1_NVCSI_CIL_B_PAD_CONFIG_3_0

Offset: 0x8766
Byte Offset: 0x21d98
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x06660666 (0bxxxx,0110,0110,0110,xxxx,0110,0110,0110)

Bit	Reset	Description
27:24	0x6	**LPDNADJ_IO1_B**: Driver pull down impedance control
23:20	0x6	**LPDNADJ_IO0_B**: Driver pull down impedance control
19:16	0x6	**LPDNADJ_CLK_B**: Driver pull down impedance control
11:8	0x6	**LPUPADJ_IO1_B**: Driver pull up impedance control
7:4	0x6	**LPUPADJ_IO0_B**: Driver pull up impedance control
3:0	0x6	**LPUPADJ_CLK_B**: Driver pull up impedance control

NVCSI_PHY_1_NVCSI_CIL_B_PAD_CONFIG_4_0

Offset: 0x8767
Byte Offset: 0x21d9c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,0000,0000,0000,xxxx,0000,0000,0000)

Bit	Reset	Description
27:24	0x0	**SLEWDNADJ_IO1_B**: Pull down slew rate adjust
23:20	0x0	**SLEWDNADJ_IO0_B**: Pull down slew rate adjust
19:16	0x0	**SLEWDNADJ_CLK_B**: Pull down slew rate adjust
11:8	0x0	**SLEWUPADJ_IO1_B**: Pull up slew rate adjust
7:4	0x0	**SLEWUPADJ_IO0_B**: Pull up slew rate adjust
3:0	0x0	**SLEWUPADJ_CLK_B**: Pull up slew rate adjust

NVCSI_PHY_1_NVCSI_CIL_B_PAD_CONFIG_5_0

Offset: 0x8768

Byte Offset: 0x21da0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000110 (0bxxxx,xxxx,xxxx,xxxx,xxxx,x001,0001,0000)

Bit	Reset	Description
10:7	0x2	**CDDNADJ_IO1_B**: Level adjust on low limit of detection
6:3	0x2	**CDDNADJ_IO0_B**: Level adjust on low limit of detection
2	DISABLE	**E_CD_IO1_B**: Data bits contention detector enable **0** = DISABLE **1** = ENABLE
1	DISABLE	**E_CD_IO0_B**: Data bits contention detector enable **0** = DISABLE **1** = ENABLE
0	DISABLE	**E_CD_CLK_B**: Clock bit contention detector enable **0** = DISABLE **1** = ENABLE

NVCSI_PHY_1_NVCSI_CIL_B_PAD_CONFIG_6_0

Offset: 0x8769
Byte Offset: 0x21da4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**E_DEEP_LPBK_B**: Enable deep loopback
0	0x0	**E_SHALLOW_LPBK_B**: Enalbe shallow loopback

NVCSI_PHY_1_NVCSI_CIL_B_PAD_CD_STATUS_0

Offset: 0x876a
Byte Offset: 0x21da8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**CDN_IO1_B**: trio1 n output, 1=contention detected.
4	0x0	**CDN_IO0_B**: trio0 n output, 1=contention detected.
3	0x0	**CDN_CLK_B**: Clock bit n output, 1=contention detected.
2	0x0	**CDP_IO1_B**: trio1 p output, 1=contention detected.
1	0x0	**CDP_IO0_B**: trio0 p output, 1=contention detected.
0	0x0	**CDP_CLK_B**: Clock bit p output, 1=contention detected.

NVCSI_PHY_1_NVCSI_CIL_B_DPHY_INADJ_CTRL_0

Offset: 0x876b
Byte Offset: 0x21dac
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,x000,0000,x000,0000,x000,0000)

Bit	Reset	Description
22	0x0	**SW_SET_DPHY_INADJ_CLK_B**: Software set for clock input delay trimmer
21:16	0x0	**DPHY_INADJ_CLK_B**: Programmable value for CLK input delay trimmer,

Bit	Reset	Description
14	0x0	**SW_SET_DPHY_INADJ_IO1_B**: Software override for bit 1 input delay trimmer 1 - SW programmed value is applied to the pads. 2 - Trimmer value from HW calibration state machine is applied to the pads
13:8	0x0	**DPHY_INADJ_IO1_B**: Programmable value for bit 1 input delay trimmer, each tap delay of 9 ps
6	0x0	**SW_SET_DPHY_INADJ_IO0_B**: Software override for bit 0 input delay trimmer 1 - SW programmed value is applied to the pads. 2 - Trimmer value from HW calibration state machine is applied to the pads
5:0	0x0	**DPHY_INADJ_IO0_B**: Programmable value for bit 0 input delay trimmer,

NVCSI_PHY_1_NVCSI_CIL_B_CLK_DESKEW_CTRL_0

Offset: 0x876c
Byte Offset: 0x21db0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000bf00 (0bxxxx,xxxx,xxxx,xxxx,1x11,1111,xx00,0000)

Bit	Reset	Description
15	ENABLE	**CLK_INADJ_SWEEP_CTRL_B**: Enable the clock trimmer sweep for D-PHY de-skew. **0** = DISABLE **1** = ENABLE
13:8	0x3f	**CLK_INADJ_LIMIT_HIGH_B**: The upper limit of the sweep range for trimmer sweep.
5:0	0x0	**CLK_INADJ_LIMIT_LOW_B**: The lower limit of the sweep range for trimmer sweep.

NVCSI_PHY_1_NVCSI_CIL_B_DATA_DESKEW_CTRL_0

Offset: 0x876d
Byte Offset: 0x21db4
Read/Write: R/W
Parity Protection: Y
Shadow: N

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x9f80bf00 (0b1xx1,1111,1x00,0000,1x11,1111,xx00,0000)

Bit	Reset	Description
31	0x1	**DATA_INADJ_SWEEP_CTRL1_B**: Enable the data lane B1 trimmer sweep for D-PHY de-skew.
28:23	0x3f	**DATA_INADJ_LIMIT_HIGH1_B**: The upper limit of the sweep range for trimmer sweep.
21:16	0x0	**DATA_INADJ_LIMIT_LOW1_B**: The lower limit of the sweep range for trimmer sweep.
15	0x1	**DATA_INADJ_SWEEP_CTRL0_B**: Enable the data lane B0 trimmer sweep for D-PHY de-skew.
13:8	0x3f	**DATA_INADJ_LIMIT_HIGH0_B**: The upper limit of the sweep range for trimmer sweep.
5:0	0x0	**DATA_INADJ_LIMIT_LOW0_B**: The lower limit of the sweep range for trimmer sweep.

NVCSI_PHY_1_NVCSI_CIL_B_DPHY_DESKEW_STATUS_0

Offset: 0x876e
Byte Offset: 0x21db8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000X0XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
15	X	**DPHY_CALIB_ERR_IO1_B**: calib error status
14	X	**DPHY_CALIB_DONE_IO1_B**: calib done status
7	X	**DPHY_CALIB_ERR_IO0_B**: calib error status
6	X	**DPHY_CALIB_DONE_IO0_B**: calib done status
1	X	**DPHY_CALIB_ERR_CTRL_B**: calib error status
0	X	**DPHY_CALIB_DONE_CTRL_B**: calib done status

NVCSI_PHY_1_NVCSI_CIL_B_DPHY_DESKEW_DATA_CALIB_STATUS_LOW_0_0

Offset: 0x876f
Byte Offset: 0x21dbc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_B**

NVCSI_PHY_1_NVCSI_CIL_B_DPHY_DESKEW_DATA_CALIB_STATUS_HIGH_0_0

Offset: 0x8770
Byte Offset: 0x21dc0
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_B**

NVCSI_PHY_1_NVCSI_CIL_B_DPHY_DESKEW_DATA_CALIB_STATUS_LOW_1_0

Offset: 0x8771
Byte Offset: 0x21dc4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_B**

NVCSI_PHY_1_NVCSI_CIL_B_DPHY_DESKEW_DATA_CALIB_STATUS_HIGH_1_0

Offset: 0x8772
Byte Offset: 0x21dc8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_B**

NVCSI_PHY_1_NVCSI_CIL_B_DPHY_DESKEW_CLK_CALIB_STATUS_LOW_0_0

Offset: 0x8773
Byte Offset: 0x21dcc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_B**

NVCSI_PHY_1_NVCSI_CIL_B_DPHY_DESKEW_CLK_CALIB_STATUS_HIGH_0_0

Offset: 0x8774
Byte Offset: 0x21dd0
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_B**

NVCSI_PHY_1_NVCSI_CIL_B_DPHY_DESKEW_CLK_CALIB_STATUS_LOW_1_0

Offset: 0x8775
Byte Offset: 0x21dd4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_B**

NVCSI_PHY_1_NVCSI_CIL_B_DPHY_DESKEW_CLK_CALIB_STATUS_HIGH_1_0

Offset: 0x8776
Byte Offset: 0x21dd8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_B**

NVCSI_PHY_1_NVCSI_CIL_B_DPHY_DESKEW_RESULT_STATUS_0

Offset: 0x8777
Byte Offset: 0x21ddc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	DONE	**DESKEW_RESULT_STATUS1_B**: For lane B1. The result status of the deskew calibration, HW check this field before start the trimmer sweep, if it's 1'b1, the trimmer sweep will not perform. HW set this bit to 1 when the calibration is done. SW should clear this field after it read out the 2 64 bits calibration status. **0** = DONE **1** = PENDING
0	DONE	**DESKEW_RESULT_STATUS0_B**: For lane B0. The result status of the deskew calibration, HW check this field before start the trimmer sweep, if it's 1'b1, the trimmer sweep will not perform. HW set this bit to 1 when the calibration is done. SW should clear this field after it read out the 2 64 bits calibration status. **0** = DONE **1** = PENDING

NVCSI_PHY_1_NVCSI_CIL_B_POLARITY_SWIZZLE_CTRL_0

Offset: 0x8778
Byte Offset: 0x21de0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xx00,0000,xxxx,x000)

Bit	Reset	Description
13:11	0x0	**POLARITY_SWIZZLE_CPHY1_B**: Polarity Swizle control for Lane B1 in CPHY mode. valid only in CPHY mode 000 - A B C --> A B C 001 - A B C --> A C B 010 - A B C --> B C A 011 - A B C --> B A C 100 - A B C --> C A B 101 - A B C --> C B A
10:8	0x0	**POLARITY_SWIZZLE_CPHY0_B**: Polarity Swizle control for Lane B0 in CPHY mode. valid only in CPHY mode 000 - A B C --> A B C 001 - A B C --> A C B 010 - A B C --> B C A 011 - A B C --> B A C 100 - A B C --> C A B 101 - A B C --> C B A
2	0x0	**POLARITY_SWIZZLE_CLK_B**: Polarity Swizzle control for clock lane B. Valid only in DPHY mode, need to work with ALLOW_FIRST_BIT_ON_CLK_NEGEDGE if enabled.

Bit	Reset	Description
1	0x0	**POLARITY_SWIZZLE_DPHY1_B**: Polarity Swizzle control for data lane B1. register bit valid only in DPHY mode.
0	0x0	**POLARITY_SWIZZLE_DPHY0_B**: Polarity Swizzle control for data lane B0. register bit valid only in DPHY mode.

NVCSI_PHY_1_NVCSI_CIL_B_DESKEW_CONTROL_0

Offset: 0x8779
Byte Offset: 0x21de4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00007411 (0bxxxx,xxxx,xxxx,xxxx,0111,0100,0001,0001)

Bit	Reset	Description
15:13	0x3	**DESKEW_DATA_DELAY**: How many cycles delay introduced by re-timing flops
12	0x1	**DESKEW_LANE_SKEW_CHECK_EN**: Check the skew between the lanes. this is used to handle the error case that error happen in the CSI link caused some data lanes detect the HS sync word (0xb8) while other data lanes detect the deskew sync word (0xff). If this bit is not set with the error case, the lane FIFO will overflow and reset is required. With this bit set, the FIFO will be auto flused and the FIFO overflow will not happen.
11:8	0x4	**DESKEW_LANE_SKEW_TOLERANCE**: The allowed skew between the data lane, only used in sync word search
7:4	0x1	**DESKEW_COMPARE**: Regsiter select to control the number of comparisons to be done for each trimmer setting during deskew calibration.
3:0	0x1	**DESKEW_SETTLE**: Regsiter select to control the number of byte clocks to wait before INADJ value settles. This is used during deskew calibration

NVCSI_PHY_1_NVCSI_CIL_B_CONTROL_0

Offset: 0x877a
Byte Offset: 0x21de8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**BYPASS_LP_FILTER**: Bypass the glitch filter.
30:24	0x0	**CLK_SETTLE1_B**: Used only in CPHY mode, Settle time for clk start when moving B1 data lane from LP to HS (LP11->LP01->LP00).
23:17	0x0	**CLK_SETTLE0_B**: Used for both CPHY/DPY mode. For DPHY: Settle time for clk lane when moving from LP to HS (LP11->LP01->LP00), this setting determines how many csicil clock cycles to wait after LP00. If the settle set to 0 or 8, the real setting is 1, otherwise, the real setting is (4+settle) For CPHY: Settle time for clk start when moving B0 data lane from LP to HS (LP11->LP01->LP00).
16:9	0x0	**THS_SETTLE1_B**: settle time for B1 data lane when moving from LP to HS (LP11->LP01->LP00), this setting determines how many LP clock cycles to wait, after LP00, before starting to look at the data.
8:1	0x0	**THS_SETTLE0_B**: settle time for B0 data lane when moving from LP to HS (LP11->LP01->LP00), this setting determines how many LP clock cycles to wait, after LP00, before starting to look at the data.
0	0x0	**BYPASS_LP_SEQ**: For DPHY: The clock lane LP signals should sequence through LP11->LP01->LP00 state, to indicate to CLOCK CIL about the mode switching to HS Rx mode. In case Camera is enabled earlier than CIL , it is highly likely that camera sends this control sequence sooner than cil can detect it. Enabling this bit allows the CLOCK CIL to overlook the LP control sequence and step in HS Rx mode directly looking at LP00 only. For CPHY: it's only used for debug purpose as it's embedded clock for CPHY, clock will always presents when the data lane toggle.

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_ERR_STATUS_0

Offset: 0x877b
Byte Offset: 0x21dec
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
29:24	X	**LANE_DEMAPPER_ERR_MUX1_B**: Debug register, error detected in CPHY de-mapper mux.
23	X	**LANE_DEMAPPER_ERR_RXCTRL1_B**: Debug register, error detected in CPHY de-mapper rx ctrl.
22:16	X	**LANE_DECODER_ERR1_B**: Debug register, error detected in CPHY de-coder.
13:8	X	**LANE_DEMAPPER_ERR_MUX0_B**: Debug register, error detected in CPHY de-mapper mux.
7	X	**LANE_DEMAPPER_ERR_RXCTRL0_B**: Debug register, error detected in CPHY de-mapper rx ctrl.
6:0	X	**LANE_DECODER_ERR0_B**: Debug register, error detected in CPHY de-coder.

NVCSI_PHY_1_NVCSI_CIL_B_ESCAPE_MODE_COMMAND_0_0

Offset: 0x877c
Byte Offset: 0x21df0
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_COMMAND0_B**: The received command from escape mode.

NVCSI_PHY_1_NVCSI_CIL_B_ESCAPE_MODE_DATA_0_0

Offset: 0x877d
Byte Offset: 0x21df4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_DATA0_B**: The received data from excape mode.

NVCSI_PHY_1_NVCSI_CIL_B_ESCAPE_MODE_COMMAND_1_0

Offset: 0x877e
Byte Offset: 0x21df8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_COMMAND1_B**: The received command from escape mode.

NVCSI_PHY_1_NVCSI_CIL_B_ESCAPE_MODE_DATA_1_0

Offset: 0x877f
Byte Offset: 0x21dfc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_DATA1_B**: The received data from excape mode.

NVCSI_PHY_1_NVCSI_CIL_B_DPHY_SYNC_PATTERN_0

Offset: 0x8780
Byte Offset: 0x21e00
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01)

Bit	Reset	Description
1	0x0	**ALLOW_FIRST_BIT_ON_CLK_NEGEDGE_B**: Allow shift 1/3/5/7 bit to search the sync word. Normal case only shift 0/2/4/6 bit to search sync word.
0	0x1	**DISABLE_SB_ERR_IN_SYNC_B**: Allow one single bit error in sync word 6MSB.

NVCSI_PHY_1_NVCSI_CIL_B_DPHY_DESKEW_SYNC_PATTERN_0

Offset: 0x8781
Byte Offset: 0x21e04
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**INVERT_DESKEW_PATTERN**: Invert the de-skew pattern.

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_SYNC_SEARCH_0

Offset: 0x8782
Byte Offset: 0x21e08
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01080108 (0bxxxx,xxx1,0000,1000,xxxx,xxx1,0000,1000)

Bit	Reset	Description
24	0x1	**COUNT_EN_TRIO1_B**: Check if the symbol is pre-amble before sync word.
23:16	0x8	**PERIOD_TRIO1_B**: For CPHY, how many cycles allowed for non pre-amble symbol before sync word.
8	0x1	**COUNT_EN_TRIO0_B**: Check if the symbol is pre-amble before sync word.

Bit	Reset	Description
7:0	0x8	**PERIOD_TRIO0_B**: For CPHY, how many cycles allowed for non pre-amble symbol before sync word.

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_START_0

Offset: 0x8783
Byte Offset: 0x21e0c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**EDGE_DELAY_CAL_TRIGGER_TRIO1_B**: Trigger the CPHY clock recover calibration for lane B1 by write 1'b1, when the calibration complete, HW will clear this bit to 1'b0. Write 1'b0 to this be take no effect.
0	0x0	**EDGE_DELAY_CAL_TRIGGER_TRIO0_B**: Trigger the CPHY clock recover calibration for lane B0 by write 1'b1, when the calibration complete, HW will clear this bit to 1'b0. Write 1'b0 to this be take no effect.

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_CTRL_0

Offset: 0x8784
Byte Offset: 0x21e10
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x12011201 (0b0001,0010,0000,0001,0001,0010,0000,0001)
PROD: 0x08000800 (0bxxx0,10xx,xxxx,xxxx,xxx0,10xx,xxxx,xxxx)

Bit	Reset	PROD	Description
31	0x0	_NONE_	**TRIM_CAL_E_NONSTOP_TRIO1_B**: Reserved. Trimmer calibration background monitoring control. 1: enable. The accumulator keeps updating.

Bit	Reset	PROD	Description
30:29	0x0	_NONE_	**TRIM_CAL_GAIN_SEL_TRIO1_B**: Trimmer calibration loop gain selection [0]: gain control 0: coarse tune (counter1) gain = 1, fine tune (counter2) gain = 1/4. For example: coarse tune: 1 UP -> trim_cal_code + 1; fine tune: 4 UPs -> trim_cal_code + 1. 1: coarse tune (counter1) gain = 1/2, fine tune (counter2) gain = 1/8 [1]: reserved
28:26	0x4	0x2	**TRIM_CAL_CNT_SEL_TRIO1_B**: Trimmer calibration counter selection [0]: counter1 (coarse tune) clock cycle. 0: 64T (128UI) 1: 128T (256UI) [2:1]: counter2 (fine tune) clock cycle. 00: 32T (64UI) 01: 64T (128UI) 10: 96T (192UI) 11: 128T (256UI) For high data rate (>3.5GS/s): 64T(cnt1)+96T(cnt2)=160T(320ui) -> cnt_sel=3'b100 For low data rate (<=3.5GS/s): 128T(cnt1)+32T(cnt2)=160T(320ui) -> cnt_se=3'b001
25	0x1	_NONE_	**TRIM_CAL_CPHY_ED_LOOPMODE_TRIO1_B**: Trimmer calibration and cphy edge detector loop mode control 0: exclude trimmer delay in 2nd round of reset loop. 1: include trimmer delay in 2nd round of reset loop. For low data rate.
24:16	0x1	_NONE_	**CALIB_LENGTH_TRIO1_B**: The length of the peramble which use to do caliration. Count by nvcsi core clock. It's used to make sure the calibration is doing inside the preamble phase. If this length set to a small value, the calibration may can't complete within one packet and need several packets to complete.
15	0x0	_NONE_	**TRIM_CAL_E_NONSTOP_TRIO0_B**: Reserved. Trimmer calibration background monitoring control. 1: enable. The accumulator keeps updating.
14:13	0x0	_NONE_	**TRIM_CAL_GAIN_SEL_TRIO0_B**: Trimmer calibration loop gain selection [0]: gain control 0: coarse tune (counter1) gain = 1, fine tune (counter2) gain = 1/4. For example: coarse tune: 1 UP -> trim_cal_code + 1; fine tune: 4 UPs -> trim_cal_code + 1. 1: coarse tune (counter1) gain = 1/2, fine tune (counter2) gain = 1/8 [1]: reserved

Bit	Reset	PROD	Description
12:10	0x4	0x2	**TRIM_CAL_CNT_SEL_TRIO0_B**: Trimmer calibration counter selection [0]: counter1 (coarse tune) clock cycle. 0: 64T (128UI) 1: 128T (256UI) [2:1]: counter2 (fine tune) clock cycle. 00: 32T (64UI) 01: 64T (128UI) 10: 96T (192UI) 11: 128T (256UI) For high data rate (>3.5GS/s): 64T(cnt1)+96T(cnt2)=160T(320ui) -> cnt_sel=3'b100 For low data rate (<=3.5GS/s): 128T(cnt1)+32T(cnt2)=160T(320ui) -> cnt_se=3'b001
9	0x1	_NONE_	**TRIM_CAL_CPHY_ED_LOOPMODE_TRIO0_B**: Trimmer calibration and cphy edge detector loop mode control 0: exclude trimmer delay in 2nd round of reset loop. 1: include trimmer delay in 2nd round of reset loop. For low data rate.
8:0	0x1	_NONE_	**CALIB_LENGTH_TRIO0_B**: The length of the peramble which use to do caliration. Count by nvcsi core clock. It's used to make sure the calibration is doing inside the preamble phase. If this length set to a small value, the calibration may can't complete within one packet and need several packets to complete.

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_CTRL1_0

Offset: 0x8785
Byte Offset: 0x21e14
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0c0e0c0e (0bx000,1100,0000,1110,x000,1100,0000,1110)

Bit	Reset	Description
30	0x0	**PERIODIC_CALIB_TRIO1_B**: Enable the periodic calibration.
29	0x0	**CALIB_APPLY_IN_POST_TRIO1_B**: 1: Apply the calibration result when detect post symbol. 0: Apply the calibration result immediately when the calibration done.

Bit	Reset	Description
28:27	0x1	**CALIB_APPLY_SELECT_TRIO1_B**: Which calibration result is applied to pad for dual calibration. 2'b11: Reserved 2'b10: Apply the middle value of two calibration to pad if the difference within threshold. 2'b01: Apply the 2nd calibration result to pad if it's difference with previous latch result within threshold. 2'b00: Apply the 1st calibration result to pad if it's difference with previous latch result within threshold.
26	0x1	**DUAL_CALIB_TRIO1_B**: Two times calibratiton. 1: Two time calibration with one trigger. 0: Single time calibration with one trigger
25:18	0x3	**TEMP_EDGE_DELAY_VALUE_TRIO1_B**: The edge delay value while doing the calibration.
17	0x1	**TEMP_EDGE_DELAY_SEL_TRIO1_B**: When the selection of edge delay value when do edge delay calibration. 1: Use the temp value from register setting. 0: Use the currently using setting. For the first calibration after reset, it's still using the temp value from register setting.
16	0x0	**BYPASS_CALIB_PACKET_TRIO1_B**: 1: Drop the packet data when do the calibration. 0: Not drop the packet data when do the calibration.
14	0x0	**PERIODIC_CALIB_TRIO0_B**: Enable the periodic calibration.
13	0x0	**CALIB_APPLY_IN_POST_TRIO0_B**: 1: Apply the calibration result when detect post symbol. 0: Apply the calibration result immediately when the calibration done.
12:11	0x1	**CALIB_APPLY_SELECT_TRIO0_B**: Which calibration result is applied to pad for dual calibration. 2'b11: Reserved 2'b10: Apply the middle value of two calibration to pad if the difference within threshold. 2'b01: Apply the 2nd calibration result to pad if it's difference with previous latch result within threshold. 2'b00: Apply the 1st calibration result to pad if it's difference with previous latch result within threshold.
10	0x1	**DUAL_CALIB_TRIO0_B**: Two times calibratiton. 1: Two time calibration with one trigger. 0: Single time calibration with one trigger
9:2	0x3	**TEMP_EDGE_DELAY_VALUE_TRIO0_B**: The edge delay value apply to pad while doing the calibration.
1	0x1	**TEMP_EDGE_DELAY_SEL_TRIO0_B**: When the selection of edge delay value when do edge delay calibration. 1: Use the temp value from register setting. 0: Use the currently using setting. For the first calibration after reset, it's still using the temp value from register setting.
0	0x0	**BYPASS_CALIB_PACKET_TRIO0_B**: 1: Drop the packet data when do the calibration. 0: Not drop the packet data when do the calibration.

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_CTRL2_0

Offset: 0x8786
Byte Offset: 0x21e18
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01000100 (0bxxxx,xxx1,0000,0000,xxxx,xxx1,0000,0000)

Bit	Reset	Description
24	ENABLE	**TRIM_CAL_PD_HOLD_TRIO1_B**: Hold in PD **0** = DISABLE **1** = ENABLE
23	DISABLE	**TRIM_CAL_E_FIXEDDLY_TRIO1_B**: Enable fixed delay in trim calibration unit. **0** = DISABLE **1** = ENABLE
22:16	0x0	**TRIM_CAL_FIXEDDLY_TRIO1_B**: Adjust fixed delay. Delay = code*5ps(TT)/7ps(SS)
8	ENABLE	**TRIM_CAL_PD_HOLD_TRIO0_B**: Hold in PD **0** = DISABLE **1** = ENABLE
7	DISABLE	**TRIM_CAL_E_FIXEDDLY_TRIO0_B**: Enable fixed delay in trim calibration unit. **0** = DISABLE **1** = ENABLE
6:0	0x0	**TRIM_CAL_FIXEDDLY_TRIO0_B**: Adjust fixed delay. Delay = code*5ps(TT)/7ps(SS)

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_CTRL3_0

Offset: 0x8787
Byte Offset: 0x21e1c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffffffff	**CALIB_PERIOD_TRIO0_B**: The period of the periodic edge delay calibration, in nvcsi clock cycle number.

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_CTRL4_0

Offset: 0x8788
Byte Offset: 0x21e20
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffffffff	**CALIB_PERIOD_TRIO1_B**: The period of the periodic edge delay calibration, in nvcsi clock cycle number.

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_CTRL5_0

Offset: 0x8789
Byte Offset: 0x21e24
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000ffff (0bxxxx,xxxx,xxxx,xxxx,1111,1111,1111,1111)

Bit	Reset	Description
15:8	0xff	**ERR_THRESHOLD_TRIO1_B**: The threshold of drop the calibration result between current result and previous result. For dual calibration, check between the two calibration result in the same trigger. For signal calibration, check between the current calibration result and previous calibration result.
7:0	0xff	**ERR_THRESHOLD_TRIO0_B**: The threshold of drop the calibration result between current result and previous result. For dual calibration, check between the two calibration result in the same trigger. For signal calibration, check between the current calibration result and previous calibration result.

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_STATUS_0

Offset: 0x878a
Byte Offset: 0x21e28
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,xxxx)

Bit	Reset	Description
9	0x0	**EDGE_DELAY_MISMATCH_TRIO1_B**: When the CALIB_APPLY_SELECT_TRIO is 0/1 or dual time calibration is not enabled, the calibration result difference from the previous latched calibration result is larger than threshold, this time calibration result is not apply to pad. When the CALIB_APPLY_SELECT_TRIO is 2, the calibration result difference of the two times calibration result is larger than threshold, this time calibration result is not apply to pad.
8	0x0	**EDGE_DELAY_MISMATCH_TRIO0_B**: When the CALIB_APPLY_SELECT_TRIO is 0/1 or dual time calibration is not enabled, the calibration result difference from the previous latched calibration result is larger than threshold, this time calibration result is not apply to pad. When the CALIB_APPLY_SELECT_TRIO is 2, the calibration result difference of the two times calibration result is larger than threshold, this time calibration result is not apply to pad.
7	0x0	**CALIB_DONE_1_TRIO1_B**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.
6	0x0	**CALIB_DONE_1_TRIO0_B**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.
5	0x0	**CALIB_DONE_0_TRIO1_B**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.
4	0x0	**CALIB_DONE_0_TRIO0_B**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_EDGE_DELAY_CTRL_0

Offset: 0x878b
Byte Offset: 0x21e2c
Read/Write: R/W
Parity Protection: See table below
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x40000000 (0b01xx,x000,0000,0000,xxxx,x000,0000,0000)

Bit	Parity Protection	Reset	Description
31:30	Y	0x1	**CPHY_CLKGEN_TRIGWIDTH_B**: Adjust AB/CB/CA pulse width in self-clock generator
26	N	0x0	**UPDATE_OFFSET_TRIO1_B**: Trigger to apply the a offset value. Used to update offset directly without trigger a new calibration.
25:17	Y	0x0	**DELAY_OFFSET_TRIO1_B**: Offset value of trio B1. Represents signed 2's complement number for calibration offset. Represents unsigned number if the field used as override value.
16	Y	0x0	**OVERRIDE_CAL_VAL_TRIO1_B**: Override for trio B1 0: Apply the DELAY_OFFSET as an offset to the calibration result. 1: Apply the DELAY_OFFSET value to the pad.
10	N	0x0	**UPDATE_OFFSET_TRIO0_B**: Trigger to apply the a offset value. Used to update offset directly without trigger a new calibration.
9:1	Y	0x0	**DELAY_OFFSET_TRIO0_B**: Offset value of trio B0. Represents signed 2's complement number for calibration offset. Represents unsigned number if the field used as override value.
0	Y	0x0	**OVERRIDE_CAL_VAL_TRIO0_B**: Override for trio B0 0: Apply the DELAY_OFFSET as an offset to the calibration result. 1: Apply the DELAY_OFFSET value to the pad.

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_EDGE_DELAY_VALUE_0

Offset: 0x878c
Byte Offset: 0x21e30
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000303 (0bxxxx,xxxx,xxxx,xxxx,0000,0011,0000,0011)

Bit	Reset	Description
15:8	0x3	**EDGE_DELAY_VALUE_TRIO1_B**: The edge delay value apply to pad finally

Bit	Reset	Description
7:0	0x3	**EDGE_DELAY_VALUE_TRIO0_B**: The edge delay value apply to pad finally

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_RESULT_0

Offset: 0x878d
Byte Offset: 0x21e34
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**EDGE_DELAY_CALIB_RESULT_1_TRIO1_B**: The edge delay calibrate result, with offset.
23:16	0x0	**EDGE_DELAY_CALIB_RESULT_1_TRIO0_B**: The edge delay calibrate result, with offset.
15:8	0x0	**EDGE_DELAY_CALIB_RESULT_0_TRIO1_B**: The edge delay calibrate result, with offset.
7:0	0x0	**EDGE_DELAY_CALIB_RESULT_0_TRIO0_B**: The edge delay calibrate result, with offset.

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_INADJ_CALIB_CTRL_0

Offset: 0x878e
Byte Offset: 0x21e38
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
16	0x0	**INADJ_CAL_TRIGGER_TRIO1_B**: Trigger INADJ calibration for trio0
0	0x0	**INADJ_CAL_TRIGGER_TRIO0_B**: Trigger INADJ calibration for trio1

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_INADJ_CALIB_CTRL1_0

Offset: 0x878f
Byte Offset: 0x21e3c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00010001 (0b0000,0000,0000,0001,0000,0000,0000,0001)

Bit	Reset	Description
31:16	0x1	**PRBS9_SEED_TRIO1_B**: PRBS9 initial seed for trio0
15:0	0x1	**PRBS9_SEED_TRIO0_B**: PRBS9 initial seed for trio1

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_INADJ_CALIB_CTRL2_0

Offset: 0x8790
Byte Offset: 0x21e40
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0fc00fc0 (0bxxxx,1111,1100,0000,xxxx,1111,1100,0000)

Bit	Reset	Description
27:22	0x3f	**INADJ_LIMIT_HIGH_TRIO1_B**: The upper limit for the INADJ sweep.
21:16	0x0	**INADJ_LIMIT_LOW_TRIO1_B**: The lower limit for the INADJ sweep.
11:6	0x3f	**INADJ_LIMIT_HIGH_TRIO0_B**: The upper limit for the INADJ sweep.
5:0	0x0	**INADJ_LIMIT_LOW_TRIO0_B**: The lower limit for the INADJ sweep.

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_INADJ_CALIB_STATUS_0

Offset: 0x8791
Byte Offset: 0x21e44

Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**INADJ_CALIB_DONE_TRIO1_B**: Trio b1 inadj calibration done.
0	0x0	**INADJ_CALIB_DONE_TRIO0_B**: Trio b0 inadj calibration done.

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_INADJ_CALIB_STATUS1_0

Offset: 0x8792
Byte Offset: 0x21e48
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_LOW_TRIO0_B**: Trio b0 PRBS check status for lower 32 INADJ value.

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_INADJ_CALIB_STATUS2_0

Offset: 0x8793
Byte Offset: 0x21e4c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_LOW_TRIO1_B**: Trio b1 PRBS check status for lower 32 INADJ value.

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_INADJ_CALIB_STATUS3_0

Offset: 0x8794
Byte Offset: 0x21e50
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_HIGH_TRIO0_B**: Trio b0 PRBS check status for upper 32 INADJ value.

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_INADJ_CALIB_STATUS4_0

Offset: 0x8795
Byte Offset: 0x21e54
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_HIGH_TRIO1_B**: Trio b1 PRBS check status for upper 32 INADJ value.

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_INADJ_OVERRIDE_0

Offset: 0x8796
Byte Offset: 0x21e58
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00080008 (0bxxxx,xxx0,0000,1000,xxxx,xxx0,0000,1000)

Bit	Reset	Description
24:17	0x4	**CPHY_INADJ_TRIO1_B**: The SW override INADJ value.
16	DISABLE	**SW_SET_CPHY_INADJ_TRIO1_B**: Enable SW override of the INADJ. **0** = DISABLE **1** = ENABLE
8:1	0x4	**CPHY_INADJ_TRIO0_B**: The SW override INADJ value.
0	DISABLE	**SW_SET_CPHY_INADJ_TRIO0_B**: Enable SW override of the INADJ. **0** = DISABLE **1** = ENABLE

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_BIST_CTRL_0

Offset: 0x8797
Byte Offset: 0x21e5c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	0x0	**STOP_BIST_TRIO1_B**: Manually stop the B1 BIST, write 1 to stop the RX bist.
2	0x0	**STOP_BIST_TRIO0_B**: Manually stop the B0 BIST, write 1 to stop the RX bist.
1	0x0	**TRIGGER_BIST_TRIO1_B**: Trigger the trio B1 BIST, write 1 to trigger the RX bist, HW will clear it to 0 when the BIST is done, write 0 to this field is with no effect.
0	0x0	**TRIGGER_BIST_TRIO0_B**: Trigger the trio B0 BIST, write 1 to trigger the RX bist, HW will clear it to 0 when the BIST is done, write 0 to this field is with no effect.

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_BIST_STATUS_0_0

Offset: 0x8798
Byte Offset: 0x21e60
Read/Write: RO
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:8	0x0	**WORD_ERR_CNT_TRIO1_B**: The trio B1 number of word comparison failure, report for BIST mode 1.
7:0	0x0	**WORD_ERR_CNT_TRIO0_B**: The trio B0 number of word comparison failure, report for BIST mode 1.

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_BIST_STATUS_1_0

Offset: 0x8799
Byte Offset: 0x21e64
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:8	0x0	**SYM_ERR_CNT_TRIO1_B**: The trio B1 number of symbol compariosn failure, it's the sum of the symbol err count and the clock shift count, the real wire state error number is 1/2 of this field, report for BIST mode 1.
7:0	0x0	**SYM_ERR_CNT_TRIO0_B**: The trio B0 number of symbol compariosn failure, it's the sum of the symbol err count and the clock shift count, the real wire state error number is 1/2 of this field, report for BIST mode 1.

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_BIST_STATUS_2_0

Offset: 0x879a
Byte Offset: 0x21e68
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WORD_CNT_LOW_TRIO0_B**: The trio B0 low 32 bits of the compared word number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_BIST_STATUS_3_0

Offset: 0x879b
Byte Offset: 0x21e6c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WORD_CNT_HIGH_TRIO0_B**: The trio B0 high 16 bits of the compared word number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_BIST_STATUS_4_0

Offset: 0x879c
Byte Offset: 0x21e70
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WORD_CNT_LOW_TRIO1_B**: The trio B1 low 32 bits of the compared word number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_BIST_STATUS_5_0

Offset: 0x879d
Byte Offset: 0x21e74
Read/Write: RO
Parity Protection: N
Shadow: N

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WORD_CNT_HIGH_TRIO1_B**: The trio B1 high 16 bits of the compared word number number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_1_NVCSI_CIL_B_CPHY_BIST_STATUS_6_0

Offset: 0x879e
Byte Offset: 0x21e78
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,x000,0000,0000,xxxx,x000,0000,0000)

Bit	Reset	Description
26	0x0	**ERR_DETECT_FAIL_TRIO1_B**: Multi symbol error in one word (7 symbol) for trio B1, BIST logic is not able to detect the accurate wire state error. With this error, if not other errors detected, the word error is accurate.
25	0x0	**SHIFT_DETECT_FAIL_TRIO1_B**: Detect symbol error in two continuous word (14 symbols) for trio B1, BIST logic is not able to determine if symbol clock is slip. (if the symbol error count is not saturating, the result is still available with this error)
24	0x0	**SHIFT_OVERFLOW_TRIO1_B**: Too many clocks slip and the BIST logic is not able to re-align for trio B1. Typically, when this error is detect, the symbol error counter will saturate.
23:20	0x0	**DOUBLE_CLK_MISSED_CNT_TRIO1_B**: Number of the times for double symbol clock lost for trio B1.
19:16	0x0	**SINGLE_CLK_MISSED_CNT_TRIO1_B**: Number of the times for single symbol clock lost for trio B1.
10	0x0	**ERR_DETECT_FAIL_TRIO0_B**: Multi symbol error in one word (7 symbol) for trio B0, BIST logic is not able to detect the accurate wire state error. With this error, if not other errors detected, the word error is accurate.
9	0x0	**SHIFT_DETECT_FAIL_TRIO0_B**: Detect symbol error in two continuous word (14 symbols) for trio B0, BIST logic is not able to determine if symbol clock is slip. (if the symbol error count is not saturating, the result is still available with this error)
8	0x0	**SHIFT_OVERFLOW_TRIO0_B**: Too many clocks slip and the BIST logic is not able to re-align for trio B0. Typically, when this error is detect, the symbol error counter will saturate.

Bit	Reset	Description
7:4	0x0	**DOUBLE_CLK_MISSED_CNT_TRIO0_B**: Number of the times for double symbol clock lost for trio B0.
3:0	0x0	**SINGLE_CLK_MISSED_CNT_TRIO0_B**: Number of the times for single symbol clock lost for trio B0.

```
base NVCSI_PHY_2_CILA 0x030400;
```

NVCSI_PHY_2_CILA_INTR_0_STATUS_CILA_0

Offset: 0xc500
Byte Offset: 0x31400
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,00x0,0xxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
30	0x0	**intr_cphy_edge_delay_cal_done_trio1_a**
29	0x0	**intr_cphy_edge_delay_cal_done_trio0_a**
28	0x0	**intr_dphy_cil_lane_align_err_a**
27	0x0	**intr_dphy_cil_deskew_calib_err_ctrl_a**
26	0x0	**intr_dphy_cil_deskew_calib_err_lane1_a**
25	0x0	**intr_dphy_cil_deskew_calib_err_lane0_a**
24	0x0	**intr_dphy_cil_deskew_calib_done_ctrl_a**
23	0x0	**intr_dphy_cil_deskew_calib_done_lane1_a**
22	0x0	**intr_dphy_cil_deskew_calib_done_lane0_a**
20	0x0	**intr_cil_data_lane_esc_mode_sync_err1_a**
19	0x0	**intr_cil_data_lane_esc_mode_sync_err0_a**
8	0x0	**intr_cil_data_lane_rxfifo_full_err1_a**
7	0x0	**intr_cil_data_lane_ctrl_err1_a**
6	0x0	**intr_cil_data_lane_sot_mb_err1_a**
5	0x0	**intr_cil_data_lane_sot_sb_err1_a**
4	0x0	**intr_cil_data_lane_rxfifo_full_err0_a**

Bit	Reset	Description
3	0x0	**intr_cil_data_lane_ctrl_err0_a**
2	0x0	**intr_cil_data_lane_sot_mb_err0_a**
1	0x0	**intr_cil_data_lane_sot_sb_err0_a**
0	0x0	**intr_dphy_cil_clk_lane_ctrl_err_a**

NVCSI_PHY_2_CILA_INTR_1_STATUS_CILA_0

Offset: 0xc501
Byte Offset: 0x31404
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**intr_cphy_inadj_cal_done_trio1_a**
13	0x0	**intr_cphy_inadj_cal_done_trio0_a**
12	0x0	**intr_idac_cal_done_trio1_a**
11	0x0	**intr_idac_cal_done_trio0_a**
10	0x0	**intr_dphy_cil_clk_lane_ulpm_req_a**
9	0x0	**intr_cil_lpdt_int1_a**
8	0x0	**intr_cil_ulps_trigger_int1_a**
7	0x0	**intr_cil_remoterst_trigger_int1_a**
6	0x0	**intr_cil_lpdt_int0_a**
5	0x0	**intr_cil_ulps_trigger_int0_a**
4	0x0	**intr_cil_remoterst_trigger_int0_a**
3	0x0	**intr_cil_data_lane_esc_data_rec1_a**
2	0x0	**intr_cil_data_lane_esc_cmd_rec1_a**
1	0x0	**intr_cil_data_lane_esc_data_rec0_a**
0	0x0	**intr_cil_data_lane_esc_cmd_rec0_a**

NVCSI_PHY_2_CILA_INTR_0_MASK_CILA_0
Offset: 0xc502
Byte Offset: 0x31408
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,00x0,0xxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
30	0x0	**intr_mask_cphy_edge_delay_cal_done_trio1_a**
29	0x0	**intr_mask_cphy_edge_delay_cal_done_trio0_a**
28	0x0	**intr_mask_dphy_cil_lane_align_err_a**
27	0x0	**intr_mask_dphy_cil_deskew_calib_err_ctrl_a**
26	0x0	**intr_mask_dphy_cil_deskew_calib_err_lane1_a**
25	0x0	**intr_mask_dphy_cil_deskew_calib_err_lane0_a**
24	0x0	**intr_mask_dphy_cil_deskew_calib_done_ctrl_a**
23	0x0	**intr_mask_dphy_cil_deskew_calib_done_lane1_a**
22	0x0	**intr_mask_dphy_cil_deskew_calib_done_lane0_a**
20	0x0	**intr_mask_cil_data_lane_esc_mode_sync_err1_a**
19	0x0	**intr_mask_cil_data_lane_esc_mode_sync_err0_a**
8	0x0	**intr_mask_cil_data_lane_rxfifo_full_err1_a**
7	0x0	**intr_mask_cil_data_lane_ctrl_err1_a**
6	0x0	**intr_mask_cil_data_lane_sot_mb_err1_a**
5	0x0	**intr_mask_cil_data_lane_sot_sb_err1_a**
4	0x0	**intr_mask_cil_data_lane_rxfifo_full_err0_a**
3	0x0	**intr_mask_cil_data_lane_ctrl_err0_a**
2	0x0	**intr_mask_cil_data_lane_sot_mb_err0_a**
1	0x0	**intr_mask_cil_data_lane_sot_sb_err0_a**
0	0x0	**intr_mask_dphy_cil_clk_lane_ctrl_err_a**

NVCSI_PHY_2_CILA_INTR_1_MASK_CILA_0

Offset: 0xc503
Byte Offset: 0x3140c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**intr_mask_cphy_inadj_cal_done_trio1_a**
13	0x0	**intr_mask_cphy_inadj_cal_done_trio0_a**
12	0x0	**intr_mask_idac_cal_done_trio1_a**
11	0x0	**intr_mask_idac_cal_done_trio0_a**
10	0x0	**intr_mask_dphy_cil_clk_lane_ulpm_req_a**
9	0x0	**intr_mask_cil_lpdt_int1_a**
8	0x0	**intr_mask_cil_ulps_trigger_int1_a**
7	0x0	**intr_mask_cil_remoterst_trigger_int1_a**
6	0x0	**intr_mask_cil_lpdt_int0_a**
5	0x0	**intr_mask_cil_ulps_trigger_int0_a**
4	0x0	**intr_mask_cil_remoterst_trigger_int0_a**
3	0x0	**intr_mask_cil_data_lane_esc_data_rec1_a**
2	0x0	**intr_mask_cil_data_lane_esc_cmd_rec1_a**
1	0x0	**intr_mask_cil_data_lane_esc_data_rec0_a**
0	0x0	**intr_mask_cil_data_lane_esc_cmd_rec0_a**

NVCSI_PHY_2_CILA_CORRECTABLE_ERR_INTR_STATUS_CILA_0

Offset: 0xc504
Byte Offset: 0x31410
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**err_intr_cil_data_lane_esc_mode_sync_err1_a**
13	0x0	**err_intr_cil_data_lane_esc_mode_sync_err0_a**
12	0x0	**err_intr_dphy_cil_lane_align_err_a**
11	0x0	**err_intr_dphy_cil_deskew_calib_err_ctrl_a**
10	0x0	**err_intr_dphy_cil_deskew_calib_err_lane1_a**
9	0x0	**err_intr_dphy_cil_deskew_calib_err_lane0_a**
8	0x0	**err_intr_cil_data_lane_rxfifo_full_err1_a**
7	0x0	**err_intr_cil_data_lane_ctrl_err1_a**
6	0x0	**err_intr_cil_data_lane_sot_mb_err1_a**
5	0x0	**err_intr_cil_data_lane_sot_sb_err1_a**
4	0x0	**err_intr_cil_data_lane_rxfifo_full_err0_a**
3	0x0	**err_intr_cil_data_lane_ctrl_err0_a**
2	0x0	**err_intr_cil_data_lane_sot_mb_err0_a**
1	0x0	**err_intr_cil_data_lane_sot_sb_err0_a**
0	0x0	**err_intr_dphy_cil_clk_lane_ctrl_err_a**

NVCSI_PHY_2_CILA_UNCORRECTABLE_ERR_INTR_STATUS_CILA_0

Offset: 0xc505
Byte Offset: 0x31414
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**err_intr_cil_data_lane_esc_mode_sync_err1_a**
13	0x0	**err_intr_cil_data_lane_esc_mode_sync_err0_a**
12	0x0	**err_intr_dphy_cil_lane_align_err_a**
11	0x0	**err_intr_dphy_cil_deskew_calib_err_ctrl_a**

Bit	Reset	Description
10	0x0	**err_intr_dphy_cil_deskew_calib_err_lane1_a**
9	0x0	**err_intr_dphy_cil_deskew_calib_err_lane0_a**
8	0x0	**err_intr_cil_data_lane_rxfifo_full_err1_a**
7	0x0	**err_intr_cil_data_lane_ctrl_err1_a**
6	0x0	**err_intr_cil_data_lane_sot_mb_err1_a**
5	0x0	**err_intr_cil_data_lane_sot_sb_err1_a**
4	0x0	**err_intr_cil_data_lane_rxfifo_full_err0_a**
3	0x0	**err_intr_cil_data_lane_ctrl_err0_a**
2	0x0	**err_intr_cil_data_lane_sot_mb_err0_a**
1	0x0	**err_intr_cil_data_lane_sot_sb_err0_a**
0	0x0	**err_intr_dphy_cil_clk_lane_ctrl_err_a**

NVCSI_PHY_2_CILA_ERR_INTR_MASK_CILA_0

Offset: 0xc506
Byte Offset: 0x31418
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00007fff (0bxxxx,xxxx,xxxx,xxxx,x111,1111,1111,1111)

Bit	Reset	Description
14	0x1	**err_intr_mask_cil_data_lane_esc_mode_sync_err1_a**
13	0x1	**err_intr_mask_cil_data_lane_esc_mode_sync_err0_a**
12	0x1	**err_intr_mask_dphy_cil_lane_align_err_a**
11	0x1	**err_intr_mask_dphy_cil_deskew_calib_err_ctrl_a**
10	0x1	**err_intr_mask_dphy_cil_deskew_calib_err_lane1_a**
9	0x1	**err_intr_mask_dphy_cil_deskew_calib_err_lane0_a**
8	0x1	**err_intr_mask_cil_data_lane_rxfifo_full_err1_a**
7	0x1	**err_intr_mask_cil_data_lane_ctrl_err1_a**
6	0x1	**err_intr_mask_cil_data_lane_sot_mb_err1_a**

Bit	Reset	Description
5	0x1	**err_intr_mask_cil_data_lane_sot_sb_err1_a**
4	0x1	**err_intr_mask_cil_data_lane_rxfifo_full_err0_a**
3	0x1	**err_intr_mask_cil_data_lane_ctrl_err0_a**
2	0x1	**err_intr_mask_cil_data_lane_sot_mb_err0_a**
1	0x1	**err_intr_mask_cil_data_lane_sot_sb_err0_a**
0	0x1	**err_intr_mask_dphy_cil_clk_lane_ctrl_err_a**

NVCSI_PHY_2_CILA_ERR_INTR_TYPE_CILA_0

Offset: 0xc507
Byte Offset: 0x3141c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**err_intr_type_cil_data_lane_esc_mode_sync_err1_a**
13	0x0	**err_intr_type_cil_data_lane_esc_mode_sync_err0_a**
12	0x0	**err_intr_type_dphy_cil_lane_align_err_a**
11	0x0	**err_intr_type_dphy_cil_deskew_calib_err_ctrl_a**
10	0x0	**err_intr_type_dphy_cil_deskew_calib_err_lane1_a**
9	0x0	**err_intr_type_dphy_cil_deskew_calib_err_lane0_a**
8	0x0	**err_intr_type_cil_data_lane_rxfifo_full_err1_a**
7	0x0	**err_intr_type_cil_data_lane_ctrl_err1_a**
6	0x0	**err_intr_type_cil_data_lane_sot_mb_err1_a**
5	0x0	**err_intr_type_cil_data_lane_sot_sb_err1_a**
4	0x0	**err_intr_type_cil_data_lane_rxfifo_full_err0_a**
3	0x0	**err_intr_type_cil_data_lane_ctrl_err0_a**
2	0x0	**err_intr_type_cil_data_lane_sot_mb_err0_a**
1	0x0	**err_intr_type_cil_data_lane_sot_sb_err0_a**

Bit	Reset	Description
0	0x0	**err_intr_type_dphy_cil_clk_lane_ctrl_err_a**

NVCSI_PHY_2_CILB_INTR_0_STATUS_CILB_0

Offset: 0xc600
Byte Offset: 0x31800
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,00x0,0xxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
30	0x0	**intr_cphy_edge_delay_cal_done_trio1_b**
29	0x0	**intr_cphy_edge_delay_cal_done_trio0_b**
28	0x0	**intr_dphy_cil_lane_align_err_b**
27	0x0	**intr_dphy_cil_deskew_calib_err_ctrl_b**
26	0x0	**intr_dphy_cil_deskew_calib_err_lane1_b**
25	0x0	**intr_dphy_cil_deskew_calib_err_lane0_b**
24	0x0	**intr_dphy_cil_deskew_calib_done_ctrl_b**
23	0x0	**intr_dphy_cil_deskew_calib_done_lane1_b**
22	0x0	**intr_dphy_cil_deskew_calib_done_lane0_b**
20	0x0	**intr_cil_data_lane_esc_mode_sync_err1_b**
19	0x0	**intr_cil_data_lane_esc_mode_sync_err0_b**
8	0x0	**intr_cil_data_lane_rxfifo_full_err1_b**
7	0x0	**intr_cil_data_lane_ctrl_err1_b**
6	0x0	**intr_cil_data_lane_sot_mb_err1_b**
5	0x0	**intr_cil_data_lane_sot_sb_err1_b**
4	0x0	**intr_cil_data_lane_rxfifo_full_err0_b**
3	0x0	**intr_cil_data_lane_ctrl_err0_b**
2	0x0	**intr_cil_data_lane_sot_mb_err0_b**

Bit	Reset	Description
1	0x0	**intr_cil_data_lane_sot_sb_err0_b**
0	0x0	**intr_dphy_cil_clk_lane_ctrl_err_b**

NVCSI_PHY_2_CILB_INTR_1_STATUS_CILB_0

Offset: 0xc601
Byte Offset: 0x31804
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**intr_cphy_inadj_cal_done_trio1_b**
13	0x0	**intr_cphy_inadj_cal_done_trio0_b**
12	0x0	**intr_idac_cal_done_trio1_b**
11	0x0	**intr_idac_cal_done_trio0_b**
10	0x0	**intr_dphy_cil_clk_lane_ulpm_req_b**
9	0x0	**intr_cil_lpdt_int1_b**
8	0x0	**intr_cil_ulps_trigger_int1_b**
7	0x0	**intr_cil_remoterst_trigger_int1_b**
6	0x0	**intr_cil_lpdt_int0_b**
5	0x0	**intr_cil_ulps_trigger_int0_b**
4	0x0	**intr_cil_remoterst_trigger_int0_b**
3	0x0	**intr_cil_data_lane_esc_data_rec1_b**
2	0x0	**intr_cil_data_lane_esc_cmd_rec1_b**
1	0x0	**intr_cil_data_lane_esc_data_rec0_b**
0	0x0	**intr_cil_data_lane_esc_cmd_rec0_b**

NVCSI_PHY_2_CILB_INTR_0_MASK_CILB_0

Offset: 0xc602

Byte Offset: 0x31808
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,00x0,0xxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
30	0x0	**intr_mask_cphy_edge_delay_cal_done_trio1_b**
29	0x0	**intr_mask_cphy_edge_delay_cal_done_trio0_b**
28	0x0	**intr_mask_dphy_cil_lane_align_err_b**
27	0x0	**intr_mask_dphy_cil_deskew_calib_err_ctrl_b**
26	0x0	**intr_mask_dphy_cil_deskew_calib_err_lane1_b**
25	0x0	**intr_mask_dphy_cil_deskew_calib_err_lane0_b**
24	0x0	**intr_mask_dphy_cil_deskew_calib_done_ctrl_b**
23	0x0	**intr_mask_dphy_cil_deskew_calib_done_lane1_b**
22	0x0	**intr_mask_dphy_cil_deskew_calib_done_lane0_b**
20	0x0	**intr_mask_cil_data_lane_esc_mode_sync_err1_b**
19	0x0	**intr_mask_cil_data_lane_esc_mode_sync_err0_b**
8	0x0	**intr_mask_cil_data_lane_rxfifo_full_err1_b**
7	0x0	**intr_mask_cil_data_lane_ctrl_err1_b**
6	0x0	**intr_mask_cil_data_lane_sot_mb_err1_b**
5	0x0	**intr_mask_cil_data_lane_sot_sb_err1_b**
4	0x0	**intr_mask_cil_data_lane_rxfifo_full_err0_b**
3	0x0	**intr_mask_cil_data_lane_ctrl_err0_b**
2	0x0	**intr_mask_cil_data_lane_sot_mb_err0_b**
1	0x0	**intr_mask_cil_data_lane_sot_sb_err0_b**
0	0x0	**intr_mask_dphy_cil_clk_lane_ctrl_err_b**

NVCSI_PHY_2_CILB_INTR_1_MASK_CILB_0

Offset: 0xc603
Byte Offset: 0x3180c
Read/Write: R/W

Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**intr_mask_cphy_inadj_cal_done_trio1_b**
13	0x0	**intr_mask_cphy_inadj_cal_done_trio0_b**
12	0x0	**intr_mask_idac_cal_done_trio1_b**
11	0x0	**intr_mask_idac_cal_done_trio0_b**
10	0x0	**intr_mask_dphy_cil_clk_lane_ulpm_req_b**
9	0x0	**intr_mask_cil_lpdt_int1_b**
8	0x0	**intr_mask_cil_ulps_trigger_int1_b**
7	0x0	**intr_mask_cil_remoterst_trigger_int1_b**
6	0x0	**intr_mask_cil_lpdt_int0_b**
5	0x0	**intr_mask_cil_ulps_trigger_int0_b**
4	0x0	**intr_mask_cil_remoterst_trigger_int0_b**
3	0x0	**intr_mask_cil_data_lane_esc_data_rec1_b**
2	0x0	**intr_mask_cil_data_lane_esc_cmd_rec1_b**
1	0x0	**intr_mask_cil_data_lane_esc_data_rec0_b**
0	0x0	**intr_mask_cil_data_lane_esc_cmd_rec0_b**

NVCSI_PHY_2_CILB_CORRECTABLE_ERR_INTR_STATUS_CILB_0

Offset: 0xc604
Byte Offset: 0x31810
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**err_intr_cil_data_lane_esc_mode_sync_err1_b**
13	0x0	**err_intr_cil_data_lane_esc_mode_sync_err0_b**

Bit	Reset	Description
12	0x0	**err_intr_dphy_cil_lane_align_err_b**
11	0x0	**err_intr_dphy_cil_deskew_calib_err_ctrl_b**
10	0x0	**err_intr_dphy_cil_deskew_calib_err_lane1_b**
9	0x0	**err_intr_dphy_cil_deskew_calib_err_lane0_b**
8	0x0	**err_intr_cil_data_lane_rxfifo_full_err1_b**
7	0x0	**err_intr_cil_data_lane_ctrl_err1_b**
6	0x0	**err_intr_cil_data_lane_sot_mb_err1_b**
5	0x0	**err_intr_cil_data_lane_sot_sb_err1_b**
4	0x0	**err_intr_cil_data_lane_rxfifo_full_err0_b**
3	0x0	**err_intr_cil_data_lane_ctrl_err0_b**
2	0x0	**err_intr_cil_data_lane_sot_mb_err0_b**
1	0x0	**err_intr_cil_data_lane_sot_sb_err0_b**
0	0x0	**err_intr_dphy_cil_clk_lane_ctrl_err_b**

NVCSI_PHY_2_CILB_UNCORRECTABLE_ERR_INTR_STATUS_CILB_0

Offset: 0xc605
Byte Offset: 0x31814
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**err_intr_cil_data_lane_esc_mode_sync_err1_b**
13	0x0	**err_intr_cil_data_lane_esc_mode_sync_err0_b**
12	0x0	**err_intr_dphy_cil_lane_align_err_b**
11	0x0	**err_intr_dphy_cil_deskew_calib_err_ctrl_b**
10	0x0	**err_intr_dphy_cil_deskew_calib_err_lane1_b**
9	0x0	**err_intr_dphy_cil_deskew_calib_err_lane0_b**
8	0x0	**err_intr_cil_data_lane_rxfifo_full_err1_b**

Bit	Reset	Description
7	0x0	**err_intr_cil_data_lane_ctrl_err1_b**
6	0x0	**err_intr_cil_data_lane_sot_mb_err1_b**
5	0x0	**err_intr_cil_data_lane_sot_sb_err1_b**
4	0x0	**err_intr_cil_data_lane_rxfifo_full_err0_b**
3	0x0	**err_intr_cil_data_lane_ctrl_err0_b**
2	0x0	**err_intr_cil_data_lane_sot_mb_err0_b**
1	0x0	**err_intr_cil_data_lane_sot_sb_err0_b**
0	0x0	**err_intr_dphy_cil_clk_lane_ctrl_err_b**

NVCSI_PHY_2_CILB_ERR_INTR_MASK_CILB_0

Offset: 0xc606
Byte Offset: 0x31818
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00007fff (0bxxxx,xxxx,xxxx,xxxx,x111,1111,1111,1111)

Bit	Reset	Description
14	0x1	**err_intr_mask_cil_data_lane_esc_mode_sync_err1_b**
13	0x1	**err_intr_mask_cil_data_lane_esc_mode_sync_err0_b**
12	0x1	**err_intr_mask_dphy_cil_lane_align_err_b**
11	0x1	**err_intr_mask_dphy_cil_deskew_calib_err_ctrl_b**
10	0x1	**err_intr_mask_dphy_cil_deskew_calib_err_lane1_b**
9	0x1	**err_intr_mask_dphy_cil_deskew_calib_err_lane0_b**
8	0x1	**err_intr_mask_cil_data_lane_rxfifo_full_err1_b**
7	0x1	**err_intr_mask_cil_data_lane_ctrl_err1_b**
6	0x1	**err_intr_mask_cil_data_lane_sot_mb_err1_b**
5	0x1	**err_intr_mask_cil_data_lane_sot_sb_err1_b**
4	0x1	**err_intr_mask_cil_data_lane_rxfifo_full_err0_b**
3	0x1	**err_intr_mask_cil_data_lane_ctrl_err0_b**

Bit	Reset	Description
2	0x1	**err_intr_mask_cil_data_lane_sot_mb_err0_b**
1	0x1	**err_intr_mask_cil_data_lane_sot_sb_err0_b**
0	0x1	**err_intr_mask_dphy_cil_clk_lane_ctrl_err_b**

NVCSI_PHY_2_CILB_ERR_INTR_TYPE_CILB_0

Offset: 0xc607
Byte Offset: 0x3181c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**err_intr_type_cil_data_lane_esc_mode_sync_err1_b**
13	0x0	**err_intr_type_cil_data_lane_esc_mode_sync_err0_b**
12	0x0	**err_intr_type_dphy_cil_lane_align_err_b**
11	0x0	**err_intr_type_dphy_cil_deskew_calib_err_ctrl_b**
10	0x0	**err_intr_type_dphy_cil_deskew_calib_err_lane1_b**
9	0x0	**err_intr_type_dphy_cil_deskew_calib_err_lane0_b**
8	0x0	**err_intr_type_cil_data_lane_rxfifo_full_err1_b**
7	0x0	**err_intr_type_cil_data_lane_ctrl_err1_b**
6	0x0	**err_intr_type_cil_data_lane_sot_mb_err1_b**
5	0x0	**err_intr_type_cil_data_lane_sot_sb_err1_b**
4	0x0	**err_intr_type_cil_data_lane_rxfifo_full_err0_b**
3	0x0	**err_intr_type_cil_data_lane_ctrl_err0_b**
2	0x0	**err_intr_type_cil_data_lane_sot_mb_err0_b**
1	0x0	**err_intr_type_cil_data_lane_sot_sb_err0_b**
0	0x0	**err_intr_type_dphy_cil_clk_lane_ctrl_err_b**

```
base NVCSI_PHY_2 0x031000;
```

NVCSI_PHY_2_NVCSI_CIL_PHY_CTRL_0

Offset: 0xc700
Byte Offset: 0x31c00
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DPHY	**CFG_PHY_MODE**: The CPHY/DPHY type mode. **0** = DPHY **1** = CPHY

NVCSI_PHY_2_LM_SLCG_CTRL_0

Offset: 0xc701
Byte Offset: 0x31c04
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	DISABLE	**cfg_lm1_slcg_override**: Enable the SLCG override for lane merger 1. **0** = DISABLE **1** = ENABLE
0	DISABLE	**cfg_lm0_slcg_override**: Enable the SLCG override for lane merger 0. **0** = DISABLE **1** = ENABLE

NVCSI_PHY_2_LM_EN_CTRL_0

Offset: 0xc702
Byte Offset: 0x31c08

Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000060c (0bxxxx,xxxx,xxxx,xxxx,xxx0,0110,xx00,11x0)

Bit	Reset	Description
12:9	0x3	**cfg_lm1_water_mark**: Water mark for LM0 detect stuck
8	DISABLE	**cfg_lm1_en**: Lane Merger 1 enable **0** = DISABLE **1** = ENABLE
5:2	0x3	**cfg_lm0_water_mark**: Water mark for LM0 detect stuck
0	DISABLE	**cfg_lm0_en**: Lane Merger 0 enable **0** = DISABLE **1** = ENABLE

NVCSI_PHY_2_LM_SW_RESET_0
Offset: 0xc703
Byte Offset: 0x31c0c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	DISABLE	**cfg_lm1_swreset**: Reset the lane merger 1. **0** = DISABLE **1** = ENABLE
0	DISABLE	**cfg_lm0_swreset**: Reset the lane merger 0. **0** = DISABLE **1** = ENABLE

NVCSI_PHY_2_NVCSI_CIL_CONFIG_0

Offset: 0xc704
Byte Offset: 0x31c10
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,x000,xxxx,x000)

Bit	Reset	Description
10:8	0x0	**DATA_LANE_B**: Lane number for lane merger B. 000: 0 lanes active 001: 1 lanes active 010: 2 lanes active Others: illegal
2:0	0x0	**DATA_LANE_A**: Lane number for lane merger A. 000: 0 lanes active 001: 1 lanes active 010: 2 lanes active 011: 3 lanes active (valid only in CPHY mode. NA in DPHY mode) 100: 4 lanes active, CLK from Partition A is used. 101: 4 lanes active, CLK from Partition B is used. Others: illegal

NVCSI_PHY_2_NVCSI_CIL_CLKEN_OVERRIDE_CTRL_0

Offset: 0xc705
Byte Offset: 0x31c14
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**CLKEN_OVERRIDE**: CLKENABLE OVERRIDE FOR CIL. **1** = ENABLE **0** = DISABLE

NVCSI_PHY_2_NVCSI_CIL_PAD_CONFIG_0

Offset: 0xc706

Byte Offset: 0x31c18
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00040100 (0bxxxx,x000,0000,0100,0000,xxx1,0000,0000)

Bit	Reset	Description
26:25	0x0	**VD09REG_LEVEL**: 2-bit regulator level control. control voltage level output of 0.9v regulator. 00:0.9v, 01:0.85v, 10: 0.95v, 11: 1v
24:23	0x0	**VD09REG_LEAKER**: 2-bit regulator leaker control. 11 will give best performance, but consumers more DC quiescent current. (control 0.9v internal regulator for all partitions)
22:21	0x0	**VD04REG_LEVEL**: 2-bit regulator level control control voltage level output of 0.4V regulator 00: 0.4v 01: 0.38V 10: 0.42V 11: 0.44V
20:19	0x0	**VD04REG_LEAKER**: 2-bit regulator leaker control. 11 will give best performance, but consumers more DC quiescent current. (control 0.4v internal tx regulator for all partitions)
18:16	0x4	**LPRX_LEVEL_SEL**: Internal vref adjustment for LP self-biased receiver(when LP_RX_SELECT_IO1_[A/B]=1)
15:12	0x0	**LOADADJ**: LOAD ADJ value to be connected to the pad not used for CSI functional mode. used only for IO bist
8	0x1	**PDVCLAMP**: Power down regular which supplies current to de-serializer logic. Active High.
7:5	0x0	**SEL_CKTEST**
4	0x0	**E_CKTEST**: Enable clock test output.
3:1	0x0	**SEL_ATEST**
0	0x0	**E_ATEST**: Enable analog test voltage output.

NVCSI_PHY_2_NVCSI_CIL_PAD_CONFIG_1_0

Offset: 0xc707
Byte Offset: 0x31c1c
Read/Write: R/W

Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE_TOP**: spare register bits for top level control

NVCSI_PHY_2_NVCSI_CIL_LANE_SWIZZLE_CTRL_0

Offset: 0xc708
Byte Offset: 0x31c20
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4:0	0x0	**LANE_SWIZZLE_CTRL**: Lane Swizzle control for Brick. Valid in both CPHY and DPHY modes. For CPHY, all modes are supported. But for DPHY, some mode are not supported with different lane count. Only for 4 lanes: 00011 A0 A1 B0 B1 --> A0 B0 A1 B1 00100 A0 A1 B0 B1 --> A0 B1 A1 B0 00101 A0 A1 B0 B1 --> A0 B1 B0 A1 00010 A0 A1 B0 B1 --> A0 B0 B1 A1 01000 A0 A1 B0 B1 --> A1 B0 B1 A0 01001 A0 A1 B0 B1 --> A1 B0 A0 B1 01010 A0 A1 B0 B1 --> A1 B1 A0 B0 01011 A0 A1 B0 B1 --> A1 B1 B0 A0 01100 A0 A1 B0 B1 --> B0 A1 A0 B1 01101 A0 A1 B0 B1 --> B0 A1 B1 A0 01110 A0 A1 B0 B1 --> B0 A0 B1 A1 01111 A0 A1 B0 B1 --> B0 A0 A1 B1 10000 A0 A1 B0 B1 --> B0 B1 A1 A0 10001 A0 A1 B0 B1 --> B0 B1 A0 A1 10010 A0 A1 B0 B1 --> B1 A1 B0 A0 10011 A0 A1 B0 B1 --> B1 A1 A0 B0 10100 A0 A1 B0 B1 --> B1 B0 A0 A1 10101 A0 A1 B0 B1 --> B1 B0 A1 A0 10110 A0 A1 B0 B1 --> B1 A0 A1 B0 10111 A0 A1 B0 B1 --> B1 A0 B0 A1 Support for 1/2/4 lanes 00000 A0 A1 B0 B1 --> A0 A1 B0 B1 00001 A0 A1 B0 B1 --> A0 A1 B1 B0 00110 A0 A1 B0 B1 --> A1 A0 B0 B1 00111 A0 A1 B0 B1 --> A1 A0 B1 B0

NVCSI_PHY_2_NVCSI_CIL_BK_MODE_STATUS_0

Offset: 0xc709
Byte Offset: 0x31c24
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
1:0	X	**BK_MODE**: 00: two independent 2x bricks 01: one 4x brick, received clock from partition A is used. Clock from partition B is not used 10: one 4x brick, received clock from partition B is used. Clock from partition A is not used 11: illegal

NVCSI_PHY_2_NVCSI_CIL_TX_TIMING_0_0

Offset: 0xc70a
Byte Offset: 0x31c28
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00006030 (0bxxxx,xxxx,xxxx,xxxx,0110,0000,0011,0000)

Bit	Reset	Description
15:8	0x60	**THSEXIT**: THSEXIT length, in slow clock cycle number
7:0	0x30	**TLPX**: TLPX length, in slow clock cycle number

NVCSI_PHY_2_NVCSI_CIL_TX_TIMING_1_0

Offset: 0xc70b
Byte Offset: 0x31c2c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00101030 (0bxxxx,xxxx,0001,0000,0001,0000,0011,0000)

Bit	Reset	Description
23:16	0x10	**T3POST**: T3POST length, in slow clock cycle number
15:8	0x10	**T3PREBEGIN**: T3PREBEGIN length, in slow clock cycle number
7:0	0x30	**T3PREPARE**: T3PREPARE length, in slow clock cycle number

NVCSI_PHY_2_NVCSI_CIL_TX_TIMING_2_0

Offset: 0xc70c
Byte Offset: 0x31c30
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:8	0xfff	**T3CALALTSEQ**: T3CALALTSEQ length, in slow clock cycle number
7:0	0xff	**T3CALPREAMBLE**: T3CALPREAMBLE length, in slow clock cycle number

NVCSI_PHY_2_NVCSI_CIL_TX_CALIB_CTRL_0

Offset: 0xc70d
Byte Offset: 0x31c34
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000004 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0100)

Bit	Reset	Description
17:2	0x1	**CALIB_SEED**: Calibration sequence format 2 PRBS seed. Should set to a non-zero value.

Bit	Reset	Description
1	FMT_1	**CALIB_SEQ**: Calibration sequence format. **0** = FMT_1 **1** = FMT_2
0	DISABLE	**CALIB_EN**: Calibration sequence enable **0** = DISABLE **1** = ENABLE

NVCSI_PHY_2_NVCSI_CIL_TX_CTRL_0

Offset: 0xc70e
Byte Offset: 0x31c38
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000004 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0100)

Bit	Reset	Description
4	DISABLE	**TX_ENABLE**: Enable the TX SCIL. **0** = DISABLE **1** = ENABLE
3	SINGLE	**PAD_BK_MODE**: Set the pad to one stream or two independent stream usage. **0** = SINGLE **1** = DUAL
2:0	0x4	**LANE_NUM**: Valid value: 1/2/4

NVCSI_PHY_2_NVCSI_CIL_TX_SW_RESET_0_0

Offset: 0xc70f
Byte Offset: 0x31c3c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	ENABLE	**SW_RESET**: Reset the TX lane **0** = DISABLE **1** = ENABLE

NVCSI_PHY_2_NVCSI_CIL_CPHY_BIST_CONFIG_0_0

Offset: 0xc710
Byte Offset: 0x31c40
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000009 (0bxxxx,xxxx,xx00,0000,0000,0000,0000,1001)

Bit	Reset	Description
21	DISABLE	**REALIGN_EN**: If set to enable, the BIST logic will re-align the reference symbol to match the received symbol when there is clock slip by wire state error. **0** = DISABLE **1** = ENABLE
20:3	0x1	**PRBS_SEED**: Seed of the PRBS pattern. Should set to a non-zero value.
2:1	0x0	**PRBS_PATTERN**: PRBS pattern PRBS9: X0+X5+X9 PRBS11: X0+X9+X11 PRBS18: X0+X11+X18 **0** = PRBS9 **1** = PRBS11 **2** = PRBS18
0	MODE1	**BIST_MODE**: BIST control mode MODE0: Set the expected error number, BIST logic will keep comparing the received symbols and the reference PRBS generated symbols. When the expected error number got, BIST logic stop. And report how many word are totally compared. MODE11: Set the expected word number, BIST logic will keep comparing the received symbols and the reference PRBS generated symbols. When the expected words(symbols) are compared, BIST logic stop and report how many errors are found. **0** = MODE0 **1** = MODE1

NVCSI_PHY_2_NVCSI_CIL_CPHY_BIST_CONFIG_1_0

Offset: 0xc711
Byte Offset: 0x31c44
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000100 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx1,0000,0000)

Bit	Reset	Description
8	0x1	**ERR_TYPE**: The expecte error number type. SYMBOL_NUM: symbol error number. **0** = WORD_NUM :WORD_NUM: word error number. **1** = SYMBOL_NUM
7:0	0x0	**ERR_NUM**: The expected error number, used for BIST MODE 0.

NVCSI_PHY_2_NVCSI_CIL_CPHY_BIST_CONFIG_2_0

Offset: 0xc712
Byte Offset: 0x31c48
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WORD_NUM_LOW**: The low 32 bits of the expect word number, used for BIST MODE 1.

NVCSI_PHY_2_NVCSI_CIL_CPHY_BIST_CONFIG_3_0

Offset: 0xc713
Byte Offset: 0x31c4c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WORD_NUM_HIGH**: The higher 16 bits of the expected word number, used for BIST MODE 1.

NVCSI_PHY_2_NVCSI_CIL_IDAC_CALIB_START_0

Offset: 0xc714
Byte Offset: 0x31c50
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**IDAC_CAL_TRIGGER**: Trigger the IDAC calibration.

NVCSI_PHY_2_NVCSI_CIL_IDAC_CALIB_CTRL_0

Offset: 0xc715
Byte Offset: 0x31c54
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,0000,xxxx,x000,0000,0101)

Bit	Reset	Description
19	0x0	**IDAC_CAL_EN_TRIO1_B**: Enable IDAC calibration for trio B1
18	0x0	**IDAC_CAL_EN_TRIO0_B**: Enable IDAC calibration for trio B0
17	0x0	**IDAC_CAL_EN_TRIO1_A**: Enable IDAC calibration for trio A1
16	0x0	**IDAC_CAL_EN_TRIO0_A**: Enable IDAC calibration for trio A0

Bit	Reset	Description
10:0	0x5	**IDAC_SETTLE_TIME**: How many cycle for settle to sample the done from pad.

NVCSI_PHY_2_NVCSI_CIL_IDAC_CALIB_STATUS_0

Offset: 0xc716
Byte Offset: 0x31c58
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,0000,0000,0000,xxxx,0000,0000,0000)

Bit	Reset	Description
27	0x0	**IDAC_CALIB_MAXOUT_RXCA_TRIO1_B**: IDAC calibration maxout for trio B1 RXCA.
26	0x0	**IDAC_CALIB_MAXOUT_RXBC_TRIO1_B**: IDAC calibration maxout for trio B1 RXBC.
25	0x0	**IDAC_CALIB_MAXOUT_RXAB_TRIO1_B**: IDAC calibration maxout for trio B1 RXAB.
24	0x0	**IDAC_CALIB_MAXOUT_RXCA_TRIO0_B**: IDAC calibration maxout for trio B0 RXCA.
23	0x0	**IDAC_CALIB_MAXOUT_RXBC_TRIO0_B**: IDAC calibration maxout for trio B0 RXBC.
22	0x0	**IDAC_CALIB_MAXOUT_RXAB_TRIO0_B**: IDAC calibration maxout for trio B0 RXAB.
21	0x0	**IDAC_CALIB_MAXOUT_RXCA_TRIO1_A**: IDAC calibration maxout for trio A1 RXCA.
20	0x0	**IDAC_CALIB_MAXOUT_RXBC_TRIO1_A**: IDAC calibration maxout for trio A1 RXBC.
19	0x0	**IDAC_CALIB_MAXOUT_RXAB_TRIO1_A**: IDAC calibration maxout for trio A1 RXAB.
18	0x0	**IDAC_CALIB_MAXOUT_RXCA_TRIO0_A**: IDAC calibration maxout for trio A0 RXCA.
17	0x0	**IDAC_CALIB_MAXOUT_RXBC_TRIO0_A**: IDAC calibration maxout for trio A0 RXBC.
16	0x0	**IDAC_CALIB_MAXOUT_RXAB_TRIO0_A**: IDAC calibration maxout for trio A0 RXAB.
11	0x0	**IDAC_CALIB_DONE_RXCA_TRIO1_B**: IDAC calibration done for trio B1 RXCA.

Bit	Reset	Description
10	0x0	**IDAC_CALIB_DONE_RXBC_TRIO1_B**: IDAC calibration done for trio B1 RXBC.
9	0x0	**IDAC_CALIB_DONE_RXAB_TRIO1_B**: IDAC calibration done for trio B1 RXAB.
8	0x0	**IDAC_CALIB_DONE_RXCA_TRIO0_B**: IDAC calibration done for trio B0 RXCA.
7	0x0	**IDAC_CALIB_DONE_RXBC_TRIO0_B**: IDAC calibration done for trio B0 RXBC.
6	0x0	**IDAC_CALIB_DONE_RXAB_TRIO0_B**: IDAC calibration done for trio B0 RXAB.
5	0x0	**IDAC_CALIB_DONE_RXCA_TRIO1_A**: IDAC calibration done for trio A1 RXCA.
4	0x0	**IDAC_CALIB_DONE_RXBC_TRIO1_A**: IDAC calibration done for trio A1 RXBC.
3	0x0	**IDAC_CALIB_DONE_RXAB_TRIO1_A**: IDAC calibration done for trio A1 RXAB.
2	0x0	**IDAC_CALIB_DONE_RXCA_TRIO0_A**: IDAC calibration done for trio A0 RXCA.
1	0x0	**IDAC_CALIB_DONE_RXBC_TRIO0_A**: IDAC calibration done for trio A0 RXBC.
0	0x0	**IDAC_CALIB_DONE_RXAB_TRIO0_A**: IDAC calibration done for trio A0 RXAB.

NVCSI_PHY_2_NVCSI_CIL_IDAC_CALIB_STATUS1_0

Offset: 0xc717
Byte Offset: 0x31c5c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x21084210 (0bxx10,0001,0000,1000,0100,0010,0001,0000)

Bit	Reset	Description
29:25	0x10	**RXCA_IDACADJ_TRIO1_A**: IDAC calibration result for trio A1 RXCA
24:20	0x10	**RXBC_IDACADJ_TRIO1_A**: IDAC calibration result for trio A1 RXBC
19:15	0x10	**RXAB_IDACADJ_TRIO1_A**: IDAC calibration result for trio A1 RXAB

Bit	Reset	Description
14:10	0x10	**RXCA_IDACADJ_TRIO0_A**: IDAC calibration result for trio A0 RXCA
9:5	0x10	**RXBC_IDACADJ_TRIO0_A**: IDAC calibration result for trio A0 RXBC
4:0	0x10	**RXAB_IDACADJ_TRIO0_A**: IDAC calibration result for trio A0 RXAB

NVCSI_PHY_2_NVCSI_CIL_IDAC_CALIB_STATUS2_0

Offset: 0xc718
Byte Offset: 0x31c60
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x21084210 (0bxx10,0001,0000,1000,0100,0010,0001,0000)

Bit	Reset	Description
29:25	0x10	**RXCA_IDACADJ_TRIO1_B**: IDAC calibration result for trio B1 RXCA
24:20	0x10	**RXBC_IDACADJ_TRIO1_B**: IDAC calibration result for trio B1 RXBC
19:15	0x10	**RXAB_IDACADJ_TRIO1_B**: IDAC calibration result for trio B1 RXAB
14:10	0x10	**RXCA_IDACADJ_TRIO0_B**: IDAC calibration result for trio B0 RXCA
9:5	0x10	**RXBC_IDACADJ_TRIO0_B**: IDAC calibration result for trio B0 RXBC
4:0	0x10	**RXAB_IDACADJ_TRIO0_B**: IDAC calibration result for trio B0 RXAB

NVCSI_PHY_2_NVCSI_CIL_IDAC_OVERRIDE_A_0

Offset: 0xc719
Byte Offset: 0x31c64
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x84218421 (0b1000,0100,0010,0001,1000,0100,0010,0001)

Bit	Reset	Description
31:27	0x10	**IDAC_VALUE_RXCA_TRIO1_A**: The override value
26:22	0x10	**IDAC_VALUE_RXBC_TRIO1_A**: The override value
21:17	0x10	**IDAC_VALUE_RXAB_TRIO1_A**: The override value
16	0x1	**SW_SET_IDAC_TRIO1_A**: Enable the SW override of the IDAC value
15:11	0x10	**IDAC_VALUE_RXCA_TRIO0_A**: The override value
10:6	0x10	**IDAC_VALUE_RXBC_TRIO0_A**: The override value
5:1	0x10	**IDAC_VALUE_RXAB_TRIO0_A**: The override value
0	0x1	**SW_SET_IDAC_TRIO0_A**: Enable the SW override of the IDAC value

NVCSI_PHY_2_NVCSI_CIL_IDAC_OVERRIDE_B_0

Offset: 0xc71a
Byte Offset: 0x31c68
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x84218421 (0b1000,0100,0010,0001,1000,0100,0010,0001)

Bit	Reset	Description
31:27	0x10	**IDAC_VALUE_RXCA_TRIO1_B**: The override value
26:22	0x10	**IDAC_VALUE_RXBC_TRIO1_B**: The override value
21:17	0x10	**IDAC_VALUE_RXAB_TRIO1_B**: The override value
16	0x1	**SW_SET_IDAC_TRIO1_B**: Enable the SW override of the IDAC value
15:11	0x10	**IDAC_VALUE_RXCA_TRIO0_B**: The override value

Bit	Reset	Description
10:6	0x10	**IDAC_VALUE_RXBC_TRIO0_B**: The override value
5:1	0x10	**IDAC_VALUE_RXAB_TRIO0_B**: The override value
0	0x1	**SW_SET_IDAC_TRIO0_B**: Enable the SW override of the IDAC value

NVCSI_PHY_2_NVCSI_CIL_IDAC_CALIB_DEBUG_0

Offset: 0xc71b
Byte Offset: 0x31c6c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000XXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
11	X	**ZI_HSRX_RXCA_TRIO1_B**
10	X	**ZI_HSRX_RXBC_TRIO1_B**
9	X	**ZI_HSRX_RXAB_TRIO1_B**
8	X	**ZI_HSRX_RXCA_TRIO0_B**
7	X	**ZI_HSRX_RXBC_TRIO0_B**
6	X	**ZI_HSRX_RXAB_TRIO0_B**
5	X	**ZI_HSRX_RXCA_TRIO1_A**
4	X	**ZI_HSRX_RXBC_TRIO1_A**
3	X	**ZI_HSRX_RXAB_TRIO1_A**
2	X	**ZI_HSRX_RXCA_TRIO0_A**
1	X	**ZI_HSRX_RXBC_TRIO0_A**
0	X	**ZI_HSRX_RXAB_TRIO0_A**

NVCSI_PHY_2_NVCSI_CIL_TEST_CONTROL_0

Offset: 0xc71c
Byte Offset: 0x31c70

Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**HS_EN_DATA_LANE1_B**: Force to enable the e_input_hs of the data lane b1, used for impedance measurements of HS mode
4	0x0	**HS_EN_DATA_LANE0_B**: Force to enable the e_input_hs of the data lane b0, used for impedance measurements of HS mode
3	0x0	**HS_EN_DATA_LANE1_A**: Force to enable the e_input_hs of the data lane a1, used for impedance measurements of HS mode
2	0x0	**HS_EN_DATA_LANE0_A**: Force to enable the e_input_hs of the data lane a0, used for impedance measurements of HS mode
1	0x0	**HS_EN_CLK_LANE_B**: Force to enable the e_input_hs of the clock lane b, used for impedance measurements of HS mode
0	0x0	**HS_EN_CLK_LANE_A**: Force to enable the e_input_hs of the clock lane a, used for impedance measurements of HS mode

NVCSI_PHY_2_NVCSI_CIL_PULLDN_CONTROL_0

Offset: 0xc71d
Byte Offset: 0x31c74
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**E_PULLDN_IO1_B**
4	0x0	**E_PULLDN_IO0_B**
3	0x0	**E_PULLDN_IO1_A**
2	0x0	**E_PULLDN_IO0_A**
1	0x0	**E_PULLDN_CLK_B**

Bit	Reset	Description
0	0x0	**E_PULLDN_CLK_A**

NVCSI_PHY_2_NVCSI_CIL_SPARE_0

Spare register
Offset: 0xc71e
Byte Offset: 0x31c78
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**phy_reg**

NVCSI_PHY_2_NVCSI_CIL_A_SW_RESET_0

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 CIL_A Registers
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Offset: 0xc71f
Byte Offset: 0x31c7c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**SW_RESET1_A**: SOFT RESET FOR LANE A1. Active High **1** = ENABLE **0** = DISABLE
0	0x0	**SW_RESET0_A**: SOFT RESET FOR LANE A0. Active High **1** = ENABLE **0** = DISABLE

NVCSI_PHY_2_NVCSI_CIL_A_CLKEN_OVERRIDE_CTRL_0

Offset: 0xc720
Byte Offset: 0x31c80
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**CLKEN_OVERRIDE1_A**: CLKENABLE OVERRIDE FOR LANE A1. Active High **1** = ENABLE **0** = DISABLE
0	0x0	**CLKEN_OVERRIDE0_A**: CLKENABLE OVERRIDE FOR LANE A0. Active High **1** = ENABLE **0** = DISABLE

NVCSI_PHY_2_NVCSI_CIL_A_CTLE_CTRL_0

Offset: 0xc721
Byte Offset: 0x31c84
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000888 (0bxxxx,xxxx,0000,0000,0000,1000,1000,1000)

Bit	Reset	Description
23:20	0x0	**AFE_DCGAIN_IO1_A**: RX AFE DC gain control
19:16	0x0	**AFE_DCGAIN_IO0_A**: RX AFE DC gain control
15:12	0x0	**AFE_DCGAIN_CLK_A**: RX AFE DC gain control
11:8	0x8	**AFE_HFGAIN_IO1_A**: RX AFE_CTLE high frequency gain control, 16 curves, ac_gain=code*1db

Bit	Reset	Description
7:4	0x8	**AFE_HFGAIN_IO0_A**: RX AFE_CTLE high frequency gain control, 16 curves, ac_gain=code*1db
3:0	0x8	**AFE_HFGAIN_CLK_A**: RX AFE_CTLE high frequency gain control, 16 curves, ac_gain=code*1db

NVCSI_PHY_2_NVCSI_CIL_A_CTLE_CTRL1_0

Offset: 0xc722
Byte Offset: 0x31c88
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000554 (0bxxxx,xxxx,0000,0000,0000,0101,0101,0100)

Bit	Reset	Description
23:20	0x0	**AFE_CTRL_IO1_A**
19:16	0x0	**AFE_CTRL_IO0_A**
15:12	0x0	**AFE_CTRL_CLK_A**
11:10	0x1	**AFE_CURRENT_IO1_A**
9:8	0x1	**AFE_CURRENT_IO0_A**
7:6	0x1	**AFE_CURRENT_CLK_A**
5:4	0x1	**AFE_FREQBAND_IO1_A**: RX AFE bandwidth selection. 2'b00: <1.5GS/s, 2'b01: 1.5~3GS/s, 2'b10: 3~5GS/s, 2'b11: reserved.
3:2	0x1	**AFE_FREQBAND_IO0_A**: RX AFE bandwidth selection. 2'b00: <1.5GS/s, 2'b01: 1.5~3GS/s, 2'b10: 3~5GS/s, 2'b11: reserved.
1:0	0x0	**AFE_FREQBAND_CLK_A**: RX AFE bandwidth selection. 2'b00: <1.5GS/s, 2'b01: 1.5~3GS/s, 2'b10: 3~5GS/s, 2'b11: reserved.

NVCSI_PHY_2_NVCSI_CIL_A_PAD_CONFIG_0

Offset: 0xc723
Byte Offset: 0x31c8c
Read/Write: R/W

Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00070000 (0b0000,0000,0000,x111,xxxx,0000,0000,0000)

Bit	Reset	Description
31	0x0	**LP_RX_SELECT_IO1_A**: LP mode receiver select for partition A lane 1. DPHY mode: controls IO1P_A and IO1N_A CPHY mode: controls IO1P_A, IO1N_A and IOCLKN_A.
30	0x0	**LP_RX_SELECT_IO0_A**: LP mode receiver select for partition A lane 0. DPHY mode: controls IO0P_A and IO0N_A CPHY mode: controls IO0P_A, IO0N_A and IOCLKP_A.
29	0x0	**LP_RX_SELECT_CLK_A**: LP mode receiver select. 0: Schmitt receiver. 1: self-biased receiver. DPHY mode: controls IOCLKP_A and IOCLKN_A. CPHY mode: dummy.
28	0x0	**E_LPRX_HYS_N_IO1_A**: Enable LP-RX hysteresis. 0: enable hysteresis 1: disable hysteresis
27	0x0	**E_LPRX_HYS_N_IO0_A**: Enable LP-RX hysteresis. 0: enable hysteresis 1: disable hysteresis
26	0x0	**E_LPRX_HYS_N_CLK_A**: Enable LP-RX hysteresis. 0: enable hysteresis 1: disable hysteresis
25	0x0	**E_HS_TERM_N_IO1_A**: Enable the HS termination of Partition A Lane 1.
24	0x0	**E_HS_TERM_N_IO0_A**: Enable the HS termination of Partition A Lane 0.
23	0x0	**E_HS_TERM_N_CLK_A**: Enable the HS termination of Clock partition A.
22	0x0	**E_INPUT_LP_IO1_A**: Enable LP receiver of Partition A Lane 1 Applicable in both CPHY and DPHY case.
21	0x0	**E_INPUT_LP_IO0_A**: Enable LP receiver of Partition A Lane 0 Applicable in both CPHY and DPHY case.
20	0x0	**E_INPUT_LP_CLK_A**: enable LP receiver of Clock partition A Applicable in DPHY case. N/A for CPHY
18	0x1	**PD_CLK_A**: Power down for CLk of Partition A. Applicable in DPHY case. N/A for CPHY

Bit	Reset	Description
17	0x1	**PD_IO1_A**: Power down for Trio 1 and Lane 1 of Partition A. Applicable in both CPHY and DPHY case.
16	0x1	**PD_IO0_A**: Power down for Trio 0 and Lane 0 of Partition A. Applicable in both CPHY and DPHY case.
11:8	0x0	**SPARE_CLK_A**: Spare control bits for pad control. Bit 11 is used as E_PULLDN_CLK_A control (0 - Pull down of Clock Lane enabled, 1 - Pull down of Clock Lane enabled). Functionality of other bits currently not defined. Undefined register bits are reserved for future use. Pull down may be enabled when Clock Lane is unused to minimized interference.
7:4	0x0	**SPARE_IO1_A**: Spare control bits for pad control. Functionality of other bits currently not defined. Undefined register bits are reserved for future use. Pull down may be enabled when Data Lane-1 is unused to minimized interference.
3:0	0x0	**SPARE_IO0_A**: Spare control bits for pad control. Functionality of other bits currently not defined. Undefined register bits are reserved for future use. Pull down may be enabled when Data Lane-0 is unused to minimized interference.

NVCSI_PHY_2_NVCSI_CIL_A_PAD_CONFIG_1_0

Offset: 0xc724
Byte Offset: 0x31c90
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000002 (0b0000,0000,0000,0000,0000,0000,xx00,0010)

Bit	Reset	Description
31:24	0x0	**OUTADJ_IO1_A**: Trio 1 output delay trimmer, each tap delays 9ps
23:16	0x0	**OUTADJ_IO0_A**: Trio 0 output delay trimmer, each tap delays 9ps
15:8	0x0	**OUTADJ_CLK_A**: Clock bit output delay trimmer, each tap delays 9ps
5	0x0	**HS_BSO_A**: 1= power-on 0.4v-TX-regulator during LP-mode; 0= power-off 0.4v-TX-regulator during LP-mode;

Bit	Reset	Description
4	0x0	**CPHY_MID_STRENGTH_TRIO1_A**: 0: when bump is driven to mid-voltage-level, both 100-ohm-pull-up and 100-ohm-pull-down is turned on to drive mid-voltage-level; 1: when bump is driven to mid-voltage-level, both pull-up and pull-down are turned off to become tri-stated;
3	0x0	**CPHY_MID_STRENGTH_TRIO0_A**: 0: when bump is driven to mid-voltage-level, both 100-ohm-pull-up and 100-ohm-pull-down is turned on to drive mid-voltage-level; 1: when bump is driven to mid-voltage-level, both pull-up and pull-down are turned off to become tri-stated;
2	DISABLE	**REV_CLK_A**: Reverse clock polarity **0** = DISABLE **1** = ENABLE
1	ENABLE	**E_PREDRV_FLOP_RESET_A**: HS predriver flop reset **0** = DISABLE **1** = ENABLE
0	DISABLE	**FCZERO_A**: Force clk bit to drive differential 0 **0** = DISABLE **1** = ENABLE

NVCSI_PHY_2_NVCSI_CIL_A_PAD_CONFIG_2_0

Offset: 0xc725
Byte Offset: 0x31c94
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,0000,0000,0000,xxxx,0000,0000,0000)

Bit	Reset	Description
27:24	0x0	**PEMPD_IO1_A**: Enable trio 1 HS driver pull down pre-emphasis.
23:20	0x0	**PEMPD_IO0_A**: Enable trio 0 HS driver pull down pre-emphasis.
19:16	0x0	**PEMPD_CLK_A**: Enable clock bit HS driver pull down pre-emphasis.
11:8	0x0	**PEMPU_IO1_A**: Enable trio 1 HS driver pull up pre-emphasis.

Bit	Reset	Description
7:4	0x0	**PEMPU_IO0_A**: Enable trio 0 HS driver pull up pre-emphasis.
3:0	0x0	**PEMPU_CLK_A**: Enable clock bit HS driver pull up pre-emphasis.

NVCSI_PHY_2_NVCSI_CIL_A_PAD_CONFIG_3_0

Offset: 0xc726
Byte Offset: 0x31c98
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x06660666 (0bxxxx,0110,0110,0110,xxxx,0110,0110,0110)

Bit	Reset	Description
27:24	0x6	**LPDNADJ_IO1_A**: Driver pull down impedance control
23:20	0x6	**LPDNADJ_IO0_A**: Driver pull down impedance control
19:16	0x6	**LPDNADJ_CLK_A**: Driver pull down impedance control
11:8	0x6	**LPUPADJ_IO1_A**: Driver pull up impedance control
7:4	0x6	**LPUPADJ_IO0_A**: Driver pull up impedance control
3:0	0x6	**LPUPADJ_CLK_A**: Driver pull up impedance control

NVCSI_PHY_2_NVCSI_CIL_A_PAD_CONFIG_4_0

Offset: 0xc727
Byte Offset: 0x31c9c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,0000,0000,0000,xxxx,0000,0000,0000)

Bit	Reset	Description
27:24	0x0	**SLEWDNADJ_IO1_A**: Pull down slew rate adjust
23:20	0x0	**SLEWDNADJ_IO0_A**: Pull down slew rate adjust
19:16	0x0	**SLEWDNADJ_CLK_A**: Pull down slew rate adjust
11:8	0x0	**SLEWUPADJ_IO1_A**: Pull up slew rate adjust
7:4	0x0	**SLEWUPADJ_IO0_A**: Pull up slew rate adjust
3:0	0x0	**SLEWUPADJ_CLK_A**: Pull up slew rate adjust

NVCSI_PHY_2_NVCSI_CIL_A_PAD_CONFIG_5_0

Offset: 0xc728
Byte Offset: 0x31ca0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000110 (0bxxxx,xxxx,xxxx,xxxx,xxxx,x001,0001,0000)

Bit	Reset	Description
10:7	0x2	**CDDNADJ_IO1_A**: Level adjust on low limit of detection
6:3	0x2	**CDDNADJ_IO0_A**: Level adjust on low limit of detection
2	DISABLE	**E_CD_IO1_A**: Data bits contention detector enable **0** = DISABLE **1** = ENABLE
1	DISABLE	**E_CD_IO0_A**: Data bits contention detector enable **0** = DISABLE **1** = ENABLE
0	DISABLE	**E_CD_CLK_A**: Clock bit contention detector enable **0** = DISABLE **1** = ENABLE

NVCSI_PHY_2_NVCSI_CIL_A_PAD_CONFIG_6_0

Offset: 0xc729
Byte Offset: 0x31ca4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**E_DEEP_LPBK_A**: Enable deep loopback
0	0x0	**E_SHALLOW_LPBK_A**: Enalbe shallow loopback

NVCSI_PHY_2_NVCSI_CIL_A_PAD_CD_STATUS_0

Offset: 0xc72a
Byte Offset: 0x31ca8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**CDN_IO1_A**: trio1 n output, 1=contention detected.
4	0x0	**CDN_IO0_A**: trio0 n output, 1=contention detected.
3	0x0	**CDN_CLK_A**: Clock bit n output, 1=contention detected.
2	0x0	**CDP_IO1_A**: trio1 p output, 1=contention detected.
1	0x0	**CDP_IO0_A**: trio0 p output, 1=contention detected.
0	0x0	**CDP_CLK_A**: Clock bit p output, 1=contention detected.

NVCSI_PHY_2_NVCSI_CIL_A_DPHY_INADJ_CTRL_0

Offset: 0xc72b
Byte Offset: 0x31cac
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,x000,0000,x000,0000,x000,0000)

Bit	Reset	Description
22	0x0	**SW_SET_DPHY_INADJ_CLK_A**: Software set for clock input delay trimmer
21:16	0x0	**DPHY_INADJ_CLK_A**: Programmable value for CLK input delay trimmer,
14	0x0	**SW_SET_DPHY_INADJ_IO1_A**: Software override for bit 1 input delay trimmer 1 - SW programmed value is applied to the pads. 2 - Trimmer value from HW calibration state machine is applied to the pads
13:8	0x0	**DPHY_INADJ_IO1_A**: Programmable value for bit 1 input delay trimmer, each tap delay of 9 ps
6	0x0	**SW_SET_DPHY_INADJ_IO0_A**: Software override for bit 0 input delay trimmer 1 - SW programmed value is applied to the pads. 2 - Trimmer value from HW calibration state machine is applied to the pads
5:0	0x0	**DPHY_INADJ_IO0_A**: Programmable value for bit 0 input delay trimmer,

NVCSI_PHY_2_NVCSI_CIL_A_CLK_DESKEW_CTRL_0

Offset: 0xc72c
Byte Offset: 0x31cb0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000bf00 (0bxxxx,xxxx,xxxx,xxxx,1x11,1111,xx00,0000)

Bit	Reset	Description
15	ENABLE	**CLK_INADJ_SWEEP_CTRL_A**: Enable the clock trimmer sweep for D-PHY de-skew. **0** = DISABLE **1** = ENABLE
13:8	0x3f	**CLK_INADJ_LIMIT_HIGH_A**: The upper limit of the sweep range for trimmer sweep.
5:0	0x0	**CLK_INADJ_LIMIT_LOW_A**: The lower limit of the sweep range for trimmer sweep.

NVCSI_PHY_2_NVCSI_CIL_A_DATA_DESKEW_CTRL_0

Offset: 0xc72d
Byte Offset: 0x31cb4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x9f80bf00 (0b1xx1,1111,1x00,0000,1x11,1111,xx00,0000)

Bit	Reset	Description
31	0x1	**DATA_INADJ_SWEEP_CTRL1_A**: Enable the data lane A1 trimmer sweep for D-PHY de-skew.
28:23	0x3f	**DATA_INADJ_LIMIT_HIGH1_A**: The upper limit of the sweep range for trimmer sweep.
21:16	0x0	**DATA_INADJ_LIMIT_LOW1_A**: The lower limit of the sweep range for trimmer sweep.
15	0x1	**DATA_INADJ_SWEEP_CTRL0_A**: Enable the data lane A0 trimmer sweep for D-PHY de-skew.
13:8	0x3f	**DATA_INADJ_LIMIT_HIGH0_A**: The upper limit of the sweep range for trimmer sweep.
5:0	0x0	**DATA_INADJ_LIMIT_LOW0_A**: The lower limit of the sweep range for trimmer sweep.

NVCSI_PHY_2_NVCSI_CIL_A_DPHY_DESKEW_STATUS_0

Offset: 0xc72e
Byte Offset: 0x31cb8
Read/Write: RO
Parity Protection: N
Shadow: N

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000X0XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
15	X	**DPHY_CALIB_ERR_IO1_A**: calib error status
14	X	**DPHY_CALIB_DONE_IO1_A**: calib done status
7	X	**DPHY_CALIB_ERR_IO0_A**: calib error status
6	X	**DPHY_CALIB_DONE_IO0_A**: calib done status
1	X	**DPHY_CALIB_ERR_CTRL_A**: calib error status
0	X	**DPHY_CALIB_DONE_CTRL_A**: calib done status

NVCSI_PHY_2_NVCSI_CIL_A_DPHY_DESKEW_DATA_CALIB_STATUS_LOW_0_0

Offset: 0xc72f
Byte Offset: 0x31cbc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_A**

NVCSI_PHY_2_NVCSI_CIL_A_DPHY_DESKEW_DATA_CALIB_STATUS_HIGH_0_0

Offset: 0xc730
Byte Offset: 0x31cc0
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_A**

NVCSI_PHY_2_NVCSI_CIL_A_DPHY_DESKEW_DATA_CALIB_STATUS_LOW_1_0

Offset: 0xc731
Byte Offset: 0x31cc4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_A**

NVCSI_PHY_2_NVCSI_CIL_A_DPHY_DESKEW_DATA_CALIB_STATUS_HIGH_1_0

Offset: 0xc732
Byte Offset: 0x31cc8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_A**

NVCSI_PHY_2_NVCSI_CIL_A_DPHY_DESKEW_CLK_CALIB_STATUS_LOW_0_0

Offset: 0xc733
Byte Offset: 0x31ccc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_A**

NVCSI_PHY_2_NVCSI_CIL_A_DPHY_DESKEW_CLK_CALIB_STATUS_HIGH_0_0

Offset: 0xc734
Byte Offset: 0x31cd0
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_A**

NVCSI_PHY_2_NVCSI_CIL_A_DPHY_DESKEW_CLK_CALIB_STATUS_LOW_1_0

Offset: 0xc735
Byte Offset: 0x31cd4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_A**

NVCSI_PHY_2_NVCSI_CIL_A_DPHY_DESKEW_CLK_CALIB_STATUS_HIGH_1_0

Offset: 0xc736
Byte Offset: 0x31cd8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_A**

NVCSI_PHY_2_NVCSI_CIL_A_DPHY_DESKEW_RESULT_STATUS_0

Offset: 0xc737
Byte Offset: 0x31cdc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	DONE	**DESKEW_RESULT_STATUS1_A**: For lane A1. The result status of the deskew calibration, HW check this field before start the trimmer sweep, if it's 1'b1, the trimmer sweep will not perform. HW set this bit to 1 when the calibration is done. SW should clear this field after it read out the 2 64 bits calibration status. **0** = DONE **1** = PENDING
0	DONE	**DESKEW_RESULT_STATUS0_A**: For lane A0. The result status of the deskew calibration, HW check this field before start the trimmer sweep, if it's 1'b1, the trimmer sweep will not perform. HW set this bit to 1 when the calibration is done. SW should clear this field after it read out the 2 64 bits calibration status. **0** = DONE **1** = PENDING

NVCSI_PHY_2_NVCSI_CIL_A_POLARITY_SWIZZLE_CTRL_0

Offset: 0xc738
Byte Offset: 0x31ce0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xx00,0000,xxxx,x000)

Bit	Reset	Description
13:11	0x0	**POLARITY_SWIZZLE_CPHY1_A**: Polarity Swizle control for Lane A1 in CPHY mode. valid only in CPHY mode 000 - A B C --> A B C 001 - A B C --> A C B 010 - A B C --> B C A 011 - A B C --> B A C 100 - A B C --> C A B 101 - A B C --> C B A
10:8	0x0	**POLARITY_SWIZZLE_CPHY0_A**: Polarity Swizle control for Lane A0 in CPHY mode. valid only in CPHY mode 000 - A B C --> A B C 001 - A B C --> A C B 010 - A B C --> B C A 011 - A B C --> B A C 100 - A B C --> C A B 101 - A B C --> C B A
2	0x0	**POLARITY_SWIZZLE_CLK_A**: Polarity Swizzle control for clock lane A. Valid only in DPHY mode, need to work with ALLOW_FIRST_BIT_ON_CLK_NEGEDGE if enabled.
1	0x0	**POLARITY_SWIZZLE_DPHY1_A**: Polarity Swizzle control for data lane A1. Valid only in DPHY mode.
0	0x0	**POLARITY_SWIZZLE_DPHY0_A**: Polarity Swizzle control for data lane A0. Valid only in DPHY mode.

NVCSI_PHY_2_NVCSI_CIL_A_DESKEW_CONTROL_0

Offset: 0xc739
Byte Offset: 0x31ce4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00007411 (0bxxxx,xxxx,xxxx,xxxx,0111,0100,0001,0001)

Bit	Reset	Description
15:13	0x3	**DESKEW_DATA_DELAY**: How many cycles delay introduced by re-timing flops
12	0x1	**DESKEW_LANE_SKEW_CHECK_EN**: Check the skew between the lanes, this is used to handle the error case that error happen in the CSI link caused some data lanes detect the HS sync word (0xb8) while other data lanes detect the deskew sync word (0xff). If this bit is not set with the error case, the lane FIFO will overflow and reset is required. With this bit set, the FIFO will be auto flused and the FIFO overflow will not happen.
11:8	0x4	**DESKEW_LANE_SKEW_TOLERANCE**: The allowed skew between the data lane, only used in sync word search

Bit	Reset	Description
7:4	0x1	**DESKEW_COMPARE**: Regsiter select to control the number of comparisons to be done for each trimmer setting during deskew calibration.
3:0	0x1	**DESKEW_SETTLE**: Regsiter select to control the number of byte clocks to wait before INADJ value settles. This is used during deskew calibration

NVCSI_PHY_2_NVCSI_CIL_A_CONTROL_0

Offset: 0xc73a
Byte Offset: 0x31ce8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**BYPASS_LP_FILTER**: Bypass the glitch filter.
30:24	0x0	**CLK_SETTLE1_A**: Used only in CPHY mode, Settle time for clk start when moving A1 data lane from LP to HS (LP11->LP01->LP00).
23:17	0x0	**CLK_SETTLE0_A**: Used for both CPHY/DPY mode. For DPHY: Settle time for clk lane A0 when moving from LP to HS (LP11->LP01->LP00), this setting determines how many csicil clock cycles to wait after LP00. If the settle set to 0 or 8, the real setting is 1, otherwise, the real setting is (4+settle) For CPHY: Settle time for clk start when moving A0 data lane from LP to HS (LP11->LP01->LP00).
16:9	0x0	**THS_SETTLE1_A**: settle time for A1 data lane when moving from LP to HS (LP11->LP01->LP00), this setting determines how many LP clock cycles to wait, after LP00,before starting to look at the data.
8:1	0x0	**THS_SETTLE0_A**: settle time for A0 data lane when moving from LP to HS (LP11->LP01->LP00), this setting determines how many LP clock cycles to wait, after LP00,before starting to look at the data.

Bit	Reset	Description
0	0x0	**BYPASS_LP_SEQ**: For DPHY: The clock lane LP signals should sequence through LP11->LP01->LP00 state, to indicate to CLOCK CIL about the mode switching to HS Rx mode. In case Camera is enabled earlier than CIL , it is highly likely that camera sends this control sequence sooner than cil can detect it. Enabling this bit allows the CLOCK CIL to overlook the LP control sequence and step in HS Rx mode directly looking at LP00 only. For CPHY: it's only used for debug purpose as it's embedded clock for CPHY, clock will always presents when the data lane toggle.

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_ERR_STATUS_0

Offset: 0xc73b
Byte Offset: 0x31cec
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
29:24	X	**LANE_DEMAPPER_ERR_MUX1_A**: Debug register, error detected in CPHY de-mapper mux.
23	X	**LANE_DEMAPPER_ERR_RXCTRL1_A**: Debug register, Error detected in CPHY de-mapper rx ctrl.
22:16	X	**LANE_DECODER_ERR1_A**: Debug register, Error detected in CPHY de-coder.
13:8	X	**LANE_DEMAPPER_ERR_MUX0_A**: Debug register, Error detected in CPHY de-mapper mux.
7	X	**LANE_DEMAPPER_ERR_RXCTRL0_A**: Debug register, Error detected in CPHY de-mapper rx ctrl.
6:0	X	**LANE_DECODER_ERR0_A**: Debug register, Error detected in CPHY de-coder.

NVCSI_PHY_2_NVCSI_CIL_A_ESCAPE_MODE_COMMAND_0_0

Offset: 0xc73c
Byte Offset: 0x31cf0
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_COMMAND0_A**: The received command from escape mode.

NVCSI_PHY_2_NVCSI_CIL_A_ESCAPE_MODE_DATA_0_0

Offset: 0xc73d
Byte Offset: 0x31cf4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_DATA0_A**: The received data from excape mode.

NVCSI_PHY_2_NVCSI_CIL_A_ESCAPE_MODE_COMMAND_1_0

Offset: 0xc73e
Byte Offset: 0x31cf8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_COMMAND1_A**: The received command from escape mode.

NVCSI_PHY_2_NVCSI_CIL_A_ESCAPE_MODE_DATA_1_0

Offset: 0xc73f
Byte Offset: 0x31cfc
Read/Write: RO
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_DATA1_A**: The received data from excape mode.

NVCSI_PHY_2_NVCSI_CIL_A_DPHY_SYNC_PATTERN_0

Offset: 0xc740
Byte Offset: 0x31d00
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01)

Bit	Reset	Description
1	0x0	**ALLOW_FIRST_BIT_ON_CLK_NEGEDGE_A**: Allow shift 1/3/5/7 bit to search the sync word. Normal case only shift 0/2/4/6 bit to search sync word.
0	0x1	**DISABLE_SB_ERR_IN_SYNC_A**: Allow one single bit error in sync word 6MSB

NVCSI_PHY_2_NVCSI_CIL_A_DPHY_DESKEW_SYNC_PATTERN_0

Offset: 0xc741
Byte Offset: 0x31d04
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**INVERT_DESKEW_PATTERN**: Invert the de-skew pattern.

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_SYNC_SEARCH_0

Offset: 0xc742
Byte Offset: 0x31d08
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01080108 (0bxxxx,xxx1,0000,1000,xxxx,xxx1,0000,1000)

Bit	Reset	Description
24	0x1	**COUNT_EN_TRIO1_A**: Check if the symbol is pre-amble before sync word.
23:16	0x8	**PERIOD_TRIO1_A**: For CPHY, how many cycles allowed for non pre-amble symbol before sync word.
8	ENABLE	**COUNT_EN_TRIO0_A**: Check if the symbol is pre-amble before sync word. **0** = DISABLE **1** = ENABLE
7:0	0x8	**PERIOD_TRIO0_A**: For CPHY, how many cycles allowed for non pre-amble symbol before sync word.

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_START_0

Offset: 0xc743
Byte Offset: 0x31d0c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**EDGE_DELAY_CAL_TRIGGER_TRIO1_A**: Trigger the CPHY clock recover calibration for lane A1 by write 1'b1, when the calibration complete, HW will clear this bit to 1'b0. Write 1'b0 to this be take no effect.
0	0x0	**EDGE_DELAY_CAL_TRIGGER_TRIO0_A**: Trigger the CPHY clock recover calibration for lane A0 by write 1'b1, when the calibration complete, HW will clear this bit to 1'b0. Write 1'b0 to this be take no effect.

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_CTRL_0

Offset: 0xc744
Byte Offset: 0x31d10
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x12011210 (0b0001,0010,0000,0001,0001,0010,0001,0000)
PROD: 0x08000800 (0bxxx0,10xx,xxxx,xxxx,xxx0,10xx,xxxx,xxxx)

Bit	Reset	PROD	Description
31	0x0	_NONE_	**TRIM_CAL_E_NONSTOP_TRIO1_A**: Reserved. Trimmer calibration background monitoring control. 1: enable. The accumulator keeps updating.
30:29	0x0	_NONE_	**TRIM_CAL_GAIN_SEL_TRIO1_A**: Trimmer calibration loop gain selection [0]: gain control 0: coarse tune (counter1) gain = 1, fine tune (counter2) gain = 1/4. For example: coarse tune: 1 UP -> trim_cal_code + 1; fine tune: 4 UPs -> trim_cal_code + 1. 1: coarse tune (counter1) gain = 1/2, fine tune (counter2) gain = 1/8 [1]: reserved
28:26	0x4	0x2	**TRIM_CAL_CNT_SEL_TRIO1_A**: Trimmer calibration counter selection [0]: counter1 (coarse tune) clock cycle. 0: 64T (128UI) 1: 128T (256UI) [2:1]: counter2 (fine tune) clock cycle. 00: 32T (64UI) 01: 64T (128UI) 10: 96T (192UI) 11: 128T (256UI) For high data rate (>3.5GS/s): 64T(cnt1)+96T(cnt2)=160T(320ui) -> cnt_sel=3'b100 For low data rate (<=3.5GS/s): 128T(cnt1)+32T(cnt2)=160T(320ui) -> cnt_se=3'b001
25	0x1	_NONE_	**TRIM_CAL_CPHY_ED_LOOPMODE_TRIO1_A**: Trimmer calibration and cphy edge detector loop mode control 0: exclude trimmer delay in 2nd round of reset loop. 1: include trimmer delay in 2nd round of reset loop. For low data rate.
24:16	0x1	_NONE_	**CALIB_LENGTH_TRIO1_A**: //The length of the peramble which use to do caliration. Count by nvcsi core clock. It's used to make sure the calibration is doing inside the preamble phase. If this length set to a small value, the calibration may can't complete within one packet and need several packets to complete.

Bit	Reset	PROD	Description
15	0x0	_NONE_	**TRIM_CAL_E_NONSTOP_TRIO0_A**: Reserved. Trimmer calibration background monitoring control. 1: enable. The accumulator keeps updating.
14:13	0x0	_NONE_	**TRIM_CAL_GAIN_SEL_TRIO0_A**: Trimmer calibration loop gain selection [0]: gain control 0: coarse tune (counter1) gain = 1, fine tune (counter2) gain = 1/4. For example: coarse tune: 1 UP -> trim_cal_code + 1; fine tune: 4 UPs -> trim_cal_code + 1. 1: coarse tune (counter1) gain = 1/2, fine tune (counter2) gain = 1/8 [1]: reserved
12:10	0x4	0x2	**TRIM_CAL_CNT_SEL_TRIO0_A**: Trimmer calibration counter selection [0]: counter1 (coarse tune) clock cycle. 0: 64T (128UI) 1: 128T (256UI) [2:1]: counter2 (fine tune) clock cycle. 00: 32T (64UI) 01: 64T (128UI) 10: 96T (192UI) 11: 128T (256UI) For high data rate (>3.5GS/s): 64T(cnt1)+96T(cnt2)=160T(320ui) -> cnt_sel=3'b100 For low data rate (<=3.5GS/s): 128T(cnt1)+32T(cnt2)=160T(320ui) -> cnt_se=3'b001
9	0x1	_NONE_	**TRIM_CAL_CPHY_ED_LOOPMODE_TRIO0_A**: Trimmer calibration and cphy edge detector loop mode control 0: exclude trimmer delay in 2nd round of reset loop. 1: include trimmer delay in 2nd round of reset loop. For low data rate.
8:0	0x10	_NONE_	**CALIB_LENGTH_TRIO0_A**: The length of the peramble which use to do caliration. Count by nvcsi core clock. It's used to make sure the calibration is doing inside the preamble phase. If this length set to a small value, the calibration may can't complete within one packet and need several packets to complete.

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_CTRL1_0

Offset: 0xc745
Byte Offset: 0x31d14
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0c0e0c0e (0bx000,1100,0000,1110,x000,1100,0000,1110)

Bit	Reset	Description
30	0x0	**PERIODIC_CALIB_TRIO1_A**: Enable the periodic calibration.
29	0x0	**CALIB_APPLY_IN_POST_TRIO1_A**: 1: Apply the calibration result when detect post symbol. 0: Apply the calibration result immediately when calibration done.
28:27	0x1	**CALIB_APPLY_SELECT_TRIO1_A**: Which calibration result is applied to pad for dual calibration. 2'b11: Reserved 2'b10: Apply the middle value of two calibration to pad if the difference within threshold. 2'b01: Apply the 2nd calibration result to pad if it's difference with previous latch result within threshold. 2'b00: Apply the 1st calibration result to pad if it's difference with previous latch result within threshold.
26	0x1	**DUAL_CALIB_TRIO1_A**: Two times calibratiton. 1: Two time calibration with one trigger. 0: Single time calibration with one trigger
25:18	0x3	**TEMP_EDGE_DELAY_VALUE_TRIO1_A**: The edge delay value while doing the calibration.
17	0x1	**TEMP_EDGE_DELAY_SEL_TRIO1_A**: When the selection of edge delay value when do edge delay calibration. 1: Use the temp value from register setting. 0: Use the currently using setting. For the first calibration after reset, it's still using the temp value from register setting.
16	0x0	**BYPASS_CALIB_PACKET_TRIO1_A**: 1: Drop the packet data when do the calibration. 0: Not drop the packet data when do the calibration.
14	0x0	**PERIODIC_CALIB_TRIO0_A**: Enable the periodic calibration.
13	0x0	**CALIB_APPLY_IN_POST_TRIO0_A**: 1: Apply the calibration result when detect post symbol. 0: Apply the calibration result immediately when calibration done.
12:11	0x1	**CALIB_APPLY_SELECT_TRIO0_A**: Which calibration result is applied to pad for dual calibration. 2'b11: Reserved 2'b10: Apply the middle value of two calibration to pad if the difference within threshold. 2'b01: Apply the 2nd calibration result to pad if it's difference with previous latch result within threshold. 2'b00: Apply the 1st calibration result to pad if it's difference with previous latch result within threshold.
10	0x1	**DUAL_CALIB_TRIO0_A**: Two times calibratiton. 1: Two time calibration with one trigger. 0: Single time calibration with one trigger
9:2	0x3	**TEMP_EDGE_DELAY_VALUE_TRIO0_A**: The edge delay value apply to pad while doing the calibration.

Bit	Reset	Description
1	0x1	**TEMP_EDGE_DELAY_SEL_TRIO0_A**: When the selection of edge delay value when do edge delay calibration. 1: Use the temp value from register setting. 0: Use the currently using setting. For the first calibration after reset, it's still using the temp value from register setting.
0	0x0	**BYPASS_CALIB_PACKET_TRIO0_A**: 1: Drop the packet data when do the calibration. 0: Not drop the packet data when do the calibration.

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_CTRL2_0

Offset: 0xc746
Byte Offset: 0x31d18
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01000100 (0bxxxx,xxx1,0000,0000,xxxx,xxx1,0000,0000)

Bit	Reset	Description
24	ENABLE	**TRIM_CAL_PD_HOLD_TRIO1_A**: Hold in PD **0** = DISABLE **1** = ENABLE
23	DISABLE	**TRIM_CAL_E_FIXEDDLY_TRIO1_A**: Enable fixed delay in trim calibration unit. **0** = DISABLE **1** = ENABLE
22:16	0x0	**TRIM_CAL_FIXEDDLY_TRIO1_A**: Adjust fixed delay. Delay = code*5ps(TT)/7ps(SS)
8	ENABLE	**TRIM_CAL_PD_HOLD_TRIO0_A**: Hold in PD **0** = DISABLE **1** = ENABLE
7	DISABLE	**TRIM_CAL_E_FIXEDDLY_TRIO0_A**: Enable fixed delay in trim calibration unit. **0** = DISABLE **1** = ENABLE
6:0	0x0	**TRIM_CAL_FIXEDDLY_TRIO0_A**: Adjust fixed delay. Delay = code*5ps(TT)/7ps(SS)

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_CTRL3_0

Offset: 0xc747
Byte Offset: 0x31d1c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffffffff	**CALIB_PERIOD_TRIO0_A**: The period of the periodic edge delay calibration, in nvcsi clock cycle number.

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_CTRL4_0

Offset: 0xc748
Byte Offset: 0x31d20
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffffffff	**CALIB_PERIOD_TRIO1_A**: The period of the periodic edge delay calibration, in nvcsi clock cycle number.

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_CTRL5_0

Offset: 0xc749
Byte Offset: 0x31d24
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000ffff (0bxxxx,xxxx,xxxx,xxxx,1111,1111,1111,1111)

Bit	Reset	Description
15:8	0xff	**ERR_THRESHOLD_TRIO1_A**: The threshold of drop the calibration result between current result and previous result. For dual calibration, check between the two calibration result in the same trigger. For signal calibration, check between the current calibration result and previous calibration result.
7:0	0xff	**ERR_THRESHOLD_TRIO0_A**: The threshold of drop the calibration result between current result and previous resulti. For dual calibration, check between the two calibration result in the same trigger. For signal calibration, check between the current calibration result and previous calibration result.

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_STATUS_0

Offset: 0xc74a
Byte Offset: 0x31d28
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,xxxx)

Bit	Reset	Description
9	0x0	**EDGE_DELAY_MISMATCH_TRIO1_A**: When the CALIB_APPLY_SELECT_TRIO is 0/1 or dual time calibration is not enabled, the calibration result difference from the previous latched calibration result is larger than threshold, this time calibration result is not apply to pad. When the CALIB_APPLY_SELECT_TRIO is 2, the calibration result difference of the two times calibration result is larger than threshold, this time calibration result is not apply to pad.
8	0x0	**EDGE_DELAY_MISMATCH_TRIO0_A**: When the CALIB_APPLY_SELECT_TRIO is 0/1 or dual time calibration is not enabled, the calibration result difference from the previous latched calibration result is larger than threshold, this time calibration result is not apply to pad. When the CALIB_APPLY_SELECT_TRIO is 2, the calibration result difference of the two times calibration result is larger than threshold, this time calibration result is not apply to pad.
7	0x0	**CALIB_DONE_1_TRIO1_A**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.
6	0x0	**CALIB_DONE_1_TRIO0_A**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.
5	0x0	**CALIB_DONE_0_TRIO1_A**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.

Bit	Reset	Description
4	0x0	**CALIB_DONE_0_TRIO0_A**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_EDGE_DELAY_CTRL_0

Offset: 0xc74b
Byte Offset: 0x31d2c
Read/Write: R/W
Parity Protection: See table below
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x40000000 (0b01xx,x000,0000,0000,xxxx,x000,0000,0000)

Bit	Parity Protection	Reset	Description
31:30	Y	0x1	**CPHY_CLKGEN_TRIGWIDTH_A**: Adjust AB/CB/CA pulse width in self-clock generator.
26	N	0x0	**UPDATE_OFFSET_TRIO1_A**: Trigger to apply the new offset value
25:17	Y	0x0	**DELAY_OFFSET_TRIO1_A**: Offset value of trio A1. Represents signed 2's complement number for calibration offset. Represents unsigned number if the field used as override value.
16	Y	0x0	**OVERRIDE_CAL_VAL_TRIO1_A**: Override for trio A1 0: Apply the DELAY_OFFSET as an offset to the calibration result. 1: Apply the DELAY_OFFSET value to the pad.
10	N	0x0	**UPDATE_OFFSET_TRIO0_A**: Trigger to apply the new offset value
9:1	Y	0x0	**DELAY_OFFSET_TRIO0_A**: Offset value of trio A0. Represents signed 2's complement number for calibration offset. Represents unsigned number if the field used as override value.
0	Y	0x0	**OVERRIDE_CAL_VAL_TRIO0_A**: Override for trio A0 0: Apply the DELAY_OFFSET as an offset to the calibration result. 1: Apply the DELAY_OFFSET value to the pad.

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_EDGE_DELAY_VALUE_0

Offset: 0xc74c
Byte Offset: 0x31d30
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000303 (0bxxxx,xxxx,xxxx,xxxx,0000,0011,0000,0011)

Bit	Reset	Description
15:8	0x3	**EDGE_DELAY_VALUE_TRIO1_A**: The edge delay value apply to pad finally
7:0	0x3	**EDGE_DELAY_VALUE_TRIO0_A**: The edge delay value apply to pad finally

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_RESULT_0

Offset: 0xc74d
Byte Offset: 0x31d34
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**EDGE_DELAY_CALIB_RESULT_1_TRIO1_A**: The edge delay calibrate result, with offset.
23:16	0x0	**EDGE_DELAY_CALIB_RESULT_1_TRIO0_A**: The edge delay calibrate result, with offset.
15:8	0x0	**EDGE_DELAY_CALIB_RESULT_0_TRIO1_A**: The edge delay calibrate result, with offset.
7:0	0x0	**EDGE_DELAY_CALIB_RESULT_0_TRIO0_A**: The edge delay calibrate result, with offset.

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_INADJ_CALIB_CTRL_0

Offset: 0xc74e
Byte Offset: 0x31d38
Read/Write: R/W
Parity Protection: N
Shadow: N

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
16	0x0	**INADJ_CAL_TRIGGER_TRIO1_A**: Trigger INADJ calibration for trio0
0	0x0	**INADJ_CAL_TRIGGER_TRIO0_A**: Trigger INADJ calibration for trio1

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_INADJ_CALIB_CTRL1_0

Offset: 0xc74f
Byte Offset: 0x31d3c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00010001 (0b0000,0000,0000,0001,0000,0000,0000,0001)

Bit	Reset	Description
31:16	0x1	**PRBS9_SEED_TRIO1_A**: PRBS9 initial seed for trio0
15:0	0x1	**PRBS9_SEED_TRIO0_A**: PRBS9 initial seed for trio1

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_INADJ_CALIB_CTRL2_0

Offset: 0xc750
Byte Offset: 0x31d40
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0fc00fc0 (0bxxxx,1111,1100,0000,xxxx,1111,1100,0000)

Bit	Reset	Description
27:22	0x3f	**INADJ_LIMIT_HIGH_TRIO1_A**: The upper limit for the INADJ sweep.
21:16	0x0	**INADJ_LIMIT_LOW_TRIO1_A**: The lower limit for the INADJ sweep.

Bit	Reset	Description
11:6	0x3f	**INADJ_LIMIT_HIGH_TRIO0_A**: The upper limit for the INADJ sweep.
5:0	0x0	**INADJ_LIMIT_LOW_TRIO0_A**: The lower limit for the INADJ sweep.

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_INADJ_CALIB_STATUS_0

Offset: 0xc751
Byte Offset: 0x31d44
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**INADJ_CALIB_DONE_TRIO1_A**: Trio a1 inadj calibration done.
0	0x0	**INADJ_CALIB_DONE_TRIO0_A**: Trio a0 inadj calibration done.

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_INADJ_CALIB_STATUS1_0

Offset: 0xc752
Byte Offset: 0x31d48
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_LOW_TRIO0_A**: Trio a0 PRBS check status for lower 32 INADJ value.

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_INADJ_CALIB_STATUS2_0

Offset: 0xc753
Byte Offset: 0x31d4c

Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_LOW_TRIO1_A**: Trio a1 PRBS check status for lower 32 INADJ value.

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_INADJ_CALIB_STATUS3_0

Offset: 0xc754
Byte Offset: 0x31d50
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_HIGH_TRIO0_A**: Trio a0 PRBS check status for upper 32 INADJ value.

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_INADJ_CALIB_STATUS4_0

Offset: 0xc755
Byte Offset: 0x31d54
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_HIGH_TRIO1_A**: Trio a1 PRBS check status for upper 32 INADJ value.

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_INADJ_OVERRIDE_0

Offset: 0xc756
Byte Offset: 0x31d58
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00080008 (0bxxxx,xxx0,0000,1000,xxxx,xxx0,0000,1000)

Bit	Reset	Description
24:17	0x4	**CPHY_INADJ_TRIO1_A**: The SW override INADJ value.
16	DISABLE	**SW_SET_CPHY_INADJ_TRIO1_A**: Enable SW override of the INADJ. **0** = DISABLE **1** = ENABLE
8:1	0x4	**CPHY_INADJ_TRIO0_A**: The SW override INADJ value.
0	DISABLE	**SW_SET_CPHY_INADJ_TRIO0_A**: Enable SW override of the INADJ. **0** = DISABLE **1** = ENABLE

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_BIST_CTRL_0

Offset: 0xc757
Byte Offset: 0x31d5c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	0x0	**STOP_BIST_TRIO1_A**: Manually stop the A1 BIST, write 1 to stop the RX bist.
2	0x0	**STOP_BIST_TRIO0_A**: Manually stop the A0 BIST, write 1 to stop the RX bist.
1	0x0	**TRIGGER_BIST_TRIO1_A**: Trigger the trio A1 BIST, write 1 to trigger the RX bist, HW will clear it to 0 when the BIST is done, write 0 to this field is with no effect.

Bit	Reset	Description
0	0x0	**TRIGGER_BIST_TRIO0_A**: Trigger the trio A0 BIST, write 1 to trigger the RX bist, HW will clear it to 0 when the BIST is done, write 0 to this field is with no effect.

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_BIST_STATUS_0_0

Offset: 0xc758
Byte Offset: 0x31d60
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:8	0x0	**WORD_ERR_CNT_TRIO1_A**: The trio A1 number of word comparison failure, report for BIST mode 1.
7:0	0x0	**WORD_ERR_CNT_TRIO0_A**: The trio A0 number of word comparison failure, report for BIST mode 1.

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_BIST_STATUS_1_0

Offset: 0xc759
Byte Offset: 0x31d64
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:8	0x0	**SYM_ERR_CNT_TRIO1_A**: The trio A1 number of symbol compariosn failure, it's the sum of the symbol err count and the clock shift count, the real wire state error number is 1/2 of this field, report for BIST mode 1.
7:0	0x0	**SYM_ERR_CNT_TRIO0_A**: The trio A0 number of symbol compariosn failure, it's the sum of the symbol err count and the clock shift count, the real wire state error number is 1/2 of this field, report for BIST mode 1.

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_BIST_STATUS_2_0

Offset: 0xc75a
Byte Offset: 0x31d68
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WORD_CNT_LOW_TRIO0_A**: The trio A0 low 32 bits of the compared word number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_BIST_STATUS_3_0

Offset: 0xc75b
Byte Offset: 0x31d6c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WORD_CNT_HIGH_TRIO0_A**: The trio A0 high 16 bits of the compared word number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_BIST_STATUS_4_0

Offset: 0xc75c
Byte Offset: 0x31d70
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WORD_CNT_LOW_TRIO1_A**: The trio A1 low 32 bits of the compared word number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_BIST_STATUS_5_0

Offset: 0xc75d
Byte Offset: 0x31d74
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WORD_CNT_HIGH_TRIO1_A**: The trio A1 high 16 bits of the compared word number number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_2_NVCSI_CIL_A_CPHY_BIST_STATUS_6_0

Offset: 0xc75e
Byte Offset: 0x31d78
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,x000,0000,0000,xxxx,x000,0000,0000)

Bit	Reset	Description
26	0x0	**ERR_DETECT_FAIL_TRIO1_A**: Multi symbol error in one word (7 symbol) for trio A1, BIST logic is not able to detect the accurate wire state error. With this error, if not other errors detected, the word error is accurate.
25	0x0	**SHIFT_DETECT_FAIL_TRIO1_A**: Detect symbol error in two continuous word (14 symbols) for trio A1, BIST logic is not able to determine if symbol clock is slip. (if the symbol error count is not saturating, the result is still available with this error)
24	0x0	**SHIFT_OVERFLOW_TRIO1_A**: Too many clocks slip and the BIST logic is not able to re-align for trio A1. Typically, when this error is detect, the symbol error counter will saturate.

Bit	Reset	Description
23:20	0x0	**DOUBLE_CLK_MISSED_CNT_TRIO1_A**: Number of the times for double symbol clock lost for trio A1.
19:16	0x0	**SINGLE_CLK_MISSED_CNT_TRIO1_A**: Number of the times for single symbol clock lost for trio A1.
10	0x0	**ERR_DETECT_FAIL_TRIO0_A**: Multi symbol error in one word (7 symbol) for trio A0, BIST logic is not able to detect the accurate wire state error. With this error, if not other errors detected, the word error is accurate.
9	0x0	**SHIFT_DETECT_FAIL_TRIO0_A**: Detect symbol error in two continuous word (14 symbols) for trio A0, BIST logic is not able to determine if symbol clock is slip. (if the symbol error count is not saturating, the result is still available with this error)
8	0x0	**SHIFT_OVERFLOW_TRIO0_A**: Too many clocks slip and the BIST logic is not able to re-align for trio A0. Typically, when this error is detect, the symbol error counter will saturate.
7:4	0x0	**DOUBLE_CLK_MISSED_CNT_TRIO0_A**: Number of the times for double symbol clock lost for trio A0.
3:0	0x0	**SINGLE_CLK_MISSED_CNT_TRIO0_A**: Number of the times for single symbol clock lost for trio A0.

NVCSI_PHY_2_NVCSI_CIL_B_SW_RESET_0

Offset: 0xc75f
Byte Offset: 0x31d7c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**SW_RESET1_B**: SOFT RESET FOR LANE B1. Active High **1** = ENABLE **0** = DISABLE
0	0x0	**SW_RESET0_B**: SOFT RESET FOR LANE B0. Active High **1** = ENABLE **0** = DISABLE

NVCSI_PHY_2_NVCSI_CIL_B_CLKEN_OVERRIDE_CTRL_0

Offset: 0xc760
Byte Offset: 0x31d80
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**CLKEN_OVERRIDE1_B**: CLKENABLE OVERRIDE FOR LANE B1. Active High **1** = ENABLE **0** = DISABLE
0	0x0	**CLKEN_OVERRIDE0_B**: CLKENABLE OVERRIDE FOR LANE B0. Active High **1** = ENABLE **0** = DISABLE

NVCSI_PHY_2_NVCSI_CIL_B_CTLE_CTRL_0

Offset: 0xc761
Byte Offset: 0x31d84
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000888 (0bxxxx,xxxx,0000,0000,0000,1000,1000,1000)

Bit	Reset	Description
23:20	0x0	**AFE_DCGAIN_IO1_B**: RX AFE DC gain control
19:16	0x0	**AFE_DCGAIN_IO0_B**: RX AFE DC gain control
15:12	0x0	**AFE_DCGAIN_CLK_B**: RX AFE DC gain control
11:8	0x8	**AFE_HFGAIN_IO1_B**: RX AFE_CTLE high frequency gain control, 16 curves, ac_gain=code*1db

Bit	Reset	Description
7:4	0x8	**AFE_HFGAIN_IO0_B**: RX AFE_CTLE high frequency gain control, 16 curves, ac_gain=code*1db
3:0	0x8	**AFE_HFGAIN_CLK_B**: RX AFE_CTLE high frequency gain control, 16 curves, ac_gain=code*1db

NVCSI_PHY_2_NVCSI_CIL_B_CTLE_CTRL1_0

Offset: 0xc762
Byte Offset: 0x31d88
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000554 (0bxxxx,xxxx,0000,0000,0000,0101,0101,0100)

Bit	Reset	Description
23:20	0x0	**AFE_CTRL_IO1_B**
19:16	0x0	**AFE_CTRL_IO0_B**
15:12	0x0	**AFE_CTRL_CLK_B**
11:10	0x1	**AFE_CURRENT_IO1_B**
9:8	0x1	**AFE_CURRENT_IO0_B**
7:6	0x1	**AFE_CURRENT_CLK_B**
5:4	0x1	**AFE_FREQBAND_IO1_B**: RX AFE bandwidth selection. 2'b00: <1.5GS/s, 2'b01: 1.5~3GS/s, 2'b10: 3~5GS/s, 2'b11: reserved.
3:2	0x1	**AFE_FREQBAND_IO0_B**: RX AFE bandwidth selection. 2'b00: <1.5GS/s, 2'b01: 1.5~3GS/s, 2'b10: 3~5GS/s, 2'b11: reserved.
1:0	0x0	**AFE_FREQBAND_CLK_B**: RX AFE bandwidth selection. 2'b00: <1.5GS/s, 2'b01: 1.5~3GS/s, 2'b10: 3~5GS/s, 2'b11: reserved.

NVCSI_PHY_2_NVCSI_CIL_B_PAD_CONFIG_0

Offset: 0xc763
Byte Offset: 0x31d8c
Read/Write: R/W

Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00070000 (0b0000,0000,0000,x111,xxxx,0000,0000,0000)

Bit	Reset	Description
31	0x0	**LP_RX_SELECT_IO1_B**: LP mode receiver select for partition B lane 1. DPHY mode: controls IO1P_B and IO1N_B CPHY mode: controls IO1P_B, IO1N_B and IOCLKN_B.
30	0x0	**LP_RX_SELECT_IO0_B**: LP mode receiver select for partition B lane 0. DPHY mode: controls IO0P_B and IO0N_B CPHY mode: controls IO0P_B, IO0N_B and IOCLKP_B.
29	0x0	**LP_RX_SELECT_CLK_B**: LP mode receiver select. 0: Schmitt receiver. 1: self-biased receiver. DPHY mode: controls IOCLKP_B and IOCLKN_B. CPHY mode: dummy.
28	0x0	**E_LPRX_HYS_N_IO1_B**: Enable LP-RX hysteresis. 0: enable hysteresis 1: disable hysteresis
27	0x0	**E_LPRX_HYS_N_IO0_B**: Enable LP-RX hysteresis. 0: enable hysteresis 1: disable hysteresis
26	0x0	**E_LPRX_HYS_N_CLK_B**: Enable LP-RX hysteresis. 0: enable hysteresis 1: disable hysteresis
25	0x0	**E_HS_TERM_N_IO1_B**: Enable the HS termination of Partition B Lane 1.
24	0x0	**E_HS_TERM_N_IO0_B**: Enable the HS termination of Partition B Lane 0.
23	0x0	**E_HS_TERM_N_CLK_B**: Enable the HS termination of Clock partition B.
22	0x0	**E_INPUT_LP_IO1_B**: Enable LP receiver of Partition B Lane 1 Applicable in both CPHY and DPHY case.
21	0x0	**E_INPUT_LP_IO0_B**: Enable LP receiver of Partition B Lane 0 Applicable in both CPHY and DPHY case.
20	0x0	**E_INPUT_LP_CLK_B**: enable LP receiver of Clock partition B Applicable in DPHY case. N/A for CPHY
18	0x1	**PD_CLK_B**: Power down for CLk of Partition B. Applicable in DPHY case. N/A for CPHY

Bit	Reset	Description
17	0x1	**PD_IO1_B**: Power down for Trio 1 and Lane 1 of Partition B. Applicable in both CPHY and DPHY case.
16	0x1	**PD_IO0_B**: Power down for Trio 0 and Lane 0 of Partition B. Applicable in both CPHY and DPHY case.
11:8	0x0	**SPARE_CLK_B**: Spare control bits for pad control. Functionality of other bits currently not defined. Undefined register bits are reserved for future use. Pull down may be enabled when Clock Lane is unused to minimized interference.
7:4	0x0	**SPARE_IO1_B**: Spare control bits for pad control. Functionality of other bits currently not defined. Undefined register bits are reserved for future use. Pull down may be enabled when Data Lane-1 is unused to minimized interference.
3:0	0x0	**SPARE_IO0_B**: Spare control bits for pad control. Functionality of other bits currently not defined. Undefined register bits are reserved for future use. Pull down may be enabled when Data Lane-0 is unused to minimized interference.

NVCSI_PHY_2_NVCSI_CIL_B_PAD_CONFIG_1_0

Offset: 0xc764
Byte Offset: 0x31d90
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000002 (0b0000,0000,0000,0000,0000,0000,xx00,0010)

Bit	Reset	Description
31:24	0x0	**OUTADJ_IO1_B**: Trio 1 output delay trimmer, each tap delays 9ps
23:16	0x0	**OUTADJ_IO0_B**: Trio 0 output delay trimmer, each tap delays 9ps
15:8	0x0	**OUTADJ_CLK_B**: Clock bit output delay trimmer, each tap delays 9ps
5	0x0	**HS_BSO_B**: 1= power-on 0.4v-TX-regulator during LP-mode; 0= power-off 0.4v-TX-regulator during LP-mode;
4	0x0	**CPHY_MID_STRENGTH_TRIO1_B**: 0: when bump is driven to mid-voltage-level, both 100-ohm-pull-up and 100-ohm-pull-down is turned on to drive mid-voltage-level; 1: when bump is driven to mid-voltage-level, both pull-up and pull-down are turned off to become tri-stated;

Bit	Reset	Description
3	0x0	**CPHY_MID_STRENGTH_TRIO0_B**: 0: when bump is driven to mid-voltage-level, both 100-ohm-pull-up and 100-ohm-pull-down is turned on to drive mid-voltage-level; 1: when bump is driven to mid-voltage-level, both pull-up and pull-down are turned off to become tri-stated;
2	DISABLE	**REV_CLK_B**: Reverse clock polarity **0** = DISABLE **1** = ENABLE
1	ENABLE	**E_PREDRV_FLOP_RESET_B**: HS predriver flop reset **0** = DISABLE **1** = ENABLE
0	DISABLE	**FCZERO_B**: Force clk bit to drive differential 0 **0** = DISABLE **1** = ENABLE

NVCSI_PHY_2_NVCSI_CIL_B_PAD_CONFIG_2_0

Offset: 0xc765
Byte Offset: 0x31d94
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,0000,0000,0000,xxxx,0000,0000,0000)

Bit	Reset	Description
27:24	0x0	**PEMPD_IO1_B**: Enable trio 1 HS driver pull down pre-emphasis.
23:20	0x0	**PEMPD_IO0_B**: Enable trio 0 HS driver pull down pre-emphasis.
19:16	0x0	**PEMPD_CLK_B**: Enable clock bit HS driver pull down pre-emphasis.
11:8	0x0	**PEMPU_IO1_B**: Enable trio 1 HS driver pull up pre-emphasis.
7:4	0x0	**PEMPU_IO0_B**: Enable trio 0 HS driver pull up pre-emphasis.
3:0	0x0	**PEMPU_CLK_B**: Enable clock bit HS driver pull up pre-emphasis.

NVCSI_PHY_2_NVCSI_CIL_B_PAD_CONFIG_3_0

Offset: 0xc766
Byte Offset: 0x31d98
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x06660666 (0bxxxx,0110,0110,0110,xxxx,0110,0110,0110)

Bit	Reset	Description
27:24	0x6	**LPDNADJ_IO1_B**: Driver pull down impedance control
23:20	0x6	**LPDNADJ_IO0_B**: Driver pull down impedance control
19:16	0x6	**LPDNADJ_CLK_B**: Driver pull down impedance control
11:8	0x6	**LPUPADJ_IO1_B**: Driver pull up impedance control
7:4	0x6	**LPUPADJ_IO0_B**: Driver pull up impedance control
3:0	0x6	**LPUPADJ_CLK_B**: Driver pull up impedance control

NVCSI_PHY_2_NVCSI_CIL_B_PAD_CONFIG_4_0

Offset: 0xc767
Byte Offset: 0x31d9c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,0000,0000,0000,xxxx,0000,0000,0000)

Bit	Reset	Description
27:24	0x0	**SLEWDNADJ_IO1_B**: Pull down slew rate adjust
23:20	0x0	**SLEWDNADJ_IO0_B**: Pull down slew rate adjust
19:16	0x0	**SLEWDNADJ_CLK_B**: Pull down slew rate adjust

Bit	Reset	Description
11:8	0x0	**SLEWUPADJ_IO1_B**: Pull up slew rate adjust
7:4	0x0	**SLEWUPADJ_IO0_B**: Pull up slew rate adjust
3:0	0x0	**SLEWUPADJ_CLK_B**: Pull up slew rate adjust

NVCSI_PHY_2_NVCSI_CIL_B_PAD_CONFIG_5_0

Offset: 0xc768
Byte Offset: 0x31da0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000110 (0bxxxx,xxxx,xxxx,xxxx,xxxx,x001,0001,0000)

Bit	Reset	Description
10:7	0x2	**CDDNADJ_IO1_B**: Level adjust on low limit of detection
6:3	0x2	**CDDNADJ_IO0_B**: Level adjust on low limit of detection
2	DISABLE	**E_CD_IO1_B**: Data bits contention detector enable **0** = DISABLE **1** = ENABLE
1	DISABLE	**E_CD_IO0_B**: Data bits contention detector enable **0** = DISABLE **1** = ENABLE
0	DISABLE	**E_CD_CLK_B**: Clock bit contention detector enable **0** = DISABLE **1** = ENABLE

NVCSI_PHY_2_NVCSI_CIL_B_PAD_CONFIG_6_0

Offset: 0xc769
Byte Offset: 0x31da4
Read/Write: R/W

Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**E_DEEP_LPBK_B**: Enable deep loopback
0	0x0	**E_SHALLOW_LPBK_B**: Enalbe shallow loopback

NVCSI_PHY_2_NVCSI_CIL_B_PAD_CD_STATUS_0

Offset: 0xc76a
Byte Offset: 0x31da8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**CDN_IO1_B**: trio1 n output, 1=contention detected.
4	0x0	**CDN_IO0_B**: trio0 n output, 1=contention detected.
3	0x0	**CDN_CLK_B**: Clock bit n output, 1=contention detected.
2	0x0	**CDP_IO1_B**: trio1 p output, 1=contention detected.
1	0x0	**CDP_IO0_B**: trio0 p output, 1=contention detected.
0	0x0	**CDP_CLK_B**: Clock bit p output, 1=contention detected.

NVCSI_PHY_2_NVCSI_CIL_B_DPHY_INADJ_CTRL_0

Offset: 0xc76b
Byte Offset: 0x31dac
Read/Write: R/W
Parity Protection: Y

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,x000,0000,x000,0000,x000,0000)

Bit	Reset	Description
22	0x0	**SW_SET_DPHY_INADJ_CLK_B**: Software set for clock input delay trimmer
21:16	0x0	**DPHY_INADJ_CLK_B**: Programmable value for CLK input delay trimmer,
14	0x0	**SW_SET_DPHY_INADJ_IO1_B**: Software override for bit 1 input delay trimmer 1 - SW programmed value is applied to the pads. 2 - Trimmer value from HW calibration state machine is applied to the pads
13:8	0x0	**DPHY_INADJ_IO1_B**: Programmable value for bit 1 input delay trimmer, each tap delay of 9 ps
6	0x0	**SW_SET_DPHY_INADJ_IO0_B**: Software override for bit 0 input delay trimmer 1 - SW programmed value is applied to the pads. 2 - Trimmer value from HW calibration state machine is applied to the pads
5:0	0x0	**DPHY_INADJ_IO0_B**: Programmable value for bit 0 input delay trimmer,

NVCSI_PHY_2_NVCSI_CIL_B_CLK_DESKEW_CTRL_0

Offset: 0xc76c
Byte Offset: 0x31db0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000bf00 (0bxxxx,xxxx,xxxx,xxxx,1x11,1111,xx00,0000)

Bit	Reset	Description
15	ENABLE	**CLK_INADJ_SWEEP_CTRL_B**: Enable the clock trimmer sweep for D-PHY de-skew. **0** = DISABLE **1** = ENABLE
13:8	0x3f	**CLK_INADJ_LIMIT_HIGH_B**: The upper limit of the sweep range for trimmer sweep.
5:0	0x0	**CLK_INADJ_LIMIT_LOW_B**: The lower limit of the sweep range for trimmer sweep.

NVCSI_PHY_2_NVCSI_CIL_B_DATA_DESKEW_CTRL_0

Offset: 0xc76d
Byte Offset: 0x31db4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x9f80bf00 (0b1xx1,1111,1x00,0000,1x11,1111,xx00,0000)

Bit	Reset	Description
31	0x1	**DATA_INADJ_SWEEP_CTRL1_B**: Enable the data lane B1 trimmer sweep for D-PHY de-skew.
28:23	0x3f	**DATA_INADJ_LIMIT_HIGH1_B**: The upper limit of the sweep range for trimmer sweep.
21:16	0x0	**DATA_INADJ_LIMIT_LOW1_B**: The lower limit of the sweep range for trimmer sweep.
15	0x1	**DATA_INADJ_SWEEP_CTRL0_B**: Enable the data lane B0 trimmer sweep for D-PHY de-skew.
13:8	0x3f	**DATA_INADJ_LIMIT_HIGH0_B**: The upper limit of the sweep range for trimmer sweep.
5:0	0x0	**DATA_INADJ_LIMIT_LOW0_B**: The lower limit of the sweep range for trimmer sweep.

NVCSI_PHY_2_NVCSI_CIL_B_DPHY_DESKEW_STATUS_0

Offset: 0xc76e
Byte Offset: 0x31db8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000X0XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
15	X	**DPHY_CALIB_ERR_IO1_B**: calib error status
14	X	**DPHY_CALIB_DONE_IO1_B**: calib done status
7	X	**DPHY_CALIB_ERR_IO0_B**: calib error status

Bit	Reset	Description
6	X	**DPHY_CALIB_DONE_IO0_B**: calib done status
1	X	**DPHY_CALIB_ERR_CTRL_B**: calib error status
0	X	**DPHY_CALIB_DONE_CTRL_B**: calib done status

NVCSI_PHY_2_NVCSI_CIL_B_DPHY_DESKEW_DATA_CALIB_STATUS_LOW_0_0

Offset: 0xc76f
Byte Offset: 0x31dbc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_B**

NVCSI_PHY_2_NVCSI_CIL_B_DPHY_DESKEW_DATA_CALIB_STATUS_HIGH_0_0

Offset: 0xc770
Byte Offset: 0x31dc0
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_B**

NVCSI_PHY_2_NVCSI_CIL_B_DPHY_DESKEW_DATA_CALIB_STATUS_LOW_1_0

Offset: 0xc771
Byte Offset: 0x31dc4
Read/Write: RO
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_B**

NVCSI_PHY_2_NVCSI_CIL_B_DPHY_DESKEW_DATA_CALIB_STATUS_HIGH_1_0

Offset: 0xc772
Byte Offset: 0x31dc8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_B**

NVCSI_PHY_2_NVCSI_CIL_B_DPHY_DESKEW_CLK_CALIB_STATUS_LOW_0_0

Offset: 0xc773
Byte Offset: 0x31dcc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_B**

NVCSI_PHY_2_NVCSI_CIL_B_DPHY_DESKEW_CLK_CALIB_STATUS_HIGH_0_0

Offset: 0xc774
Byte Offset: 0x31dd0
Read/Write: RO
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_B**

NVCSI_PHY_2_NVCSI_CIL_B_DPHY_DESKEW_CLK_CALIB_STATUS_LOW_1_0

Offset: 0xc775
Byte Offset: 0x31dd4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_B**

NVCSI_PHY_2_NVCSI_CIL_B_DPHY_DESKEW_CLK_CALIB_STATUS_HIGH_1_0

Offset: 0xc776
Byte Offset: 0x31dd8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_B**

NVCSI_PHY_2_NVCSI_CIL_B_DPHY_DESKEW_RESULT_STATUS_0

Offset: 0xc777
Byte Offset: 0x31ddc
Read/Write: R/W
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	DONE	**DESKEW_RESULT_STATUS1_B**: For lane B1. The result status of the deskew calibration, HW check this field before start the trimmer sweep, if it's 1'b1, the trimmer sweep will not perform. HW set this bit to 1 when the calibration is done. SW should clear this field after it read out the 2 64 bits calibration status. **0** = DONE **1** = PENDING
0	DONE	**DESKEW_RESULT_STATUS0_B**: For lane B0. The result status of the deskew calibration, HW check this field before start the trimmer sweep, if it's 1'b1, the trimmer sweep will not perform. HW set this bit to 1 when the calibration is done. SW should clear this field after it read out the 2 64 bits calibration status. **0** = DONE **1** = PENDING

NVCSI_PHY_2_NVCSI_CIL_B_POLARITY_SWIZZLE_CTRL_0

Offset: 0xc778
Byte Offset: 0x31de0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xx00,0000,xxxx,x000)

Bit	Reset	Description
13:11	0x0	**POLARITY_SWIZZLE_CPHY1_B**: Polarity Swizle control for Lane B1 in CPHY mode. valid only in CPHY mode 000 - A B C --> A B C 001 - A B C --> A C B 010 - A B C --> B C A 011 - A B C --> B A C 100 - A B C --> C A B 101 - A B C --> C B A
10:8	0x0	**POLARITY_SWIZZLE_CPHY0_B**: Polarity Swizle control for Lane B0 in CPHY mode. valid only in CPHY mode 000 - A B C --> A B C 001 - A B C --> A C B 010 - A B C --> B C A 011 - A B C --> B A C 100 - A B C --> C A B 101 - A B C --> C B A

Bit	Reset	Description
2	0x0	**POLARITY_SWIZZLE_CLK_B**: Polarity Swizzle control for clock lane B. Valid only in DPHY mode, need to work with ALLOW_FIRST_BIT_ON_CLK_NEGEDGE if enabled.
1	0x0	**POLARITY_SWIZZLE_DPHY1_B**: Polarity Swizzle control for data lane B1. register bit valid only in DPHY mode.
0	0x0	**POLARITY_SWIZZLE_DPHY0_B**: Polarity Swizzle control for data lane B0. register bit valid only in DPHY mode.

NVCSI_PHY_2_NVCSI_CIL_B_DESKEW_CONTROL_0

Offset: 0xc779
Byte Offset: 0x31de4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00007411 (0bxxxx,xxxx,xxxx,xxxx,0111,0100,0001,0001)

Bit	Reset	Description
15:13	0x3	**DESKEW_DATA_DELAY**: How many cycles delay introduced by re-timing flops
12	0x1	**DESKEW_LANE_SKEW_CHECK_EN**: Check the skew between the lanes. this is used to handle the error case that error happen in the CSI link caused some data lanes detect the HS sync word (0xb8) while other data lanes detect the deskew sync word (0xff). If this bit is not set with the error case, the lane FIFO will overflow and reset is required. With this bit set, the FIFO will be auto flused and the FIFO overflow will not happen.
11:8	0x4	**DESKEW_LANE_SKEW_TOLERANCE**: The allowed skew between the data lane, only used in sync word search
7:4	0x1	**DESKEW_COMPARE**: Regsiter select to control the number of comparisons to be done for each trimmer setting during deskew calibration.
3:0	0x1	**DESKEW_SETTLE**: Regsiter select to control the number of byte clocks to wait before INADJ value settles. This is used during deskew calibration

NVCSI_PHY_2_NVCSI_CIL_B_CONTROL_0

Offset: 0xc77a
Byte Offset: 0x31de8
Read/Write: R/W

Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**BYPASS_LP_FILTER**: Bypass the glitch filter.
30:24	0x0	**CLK_SETTLE1_B**: Used only in CPHY mode, Settle time for clk start when moving B1 data lane from LP to HS (LP11->LP01->LP00).
23:17	0x0	**CLK_SETTLE0_B**: Used for both CPHY/DPY mode. For DPHY: Settle time for clk lane when moving from LP to HS (LP11->LP01->LP00), this setting determines how many csicil clock cycles to wait after LP00. If the settle set to 0 or 8, the real setting is 1, otherwise, the real setting is (4+settle) For CPHY: Settle time for clk start when moving B0 data lane from LP to HS (LP11->LP01->LP00).
16:9	0x0	**THS_SETTLE1_B**: settle time for B1 data lane when moving from LP to HS (LP11->LP01->LP00), this setting determines how many LP clock cycles to wait, after LP00, before starting to look at the data.
8:1	0x0	**THS_SETTLE0_B**: settle time for B0 data lane when moving from LP to HS (LP11->LP01->LP00), this setting determines how many LP clock cycles to wait, after LP00, before starting to look at the data.
0	0x0	**BYPASS_LP_SEQ**: For DPHY: The clock lane LP signals should sequence through LP11->LP01->LP00 state, to indicate to CLOCK CIL about the mode switching to HS Rx mode. In case Camera is enabled earlier than CIL , it is highly likely that camera sends this control sequence sooner than cil can detect it. Enabling this bit allows the CLOCK CIL to overlook the LP control sequence and step in HS Rx mode directly looking at LP00 only. For CPHY: it's only used for debug purpose as it's embedded clock for CPHY, clock will always presents when the data lane toggle.

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_ERR_STATUS_0

Offset: 0xc77b
Byte Offset: 0x31dec
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
29:24	X	**LANE_DEMAPPER_ERR_MUX1_B**: Debug register, error detected in CPHY de-mapper mux.
23	X	**LANE_DEMAPPER_ERR_RXCTRL1_B**: Debug register, error detected in CPHY de-mapper rx ctrl.
22:16	X	**LANE_DECODER_ERR1_B**: Debug register, error detected in CPHY de-coder.
13:8	X	**LANE_DEMAPPER_ERR_MUX0_B**: Debug register, error detected in CPHY de-mapper mux.
7	X	**LANE_DEMAPPER_ERR_RXCTRL0_B**: Debug register, error detected in CPHY de-mapper rx ctrl.
6:0	X	**LANE_DECODER_ERR0_B**: Debug register, error detected in CPHY de-coder.

NVCSI_PHY_2_NVCSI_CIL_B_ESCAPE_MODE_COMMAND_0_0

Offset: 0xc77c
Byte Offset: 0x31df0
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_COMMAND0_B**: The received command from escape mode.

NVCSI_PHY_2_NVCSI_CIL_B_ESCAPE_MODE_DATA_0_0

Offset: 0xc77d
Byte Offset: 0x31df4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_DATA0_B**: The received data from excape mode.

NVCSI_PHY_2_NVCSI_CIL_B_ESCAPE_MODE_COMMAND_1_0

Offset: 0xc77e
Byte Offset: 0x31df8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_COMMAND1_B**: The received command from escape mode.

NVCSI_PHY_2_NVCSI_CIL_B_ESCAPE_MODE_DATA_1_0

Offset: 0xc77f
Byte Offset: 0x31dfc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_DATA1_B**: The received data from excape mode.

NVCSI_PHY_2_NVCSI_CIL_B_DPHY_SYNC_PATTERN_0

Offset: 0xc780
Byte Offset: 0x31e00
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01)

Bit	Reset	Description
1	0x0	**ALLOW_FIRST_BIT_ON_CLK_NEGEDGE_B**: Allow shift 1/3/5/7 bit to search the sync word. Normal case only shift 0/2/4/6 bit to search sync word.
0	0x1	**DISABLE_SB_ERR_IN_SYNC_B**: Allow one single bit error in sync word 6MSB.

NVCSI_PHY_2_NVCSI_CIL_B_DPHY_DESKEW_SYNC_PATTERN_0

Offset: 0xc781
Byte Offset: 0x31e04
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**INVERT_DESKEW_PATTERN**: Invert the de-skew pattern.

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_SYNC_SEARCH_0

Offset: 0xc782
Byte Offset: 0x31e08
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01080108 (0bxxxx,xxx1,0000,1000,xxxx,xxx1,0000,1000)

Bit	Reset	Description
24	0x1	**COUNT_EN_TRIO1_B**: Check if the symbol is pre-amble before sync word.
23:16	0x8	**PERIOD_TRIO1_B**: For CPHY, how many cycles allowed for non pre-amble symbol before sync word.
8	0x1	**COUNT_EN_TRIO0_B**: Check if the symbol is pre-amble before sync word.

Bit	Reset	Description
7:0	0x8	**PERIOD_TRIO0_B**: For CPHY, how many cycles allowed for non pre-amble symbol before sync word.

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_START_0

Offset: 0xc783
Byte Offset: 0x31e0c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**EDGE_DELAY_CAL_TRIGGER_TRIO1_B**: Trigger the CPHY clock recover calibration for lane B1 by write 1'b1, when the calibration complete, HW will clear this bit to 1'b0. Write 1'b0 to this be take no effect.
0	0x0	**EDGE_DELAY_CAL_TRIGGER_TRIO0_B**: Trigger the CPHY clock recover calibration for lane B0 by write 1'b1, when the calibration complete, HW will clear this bit to 1'b0. Write 1'b0 to this be take no effect.

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_CTRL_0

Offset: 0xc784
Byte Offset: 0x31e10
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x12011201 (0b0001,0010,0000,0001,0001,0010,0000,0001)
PROD: 0x08000800 (0bxxx0,10xx,xxxx,xxxx,xxx0,10xx,xxxx,xxxx)

Bit	Reset	PROD	Description
31	0x0	_NONE_	**TRIM_CAL_E_NONSTOP_TRIO1_B**: Reserved. Trimmer calibration background monitoring control. 1: enable. The accumulator keeps updating.

Bit	Reset	PROD	Description
30:29	0x0	_NONE_	**TRIM_CAL_GAIN_SEL_TRIO1_B**: Trimmer calibration loop gain selection [0]: gain control 0: coarse tune (counter1) gain = 1, fine tune (counter2) gain = 1/4. For example: coarse tune: 1 UP -> trim_cal_code + 1; fine tune: 4 UPs -> trim_cal_code + 1. 1: coarse tune (counter1) gain = 1/2, fine tune (counter2) gain = 1/8 [1]: reserved
28:26	0x4	0x2	**TRIM_CAL_CNT_SEL_TRIO1_B**: Trimmer calibration counter selection [0]: counter1 (coarse tune) clock cycle. 0: 64T (128UI) 1: 128T (256UI) [2:1]: counter2 (fine tune) clock cycle. 00: 32T (64UI) 01: 64T (128UI) 10: 96T (192UI) 11: 128T (256UI) For high data rate (>3.5GS/s): 64T(cnt1)+96T(cnt2)=160T(320ui) -> cnt_sel=3'b100 For low data rate (<=3.5GS/s): 128T(cnt1)+32T(cnt2)=160T(320ui) -> cnt_se=3'b001
25	0x1	_NONE_	**TRIM_CAL_CPHY_ED_LOOPMODE_TRIO1_B**: Trimmer calibration and cphy edge detector loop mode control 0: exclude trimmer delay in 2nd round of reset loop. 1: include trimmer delay in 2nd round of reset loop. For low data rate.
24:16	0x1	_NONE_	**CALIB_LENGTH_TRIO1_B**: The length of the peramble which use to do caliration. Count by nvcsi core clock. It's used to make sure the calibration is doing inside the preamble phase. If this length set to a small value, the calibration may can't complete within one packet and need several packets to complete.
15	0x0	_NONE_	**TRIM_CAL_E_NONSTOP_TRIO0_B**: Reserved. Trimmer calibration background monitoring control. 1: enable. The accumulator keeps updating.
14:13	0x0	_NONE_	**TRIM_CAL_GAIN_SEL_TRIO0_B**: Trimmer calibration loop gain selection [0]: gain control 0: coarse tune (counter1) gain = 1, fine tune (counter2) gain = 1/4. For example: coarse tune: 1 UP -> trim_cal_code + 1; fine tune: 4 UPs -> trim_cal_code + 1. 1: coarse tune (counter1) gain = 1/2, fine tune (counter2) gain = 1/8 [1]: reserved

Bit	Reset	PROD	Description
12:10	0x4	0x2	**TRIM_CAL_CNT_SEL_TRIO0_B**: Trimmer calibration counter selection [0]: counter1 (coarse tune) clock cycle. 0: 64T (128UI) 1: 128T (256UI) [2:1]: counter2 (fine tune) clock cycle. 00: 32T (64UI) 01: 64T (128UI) 10: 96T (192UI) 11: 128T (256UI) For high data rate (>3.5GS/s): 64T(cnt1)+96T(cnt2)=160T(320ui) -> cnt_sel=3'b100 For low data rate (<=3.5GS/s): 128T(cnt1)+32T(cnt2)=160T(320ui) -> cnt_se=3'b001
9	0x1	_NONE_	**TRIM_CAL_CPHY_ED_LOOPMODE_TRIO0_B**: Trimmer calibration and cphy edge detector loop mode control 0: exclude trimmer delay in 2nd round of reset loop. 1: include trimmer delay in 2nd round of reset loop. For low data rate.
8:0	0x1	_NONE_	**CALIB_LENGTH_TRIO0_B**: The length of the peramble which use to do caliration. Count by nvcsi core clock. It's used to make sure the calibration is doing inside the preamble phase. If this length set to a small value, the calibration may can't complete within one packet and need several packets to complete.

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_CTRL1_0

Offset: 0xc785
Byte Offset: 0x31e14
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0c0e0c0e (0bx000,1100,0000,1110,x000,1100,0000,1110)

Bit	Reset	Description
30	0x0	**PERIODIC_CALIB_TRIO1_B**: Enable the periodic calibration.
29	0x0	**CALIB_APPLY_IN_POST_TRIO1_B**: 1: Apply the calibration result when detect post symbol. 0: Apply the calibration result immediately when the calibration done.

Bit	Reset	Description
28:27	0x1	**CALIB_APPLY_SELECT_TRIO1_B**: Which calibration result is applied to pad for dual calibration. 2'b11: Reserved 2'b10: Apply the middle value of two calibration to pad if the difference within threshold. 2'b01: Apply the 2nd calibration result to pad if it's difference with previous latch result within threshold. 2'b00: Apply the 1st calibration result to pad if it's difference with previous latch result within threshold.
26	0x1	**DUAL_CALIB_TRIO1_B**: Two times calibratiton. 1: Two time calibration with one trigger. 0: Single time calibration with one trigger
25:18	0x3	**TEMP_EDGE_DELAY_VALUE_TRIO1_B**: The edge delay value while doing the calibration.
17	0x1	**TEMP_EDGE_DELAY_SEL_TRIO1_B**: When the selection of edge delay value when do edge delay calibration. 1: Use the temp value from register setting. 0: Use the currently using setting. For the first calibration after reset, it's still using the temp value from register setting.
16	0x0	**BYPASS_CALIB_PACKET_TRIO1_B**: 1: Drop the packet data when do the calibration. 0: Not drop the packet data when do the calibration.
14	0x0	**PERIODIC_CALIB_TRIO0_B**: Enable the periodic calibration.
13	0x0	**CALIB_APPLY_IN_POST_TRIO0_B**: 1: Apply the calibration result when detect post symbol. 0: Apply the calibration result immediately when the calibration done.
12:11	0x1	**CALIB_APPLY_SELECT_TRIO0_B**: Which calibration result is applied to pad for dual calibration. 2'b11: Reserved 2'b10: Apply the middle value of two calibration to pad if the difference within threshold. 2'b01: Apply the 2nd calibration result to pad if it's difference with previous latch result within threshold. 2'b00: Apply the 1st calibration result to pad if it's difference with previous latch result within threshold.
10	0x1	**DUAL_CALIB_TRIO0_B**: Two times calibratiton. 1: Two time calibration with one trigger. 0: Single time calibration with one trigger
9:2	0x3	**TEMP_EDGE_DELAY_VALUE_TRIO0_B**: The edge delay value apply to pad while doing the calibration.
1	0x1	**TEMP_EDGE_DELAY_SEL_TRIO0_B**: When the selection of edge delay value when do edge delay calibration. 1: Use the temp value from register setting. 0: Use the currently using setting. For the first calibration after reset, it's still using the temp value from register setting.
0	0x0	**BYPASS_CALIB_PACKET_TRIO0_B**: 1: Drop the packet data when do the calibration. 0: Not drop the packet data when do the calibration.

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_CTRL2_0

Offset: 0xc786
Byte Offset: 0x31e18
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01000100 (0bxxxx,xxx1,0000,0000,xxxx,xxx1,0000,0000)

Bit	Reset	Description
24	ENABLE	**TRIM_CAL_PD_HOLD_TRIO1_B**: Hold in PD **0** = DISABLE **1** = ENABLE
23	DISABLE	**TRIM_CAL_E_FIXEDDLY_TRIO1_B**: Enable fixed delay in trim calibration unit. **0** = DISABLE **1** = ENABLE
22:16	0x0	**TRIM_CAL_FIXEDDLY_TRIO1_B**: Adjust fixed delay. Delay = code*5ps(TT)/7ps(SS)
8	ENABLE	**TRIM_CAL_PD_HOLD_TRIO0_B**: Hold in PD **0** = DISABLE **1** = ENABLE
7	DISABLE	**TRIM_CAL_E_FIXEDDLY_TRIO0_B**: Enable fixed delay in trim calibration unit. **0** = DISABLE **1** = ENABLE
6:0	0x0	**TRIM_CAL_FIXEDDLY_TRIO0_B**: Adjust fixed delay. Delay = code*5ps(TT)/7ps(SS)

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_CTRL3_0

Offset: 0xc787
Byte Offset: 0x31e1c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffffffff	**CALIB_PERIOD_TRIO0_B**: The period of the periodic edge delay calibration, in nvcsi clock cycle number.

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_CTRL4_0

Offset: 0xc788
Byte Offset: 0x31e20
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffffffff	**CALIB_PERIOD_TRIO1_B**: The period of the periodic edge delay calibration, in nvcsi clock cycle number.

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_CTRL5_0

Offset: 0xc789
Byte Offset: 0x31e24
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000ffff (0bxxxx,xxxx,xxxx,xxxx,1111,1111,1111,1111)

Bit	Reset	Description
15:8	0xff	**ERR_THRESHOLD_TRIO1_B**: The threshold of drop the calibration result between current result and previous result. For dual calibration, check between the two calibration result in the same trigger. For signal calibration, check between the current calibration result and previous calibration result.
7:0	0xff	**ERR_THRESHOLD_TRIO0_B**: The threshold of drop the calibration result between current result and previous result. For dual calibration, check between the two calibration result in the same trigger. For signal calibration, check between the current calibration result and previous calibration result.

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_STATUS_0

Offset: 0xc78a
Byte Offset: 0x31e28
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,xxxx)

Bit	Reset	Description
9	0x0	**EDGE_DELAY_MISMATCH_TRIO1_B**: When the CALIB_APPLY_SELECT_TRIO is 0/1 or dual time calibration is not enabled, the calibration result difference from the previous latched calibration result is larger than threshold, this time calibration result is not apply to pad. When the CALIB_APPLY_SELECT_TRIO is 2, the calibration result difference of the two times calibration result is larger than threshold, this time calibration result is not apply to pad.
8	0x0	**EDGE_DELAY_MISMATCH_TRIO0_B**: When the CALIB_APPLY_SELECT_TRIO is 0/1 or dual time calibration is not enabled, the calibration result difference from the previous latched calibration result is larger than threshold, this time calibration result is not apply to pad. When the CALIB_APPLY_SELECT_TRIO is 2, the calibration result difference of the two times calibration result is larger than threshold, this time calibration result is not apply to pad.
7	0x0	**CALIB_DONE_1_TRIO1_B**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.
6	0x0	**CALIB_DONE_1_TRIO0_B**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.
5	0x0	**CALIB_DONE_0_TRIO1_B**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.
4	0x0	**CALIB_DONE_0_TRIO0_B**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_EDGE_DELAY_CTRL_0

Offset: 0xc78b
Byte Offset: 0x31e2c
Read/Write: R/W
Parity Protection: See table below
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x40000000 (0b01xx,x000,0000,0000,xxxx,x000,0000,0000)

Bit	Parity Protection	Reset	Description
31:30	Y	0x1	**CPHY_CLKGEN_TRIGWIDTH_B**: Adjust AB/CB/CA pulse width in self-clock generator
26	N	0x0	**UPDATE_OFFSET_TRIO1_B**: Trigger to apply the a offset value. Used to update offset directly without trigger a new calibration.
25:17	Y	0x0	**DELAY_OFFSET_TRIO1_B**: Offset value of trio B1. Represents signed 2's complement number for calibration offset. Represents unsigned number if the field used as override value.
16	Y	0x0	**OVERRIDE_CAL_VAL_TRIO1_B**: Override for trio B1 0: Apply the DELAY_OFFSET as an offset to the calibration result. 1: Apply the DELAY_OFFSET value to the pad.
10	N	0x0	**UPDATE_OFFSET_TRIO0_B**: Trigger to apply the a offset value. Used to update offset directly without trigger a new calibration.
9:1	Y	0x0	**DELAY_OFFSET_TRIO0_B**: Offset value of trio B0. Represents signed 2's complement number for calibration offset. Represents unsigned number if the field used as override value.
0	Y	0x0	**OVERRIDE_CAL_VAL_TRIO0_B**: Override for trio B0 0: Apply the DELAY_OFFSET as an offset to the calibration result. 1: Apply the DELAY_OFFSET value to the pad.

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_EDGE_DELAY_VALUE_0

Offset: 0xc78c
Byte Offset: 0x31e30
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000303 (0bxxxx,xxxx,xxxx,xxxx,0000,0011,0000,0011)

Bit	Reset	Description
15:8	0x3	**EDGE_DELAY_VALUE_TRIO1_B**: The edge delay value apply to pad finally

Bit	Reset	Description
7:0	0x3	**EDGE_DELAY_VALUE_TRIO0_B**: The edge delay value apply to pad finally

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_RESULT_0

Offset: 0xc78d
Byte Offset: 0x31e34
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**EDGE_DELAY_CALIB_RESULT_1_TRIO1_B**: The edge delay calibrate result, with offset.
23:16	0x0	**EDGE_DELAY_CALIB_RESULT_1_TRIO0_B**: The edge delay calibrate result, with offset.
15:8	0x0	**EDGE_DELAY_CALIB_RESULT_0_TRIO1_B**: The edge delay calibrate result, with offset.
7:0	0x0	**EDGE_DELAY_CALIB_RESULT_0_TRIO0_B**: The edge delay calibrate result, with offset.

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_INADJ_CALIB_CTRL_0

Offset: 0xc78e
Byte Offset: 0x31e38
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
16	0x0	**INADJ_CAL_TRIGGER_TRIO1_B**: Trigger INADJ calibration for trio0
0	0x0	**INADJ_CAL_TRIGGER_TRIO0_B**: Trigger INADJ calibration for trio1

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_INADJ_CALIB_CTRL1_0

Offset: 0xc78f
Byte Offset: 0x31e3c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00010001 (0b0000,0000,0000,0001,0000,0000,0000,0001)

Bit	Reset	Description
31:16	0x1	**PRBS9_SEED_TRIO1_B**: PRBS9 initial seed for trio0
15:0	0x1	**PRBS9_SEED_TRIO0_B**: PRBS9 initial seed for trio1

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_INADJ_CALIB_CTRL2_0

Offset: 0xc790
Byte Offset: 0x31e40
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0fc00fc0 (0bxxxx,1111,1100,0000,xxxx,1111,1100,0000)

Bit	Reset	Description
27:22	0x3f	**INADJ_LIMIT_HIGH_TRIO1_B**: The upper limit for the INADJ sweep.
21:16	0x0	**INADJ_LIMIT_LOW_TRIO1_B**: The lower limit for the INADJ sweep.
11:6	0x3f	**INADJ_LIMIT_HIGH_TRIO0_B**: The upper limit for the INADJ sweep.
5:0	0x0	**INADJ_LIMIT_LOW_TRIO0_B**: The lower limit for the INADJ sweep.

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_INADJ_CALIB_STATUS_0

Offset: 0xc791
Byte Offset: 0x31e44

Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**INADJ_CALIB_DONE_TRIO1_B**: Trio b1 inadj calibration done.
0	0x0	**INADJ_CALIB_DONE_TRIO0_B**: Trio b0 inadj calibration done.

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_INADJ_CALIB_STATUS1_0

Offset: 0xc792
Byte Offset: 0x31e48
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_LOW_TRIO0_B**: Trio b0 PRBS check status for lower 32 INADJ value.

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_INADJ_CALIB_STATUS2_0

Offset: 0xc793
Byte Offset: 0x31e4c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_LOW_TRIO1_B**: Trio b1 PRBS check status for lower 32 INADJ value.

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_INADJ_CALIB_STATUS3_0

Offset: 0xc794
Byte Offset: 0x31e50
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_HIGH_TRIO0_B**: Trio b0 PRBS check status for upper 32 INADJ value.

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_INADJ_CALIB_STATUS4_0

Offset: 0xc795
Byte Offset: 0x31e54
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_HIGH_TRIO1_B**: Trio b1 PRBS check status for upper 32 INADJ value.

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_INADJ_OVERRIDE_0

Offset: 0xc796
Byte Offset: 0x31e58
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00080008 (0bxxxx,xxx0,0000,1000,xxxx,xxx0,0000,1000)

Bit	Reset	Description
24:17	0x4	**CPHY_INADJ_TRIO1_B**: The SW override INADJ value.
16	DISABLE	**SW_SET_CPHY_INADJ_TRIO1_B**: Enable SW override of the INADJ. **0** = DISABLE **1** = ENABLE
8:1	0x4	**CPHY_INADJ_TRIO0_B**: The SW override INADJ value.
0	DISABLE	**SW_SET_CPHY_INADJ_TRIO0_B**: Enable SW override of the INADJ. **0** = DISABLE **1** = ENABLE

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_BIST_CTRL_0

Offset: 0xc797
Byte Offset: 0x31e5c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	0x0	**STOP_BIST_TRIO1_B**: Manually stop the B1 BIST, write 1 to stop the RX bist.
2	0x0	**STOP_BIST_TRIO0_B**: Manually stop the B0 BIST, write 1 to stop the RX bist.
1	0x0	**TRIGGER_BIST_TRIO1_B**: Trigger the trio B1 BIST, write 1 to trigger the RX bist, HW will clear it to 0 when the BIST is done, write 0 to this field is with no effect.
0	0x0	**TRIGGER_BIST_TRIO0_B**: Trigger the trio B0 BIST, write 1 to trigger the RX bist, HW will clear it to 0 when the BIST is done, write 0 to this field is with no effect.

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_BIST_STATUS_0_0

Offset: 0xc798
Byte Offset: 0x31e60
Read/Write: RO
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:8	0x0	**WORD_ERR_CNT_TRIO1_B**: The trio B1 number of word comparison failure, report for BIST mode 1.
7:0	0x0	**WORD_ERR_CNT_TRIO0_B**: The trio B0 number of word comparison failure, report for BIST mode 1.

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_BIST_STATUS_1_0

Offset: 0xc799
Byte Offset: 0x31e64
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:8	0x0	**SYM_ERR_CNT_TRIO1_B**: The trio B1 number of symbol compariosn failure, it's the sum of the symbol err count and the clock shift count, the real wire state error number is 1/2 of this field, report for BIST mode 1.
7:0	0x0	**SYM_ERR_CNT_TRIO0_B**: The trio B0 number of symbol compariosn failure, it's the sum of the symbol err count and the clock shift count, the real wire state error number is 1/2 of this field, report for BIST mode 1.

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_BIST_STATUS_2_0

Offset: 0xc79a
Byte Offset: 0x31e68
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WORD_CNT_LOW_TRIO0_B**: The trio B0 low 32 bits of the compared word number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_BIST_STATUS_3_0

Offset: 0xc79b
Byte Offset: 0x31e6c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WORD_CNT_HIGH_TRIO0_B**: The trio B0 high 16 bits of the compared word number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_BIST_STATUS_4_0

Offset: 0xc79c
Byte Offset: 0x31e70
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WORD_CNT_LOW_TRIO1_B**: The trio B1 low 32 bits of the compared word number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_BIST_STATUS_5_0

Offset: 0xc79d
Byte Offset: 0x31e74
Read/Write: RO
Parity Protection: N
Shadow: N

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WORD_CNT_HIGH_TRIO1_B**: The trio B1 high 16 bits of the compared word number number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_2_NVCSI_CIL_B_CPHY_BIST_STATUS_6_0

Offset: 0xc79e
Byte Offset: 0x31e78
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,x000,0000,0000,xxxx,x000,0000,0000)

Bit	Reset	Description
26	0x0	**ERR_DETECT_FAIL_TRIO1_B**: Multi symbol error in one word (7 symbol) for trio B1, BIST logic is not able to detect the accurate wire state error. With this error, if not other errors detected, the word error is accurate.
25	0x0	**SHIFT_DETECT_FAIL_TRIO1_B**: Detect symbol error in two continuous word (14 symbols) for trio B1, BIST logic is not able to determine if symbol clock is slip. (if the symbol error count is not saturating, the result is still available with this error)
24	0x0	**SHIFT_OVERFLOW_TRIO1_B**: Too many clocks slip and the BIST logic is not able to re-align for trio B1. Typically, when this error is detect, the symbol error counter will saturate.
23:20	0x0	**DOUBLE_CLK_MISSED_CNT_TRIO1_B**: Number of the times for double symbol clock lost for trio B1.
19:16	0x0	**SINGLE_CLK_MISSED_CNT_TRIO1_B**: Number of the times for single symbol clock lost for trio B1.
10	0x0	**ERR_DETECT_FAIL_TRIO0_B**: Multi symbol error in one word (7 symbol) for trio B0, BIST logic is not able to detect the accurate wire state error. With this error, if not other errors detected, the word error is accurate.
9	0x0	**SHIFT_DETECT_FAIL_TRIO0_B**: Detect symbol error in two continuous word (14 symbols) for trio B0, BIST logic is not able to determine if symbol clock is slip. (if the symbol error count is not saturating, the result is still available with this error)
8	0x0	**SHIFT_OVERFLOW_TRIO0_B**: Too many clocks slip and the BIST logic is not able to re-align for trio B0. Typically, when this error is detect, the symbol error counter will saturate.

Bit	Reset	Description
7:4	0x0	**DOUBLE_CLK_MISSED_CNT_TRIO0_B**: Number of the times for double symbol clock lost for trio B0.
3:0	0x0	**SINGLE_CLK_MISSED_CNT_TRIO0_B**: Number of the times for single symbol clock lost for trio B0.

NVCSI_PHY_3_CILA_INTR_0_STATUS_CILA_0

Offset: 0x10500
Byte Offset: 0x41400
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,00x0,0xxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
30	0x0	**intr_cphy_edge_delay_cal_done_trio1_a**
29	0x0	**intr_cphy_edge_delay_cal_done_trio0_a**
28	0x0	**intr_dphy_cil_lane_align_err_a**
27	0x0	**intr_dphy_cil_deskew_calib_err_ctrl_a**
26	0x0	**intr_dphy_cil_deskew_calib_err_lane1_a**
25	0x0	**intr_dphy_cil_deskew_calib_err_lane0_a**
24	0x0	**intr_dphy_cil_deskew_calib_done_ctrl_a**
23	0x0	**intr_dphy_cil_deskew_calib_done_lane1_a**
22	0x0	**intr_dphy_cil_deskew_calib_done_lane0_a**
20	0x0	**intr_cil_data_lane_esc_mode_sync_err1_a**
19	0x0	**intr_cil_data_lane_esc_mode_sync_err0_a**
8	0x0	**intr_cil_data_lane_rxfifo_full_err1_a**
7	0x0	**intr_cil_data_lane_ctrl_err1_a**
6	0x0	**intr_cil_data_lane_sot_mb_err1_a**
5	0x0	**intr_cil_data_lane_sot_sb_err1_a**
4	0x0	**intr_cil_data_lane_rxfifo_full_err0_a**

Bit	Reset	Description
3	0x0	**intr_cil_data_lane_ctrl_err0_a**
2	0x0	**intr_cil_data_lane_sot_mb_err0_a**
1	0x0	**intr_cil_data_lane_sot_sb_err0_a**
0	0x0	**intr_dphy_cil_clk_lane_ctrl_err_a**

NVCSI_PHY_3_CILA_INTR_1_STATUS_CILA_0

Offset: 0x10501
Byte Offset: 0x41404
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**intr_cphy_inadj_cal_done_trio1_a**
13	0x0	**intr_cphy_inadj_cal_done_trio0_a**
12	0x0	**intr_idac_cal_done_trio1_a**
11	0x0	**intr_idac_cal_done_trio0_a**
10	0x0	**intr_dphy_cil_clk_lane_ulpm_req_a**
9	0x0	**intr_cil_lpdt_int1_a**
8	0x0	**intr_cil_ulps_trigger_int1_a**
7	0x0	**intr_cil_remoterst_trigger_int1_a**
6	0x0	**intr_cil_lpdt_int0_a**
5	0x0	**intr_cil_ulps_trigger_int0_a**
4	0x0	**intr_cil_remoterst_trigger_int0_a**
3	0x0	**intr_cil_data_lane_esc_data_rec1_a**
2	0x0	**intr_cil_data_lane_esc_cmd_rec1_a**
1	0x0	**intr_cil_data_lane_esc_data_rec0_a**
0	0x0	**intr_cil_data_lane_esc_cmd_rec0_a**

NVCSI_PHY_3_CILA_INTR_0_MASK_CILA_0
Offset: 0x10502
Byte Offset: 0x41408
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,00x0,0xxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
30	0x0	**intr_mask_cphy_edge_delay_cal_done_trio1_a**
29	0x0	**intr_mask_cphy_edge_delay_cal_done_trio0_a**
28	0x0	**intr_mask_dphy_cil_lane_align_err_a**
27	0x0	**intr_mask_dphy_cil_deskew_calib_err_ctrl_a**
26	0x0	**intr_mask_dphy_cil_deskew_calib_err_lane1_a**
25	0x0	**intr_mask_dphy_cil_deskew_calib_err_lane0_a**
24	0x0	**intr_mask_dphy_cil_deskew_calib_done_ctrl_a**
23	0x0	**intr_mask_dphy_cil_deskew_calib_done_lane1_a**
22	0x0	**intr_mask_dphy_cil_deskew_calib_done_lane0_a**
20	0x0	**intr_mask_cil_data_lane_esc_mode_sync_err1_a**
19	0x0	**intr_mask_cil_data_lane_esc_mode_sync_err0_a**
8	0x0	**intr_mask_cil_data_lane_rxfifo_full_err1_a**
7	0x0	**intr_mask_cil_data_lane_ctrl_err1_a**
6	0x0	**intr_mask_cil_data_lane_sot_mb_err1_a**
5	0x0	**intr_mask_cil_data_lane_sot_sb_err1_a**
4	0x0	**intr_mask_cil_data_lane_rxfifo_full_err0_a**
3	0x0	**intr_mask_cil_data_lane_ctrl_err0_a**
2	0x0	**intr_mask_cil_data_lane_sot_mb_err0_a**
1	0x0	**intr_mask_cil_data_lane_sot_sb_err0_a**
0	0x0	**intr_mask_dphy_cil_clk_lane_ctrl_err_a**

NVCSI_PHY_3_CILA_INTR_1_MASK_CILA_0

Offset: 0x10503
Byte Offset: 0x4140c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**intr_mask_cphy_inadj_cal_done_trio1_a**
13	0x0	**intr_mask_cphy_inadj_cal_done_trio0_a**
12	0x0	**intr_mask_idac_cal_done_trio1_a**
11	0x0	**intr_mask_idac_cal_done_trio0_a**
10	0x0	**intr_mask_dphy_cil_clk_lane_ulpm_req_a**
9	0x0	**intr_mask_cil_lpdt_int1_a**
8	0x0	**intr_mask_cil_ulps_trigger_int1_a**
7	0x0	**intr_mask_cil_remoterst_trigger_int1_a**
6	0x0	**intr_mask_cil_lpdt_int0_a**
5	0x0	**intr_mask_cil_ulps_trigger_int0_a**
4	0x0	**intr_mask_cil_remoterst_trigger_int0_a**
3	0x0	**intr_mask_cil_data_lane_esc_data_rec1_a**
2	0x0	**intr_mask_cil_data_lane_esc_cmd_rec1_a**
1	0x0	**intr_mask_cil_data_lane_esc_data_rec0_a**
0	0x0	**intr_mask_cil_data_lane_esc_cmd_rec0_a**

NVCSI_PHY_3_CILA_CORRECTABLE_ERR_INTR_STATUS_CILA_0

Offset: 0x10504
Byte Offset: 0x41410
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**err_intr_cil_data_lane_esc_mode_sync_err1_a**
13	0x0	**err_intr_cil_data_lane_esc_mode_sync_err0_a**
12	0x0	**err_intr_dphy_cil_lane_align_err_a**
11	0x0	**err_intr_dphy_cil_deskew_calib_err_ctrl_a**
10	0x0	**err_intr_dphy_cil_deskew_calib_err_lane1_a**
9	0x0	**err_intr_dphy_cil_deskew_calib_err_lane0_a**
8	0x0	**err_intr_cil_data_lane_rxfifo_full_err1_a**
7	0x0	**err_intr_cil_data_lane_ctrl_err1_a**
6	0x0	**err_intr_cil_data_lane_sot_mb_err1_a**
5	0x0	**err_intr_cil_data_lane_sot_sb_err1_a**
4	0x0	**err_intr_cil_data_lane_rxfifo_full_err0_a**
3	0x0	**err_intr_cil_data_lane_ctrl_err0_a**
2	0x0	**err_intr_cil_data_lane_sot_mb_err0_a**
1	0x0	**err_intr_cil_data_lane_sot_sb_err0_a**
0	0x0	**err_intr_dphy_cil_clk_lane_ctrl_err_a**

NVCSI_PHY_3_CILA_UNCORRECTABLE_ERR_INTR_STATUS_CILA_0

Offset: 0x10505
Byte Offset: 0x41414
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**err_intr_cil_data_lane_esc_mode_sync_err1_a**
13	0x0	**err_intr_cil_data_lane_esc_mode_sync_err0_a**
12	0x0	**err_intr_dphy_cil_lane_align_err_a**
11	0x0	**err_intr_dphy_cil_deskew_calib_err_ctrl_a**

Bit	Reset	Description
10	0x0	**err_intr_dphy_cil_deskew_calib_err_lane1_a**
9	0x0	**err_intr_dphy_cil_deskew_calib_err_lane0_a**
8	0x0	**err_intr_cil_data_lane_rxfifo_full_err1_a**
7	0x0	**err_intr_cil_data_lane_ctrl_err1_a**
6	0x0	**err_intr_cil_data_lane_sot_mb_err1_a**
5	0x0	**err_intr_cil_data_lane_sot_sb_err1_a**
4	0x0	**err_intr_cil_data_lane_rxfifo_full_err0_a**
3	0x0	**err_intr_cil_data_lane_ctrl_err0_a**
2	0x0	**err_intr_cil_data_lane_sot_mb_err0_a**
1	0x0	**err_intr_cil_data_lane_sot_sb_err0_a**
0	0x0	**err_intr_dphy_cil_clk_lane_ctrl_err_a**

NVCSI_PHY_3_CILA_ERR_INTR_MASK_CILA_0

Offset: 0x10506
Byte Offset: 0x41418
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00007fff (0bxxxx,xxxx,xxxx,xxxx,x111,1111,1111,1111)

Bit	Reset	Description
14	0x1	**err_intr_mask_cil_data_lane_esc_mode_sync_err1_a**
13	0x1	**err_intr_mask_cil_data_lane_esc_mode_sync_err0_a**
12	0x1	**err_intr_mask_dphy_cil_lane_align_err_a**
11	0x1	**err_intr_mask_dphy_cil_deskew_calib_err_ctrl_a**
10	0x1	**err_intr_mask_dphy_cil_deskew_calib_err_lane1_a**
9	0x1	**err_intr_mask_dphy_cil_deskew_calib_err_lane0_a**
8	0x1	**err_intr_mask_cil_data_lane_rxfifo_full_err1_a**
7	0x1	**err_intr_mask_cil_data_lane_ctrl_err1_a**
6	0x1	**err_intr_mask_cil_data_lane_sot_mb_err1_a**

Bit	Reset	Description
5	0x1	**err_intr_mask_cil_data_lane_sot_sb_err1_a**
4	0x1	**err_intr_mask_cil_data_lane_rxfifo_full_err0_a**
3	0x1	**err_intr_mask_cil_data_lane_ctrl_err0_a**
2	0x1	**err_intr_mask_cil_data_lane_sot_mb_err0_a**
1	0x1	**err_intr_mask_cil_data_lane_sot_sb_err0_a**
0	0x1	**err_intr_mask_dphy_cil_clk_lane_ctrl_err_a**

NVCSI_PHY_3_CILA_ERR_INTR_TYPE_CILA_0

Offset: 0x10507
Byte Offset: 0x4141c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**err_intr_type_cil_data_lane_esc_mode_sync_err1_a**
13	0x0	**err_intr_type_cil_data_lane_esc_mode_sync_err0_a**
12	0x0	**err_intr_type_dphy_cil_lane_align_err_a**
11	0x0	**err_intr_type_dphy_cil_deskew_calib_err_ctrl_a**
10	0x0	**err_intr_type_dphy_cil_deskew_calib_err_lane1_a**
9	0x0	**err_intr_type_dphy_cil_deskew_calib_err_lane0_a**
8	0x0	**err_intr_type_cil_data_lane_rxfifo_full_err1_a**
7	0x0	**err_intr_type_cil_data_lane_ctrl_err1_a**
6	0x0	**err_intr_type_cil_data_lane_sot_mb_err1_a**
5	0x0	**err_intr_type_cil_data_lane_sot_sb_err1_a**
4	0x0	**err_intr_type_cil_data_lane_rxfifo_full_err0_a**
3	0x0	**err_intr_type_cil_data_lane_ctrl_err0_a**
2	0x0	**err_intr_type_cil_data_lane_sot_mb_err0_a**
1	0x0	**err_intr_type_cil_data_lane_sot_sb_err0_a**

Bit	Reset	Description
0	0x0	**err_intr_type_dphy_cil_clk_lane_ctrl_err_a**

NVCSI_PHY_3_CILB_INTR_0_STATUS_CILB_0

Offset: 0x10600
Byte Offset: 0x41800
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,00x0,0xxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
30	0x0	**intr_cphy_edge_delay_cal_done_trio1_b**
29	0x0	**intr_cphy_edge_delay_cal_done_trio0_b**
28	0x0	**intr_dphy_cil_lane_align_err_b**
27	0x0	**intr_dphy_cil_deskew_calib_err_ctrl_b**
26	0x0	**intr_dphy_cil_deskew_calib_err_lane1_b**
25	0x0	**intr_dphy_cil_deskew_calib_err_lane0_b**
24	0x0	**intr_dphy_cil_deskew_calib_done_ctrl_b**
23	0x0	**intr_dphy_cil_deskew_calib_done_lane1_b**
22	0x0	**intr_dphy_cil_deskew_calib_done_lane0_b**
20	0x0	**intr_cil_data_lane_esc_mode_sync_err1_b**
19	0x0	**intr_cil_data_lane_esc_mode_sync_err0_b**
8	0x0	**intr_cil_data_lane_rxfifo_full_err1_b**
7	0x0	**intr_cil_data_lane_ctrl_err1_b**
6	0x0	**intr_cil_data_lane_sot_mb_err1_b**
5	0x0	**intr_cil_data_lane_sot_sb_err1_b**
4	0x0	**intr_cil_data_lane_rxfifo_full_err0_b**
3	0x0	**intr_cil_data_lane_ctrl_err0_b**
2	0x0	**intr_cil_data_lane_sot_mb_err0_b**

Bit	Reset	Description
1	0x0	**intr_cil_data_lane_sot_sb_err0_b**
0	0x0	**intr_dphy_cil_clk_lane_ctrl_err_b**

NVCSI_PHY_3_CILB_INTR_1_STATUS_CILB_0

Offset: 0x10601
Byte Offset: 0x41804
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**intr_cphy_inadj_cal_done_trio1_b**
13	0x0	**intr_cphy_inadj_cal_done_trio0_b**
12	0x0	**intr_idac_cal_done_trio1_b**
11	0x0	**intr_idac_cal_done_trio0_b**
10	0x0	**intr_dphy_cil_clk_lane_ulpm_req_b**
9	0x0	**intr_cil_lpdt_int1_b**
8	0x0	**intr_cil_ulps_trigger_int1_b**
7	0x0	**intr_cil_remoterst_trigger_int1_b**
6	0x0	**intr_cil_lpdt_int0_b**
5	0x0	**intr_cil_ulps_trigger_int0_b**
4	0x0	**intr_cil_remoterst_trigger_int0_b**
3	0x0	**intr_cil_data_lane_esc_data_rec1_b**
2	0x0	**intr_cil_data_lane_esc_cmd_rec1_b**
1	0x0	**intr_cil_data_lane_esc_data_rec0_b**
0	0x0	**intr_cil_data_lane_esc_cmd_rec0_b**

NVCSI_PHY_3_CILB_INTR_0_MASK_CILB_0

Offset: 0x10602

Byte Offset: 0x41808
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,00x0,0xxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
30	0x0	**intr_mask_cphy_edge_delay_cal_done_trio1_b**
29	0x0	**intr_mask_cphy_edge_delay_cal_done_trio0_b**
28	0x0	**intr_mask_dphy_cil_lane_align_err_b**
27	0x0	**intr_mask_dphy_cil_deskew_calib_err_ctrl_b**
26	0x0	**intr_mask_dphy_cil_deskew_calib_err_lane1_b**
25	0x0	**intr_mask_dphy_cil_deskew_calib_err_lane0_b**
24	0x0	**intr_mask_dphy_cil_deskew_calib_done_ctrl_b**
23	0x0	**intr_mask_dphy_cil_deskew_calib_done_lane1_b**
22	0x0	**intr_mask_dphy_cil_deskew_calib_done_lane0_b**
20	0x0	**intr_mask_cil_data_lane_esc_mode_sync_err1_b**
19	0x0	**intr_mask_cil_data_lane_esc_mode_sync_err0_b**
8	0x0	**intr_mask_cil_data_lane_rxfifo_full_err1_b**
7	0x0	**intr_mask_cil_data_lane_ctrl_err1_b**
6	0x0	**intr_mask_cil_data_lane_sot_mb_err1_b**
5	0x0	**intr_mask_cil_data_lane_sot_sb_err1_b**
4	0x0	**intr_mask_cil_data_lane_rxfifo_full_err0_b**
3	0x0	**intr_mask_cil_data_lane_ctrl_err0_b**
2	0x0	**intr_mask_cil_data_lane_sot_mb_err0_b**
1	0x0	**intr_mask_cil_data_lane_sot_sb_err0_b**
0	0x0	**intr_mask_dphy_cil_clk_lane_ctrl_err_b**

NVCSI_PHY_3_CILB_INTR_1_MASK_CILB_0

Offset: 0x10603
Byte Offset: 0x4180c
Read/Write: R/W

Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**intr_mask_cphy_inadj_cal_done_trio1_b**
13	0x0	**intr_mask_cphy_inadj_cal_done_trio0_b**
12	0x0	**intr_mask_idac_cal_done_trio1_b**
11	0x0	**intr_mask_idac_cal_done_trio0_b**
10	0x0	**intr_mask_dphy_cil_clk_lane_ulpm_req_b**
9	0x0	**intr_mask_cil_lpdt_int1_b**
8	0x0	**intr_mask_cil_ulps_trigger_int1_b**
7	0x0	**intr_mask_cil_remoterst_trigger_int1_b**
6	0x0	**intr_mask_cil_lpdt_int0_b**
5	0x0	**intr_mask_cil_ulps_trigger_int0_b**
4	0x0	**intr_mask_cil_remoterst_trigger_int0_b**
3	0x0	**intr_mask_cil_data_lane_esc_data_rec1_b**
2	0x0	**intr_mask_cil_data_lane_esc_cmd_rec1_b**
1	0x0	**intr_mask_cil_data_lane_esc_data_rec0_b**
0	0x0	**intr_mask_cil_data_lane_esc_cmd_rec0_b**

NVCSI_PHY_3_CILB_CORRECTABLE_ERR_INTR_STATUS_CILB_0

Offset: 0x10604
Byte Offset: 0x41810
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**err_intr_cil_data_lane_esc_mode_sync_err1_b**
13	0x0	**err_intr_cil_data_lane_esc_mode_sync_err0_b**

Bit	Reset	Description
12	0x0	**err_intr_dphy_cil_lane_align_err_b**
11	0x0	**err_intr_dphy_cil_deskew_calib_err_ctrl_b**
10	0x0	**err_intr_dphy_cil_deskew_calib_err_lane1_b**
9	0x0	**err_intr_dphy_cil_deskew_calib_err_lane0_b**
8	0x0	**err_intr_cil_data_lane_rxfifo_full_err1_b**
7	0x0	**err_intr_cil_data_lane_ctrl_err1_b**
6	0x0	**err_intr_cil_data_lane_sot_mb_err1_b**
5	0x0	**err_intr_cil_data_lane_sot_sb_err1_b**
4	0x0	**err_intr_cil_data_lane_rxfifo_full_err0_b**
3	0x0	**err_intr_cil_data_lane_ctrl_err0_b**
2	0x0	**err_intr_cil_data_lane_sot_mb_err0_b**
1	0x0	**err_intr_cil_data_lane_sot_sb_err0_b**
0	0x0	**err_intr_dphy_cil_clk_lane_ctrl_err_b**

NVCSI_PHY_3_CILB_UNCORRECTABLE_ERR_INTR_STATUS_CILB_0

Offset: 0x10605
Byte Offset: 0x41814
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**err_intr_cil_data_lane_esc_mode_sync_err1_b**
13	0x0	**err_intr_cil_data_lane_esc_mode_sync_err0_b**
12	0x0	**err_intr_dphy_cil_lane_align_err_b**
11	0x0	**err_intr_dphy_cil_deskew_calib_err_ctrl_b**
10	0x0	**err_intr_dphy_cil_deskew_calib_err_lane1_b**
9	0x0	**err_intr_dphy_cil_deskew_calib_err_lane0_b**
8	0x0	**err_intr_cil_data_lane_rxfifo_full_err1_b**

Bit	Reset	Description
7	0x0	**err_intr_cil_data_lane_ctrl_err1_b**
6	0x0	**err_intr_cil_data_lane_sot_mb_err1_b**
5	0x0	**err_intr_cil_data_lane_sot_sb_err1_b**
4	0x0	**err_intr_cil_data_lane_rxfifo_full_err0_b**
3	0x0	**err_intr_cil_data_lane_ctrl_err0_b**
2	0x0	**err_intr_cil_data_lane_sot_mb_err0_b**
1	0x0	**err_intr_cil_data_lane_sot_sb_err0_b**
0	0x0	**err_intr_dphy_cil_clk_lane_ctrl_err_b**

NVCSI_PHY_3_CILB_ERR_INTR_MASK_CILB_0

Offset: 0x10606
Byte Offset: 0x41818
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00007fff (0bxxxx,xxxx,xxxx,xxxx,x111,1111,1111,1111)

Bit	Reset	Description
14	0x1	**err_intr_mask_cil_data_lane_esc_mode_sync_err1_b**
13	0x1	**err_intr_mask_cil_data_lane_esc_mode_sync_err0_b**
12	0x1	**err_intr_mask_dphy_cil_lane_align_err_b**
11	0x1	**err_intr_mask_dphy_cil_deskew_calib_err_ctrl_b**
10	0x1	**err_intr_mask_dphy_cil_deskew_calib_err_lane1_b**
9	0x1	**err_intr_mask_dphy_cil_deskew_calib_err_lane0_b**
8	0x1	**err_intr_mask_cil_data_lane_rxfifo_full_err1_b**
7	0x1	**err_intr_mask_cil_data_lane_ctrl_err1_b**
6	0x1	**err_intr_mask_cil_data_lane_sot_mb_err1_b**
5	0x1	**err_intr_mask_cil_data_lane_sot_sb_err1_b**
4	0x1	**err_intr_mask_cil_data_lane_rxfifo_full_err0_b**
3	0x1	**err_intr_mask_cil_data_lane_ctrl_err0_b**

Bit	Reset	Description
2	0x1	**err_intr_mask_cil_data_lane_sot_mb_err0_b**
1	0x1	**err_intr_mask_cil_data_lane_sot_sb_err0_b**
0	0x1	**err_intr_mask_dphy_cil_clk_lane_ctrl_err_b**

NVCSI_PHY_3_CILB_ERR_INTR_TYPE_CILB_0

Offset: 0x10607
Byte Offset: 0x4181c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	Reset	Description
14	0x0	**err_intr_type_cil_data_lane_esc_mode_sync_err1_b**
13	0x0	**err_intr_type_cil_data_lane_esc_mode_sync_err0_b**
12	0x0	**err_intr_type_dphy_cil_lane_align_err_b**
11	0x0	**err_intr_type_dphy_cil_deskew_calib_err_ctrl_b**
10	0x0	**err_intr_type_dphy_cil_deskew_calib_err_lane1_b**
9	0x0	**err_intr_type_dphy_cil_deskew_calib_err_lane0_b**
8	0x0	**err_intr_type_cil_data_lane_rxfifo_full_err1_b**
7	0x0	**err_intr_type_cil_data_lane_ctrl_err1_b**
6	0x0	**err_intr_type_cil_data_lane_sot_mb_err1_b**
5	0x0	**err_intr_type_cil_data_lane_sot_sb_err1_b**
4	0x0	**err_intr_type_cil_data_lane_rxfifo_full_err0_b**
3	0x0	**err_intr_type_cil_data_lane_ctrl_err0_b**
2	0x0	**err_intr_type_cil_data_lane_sot_mb_err0_b**
1	0x0	**err_intr_type_cil_data_lane_sot_sb_err0_b**
0	0x0	**err_intr_type_dphy_cil_clk_lane_ctrl_err_b**

NVCSI_PHY_3_NVCSI_CIL_PHY_CTRL_0

Offset: 0x10700
Byte Offset: 0x41c00
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DPHY	**CFG_PHY_MODE**: The CPHY/DPHY type mode. **0** = DPHY **1** = CPHY

NVCSI_PHY_3_LM_SLCG_CTRL_0

Offset: 0x10701
Byte Offset: 0x41c04
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	DISABLE	**cfg_lm1_slcg_override**: Enable the SLCG override for lane merger 1. **0** = DISABLE **1** = ENABLE
0	DISABLE	**cfg_lm0_slcg_override**: Enable the SLCG override for lane merger 0. **0** = DISABLE **1** = ENABLE

NVCSI_PHY_3_LM_EN_CTRL_0

Offset: 0x10702
Byte Offset: 0x41c08
Read/Write: R/W

Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000060c (0bxxxx,xxxx,xxxx,xxxx,xxx0,0110,xx00,11x0)

Bit	Reset	Description
12:9	0x3	**cfg_lm1_water_mark**: Water mark for LM0 detect stuck
8	DISABLE	**cfg_lm1_en**: Lane Merger 1 enable **0** = DISABLE **1** = ENABLE
5:2	0x3	**cfg_lm0_water_mark**: Water mark for LM0 detect stuck
0	DISABLE	**cfg_lm0_en**: Lane Merger 0 enable **0** = DISABLE **1** = ENABLE

NVCSI_PHY_3_LM_SW_RESET_0

Offset: 0x10703
Byte Offset: 0x41c0c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	DISABLE	**cfg_lm1_swreset**: Reset the lane merger 1. **0** = DISABLE **1** = ENABLE
0	DISABLE	**cfg_lm0_swreset**: Reset the lane merger 0. **0** = DISABLE **1** = ENABLE

NVCSI_PHY_3_NVCSI_CIL_CONFIG_0

Offset: 0x10704
Byte Offset: 0x41c10
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,x000,xxxx,x000)

Bit	Reset	Description
10:8	0x0	**DATA_LANE_B**: Lane number for lane merger B. 000: 0 lanes active 001: 1 lanes active 010: 2 lanes active Others: illegal
2:0	0x0	**DATA_LANE_A**: Lane number for lane merger A. 000: 0 lanes active 001: 1 lanes active 010: 2 lanes active 011: 3 lanes active (valid only in CPHY mode. NA in DPHY mode) 100: 4 lanes active, CLK from Partition A is used. 101: 4 lanes active, CLK from Partition B is used. Others: illegal

NVCSI_PHY_3_NVCSI_CIL_CLKEN_OVERRIDE_CTRL_0

Offset: 0x10705
Byte Offset: 0x41c14
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**CLKEN_OVERRIDE**: CLKENABLE OVERRIDE FOR CIL. **1** = ENABLE **0** = DISABLE

NVCSI_PHY_3_NVCSI_CIL_PAD_CONFIG_0

Offset: 0x10706
Byte Offset: 0x41c18

Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00040100 (0bxxxx,x000,0000,0100,0000,xxx1,0000,0000)

Bit	Reset	Description
26:25	0x0	**VD09REG_LEVEL**: 2-bit regulator level control. control voltage level output of 0.9v regulator. 00:0.9v, 01:0.85v, 10: 0.95v, 11: 1v
24:23	0x0	**VD09REG_LEAKER**: 2-bit regulator leaker control. 11 will give best performance, but consumers more DC quiescent current. (control 0.9v internal regulator for all partitions)
22:21	0x0	**VD04REG_LEVEL**: 2-bit regulator level control control voltage level output of 0.4V regulator 00: 0.4v 01: 0.38V 10: 0.42V 11: 0.44V
20:19	0x0	**VD04REG_LEAKER**: 2-bit regulator leaker control. 11 will give best performance, but consumers more DC quiescent current. (control 0.4v internal tx regulator for all partitions)
18:16	0x4	**LPRX_LEVEL_SEL**: Internal vref adjustment for LP self-biased receiver(when LP_RX_SELECT_IO1_[A/B]=1)
15:12	0x0	**LOADADJ**: LOAD ADJ value to be connected to the pad not used for CSI functional mode. used only for IO bist
8	0x1	**PDVCLAMP**: Power down regular which supplies current to de-serializer logic. Active High.
7:5	0x0	**SEL_CKTEST**
4	0x0	**E_CKTEST**: Enable clock test output.
3:1	0x0	**SEL_ATEST**
0	0x0	**E_ATEST**: Enable analog test voltage output.

NVCSI_PHY_3_NVCSI_CIL_PAD_CONFIG_1_0

Offset: 0x10707
Byte Offset: 0x41c1c
Read/Write: R/W
Parity Protection: Y

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SPARE_TOP**: spare register bits for top level control

NVCSI_PHY_3_NVCSI_CIL_LANE_SWIZZLE_CTRL_0

Offset: 0x10708
Byte Offset: 0x41c20
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4:0	0x0	**LANE_SWIZZLE_CTRL**: Lane Swizzle control for Brick. Valid in both CPHY and DPHY modes. For CPHY, all modes are supported. But for DPHY, some mode are not supported with different lane count. Only for 4 lanes: 00011 A0 A1 B0 B1 --> A0 B0 A1 B1 00100 A0 A1 B0 B1 --> A0 B1 A1 B0 00101 A0 A1 B0 B1 --> A0 B1 B0 A1 00010 A0 A1 B0 B1 --> A0 B0 B1 A1 01000 A0 A1 B0 B1 --> A1 B0 B1 A0 01001 A0 A1 B0 B1 --> A1 B0 A0 B1 01010 A0 A1 B0 B1 --> A1 B1 A0 B0 01011 A0 A1 B0 B1 --> A1 B1 B0 A0 01100 A0 A1 B0 B1 --> B0 A1 A0 B1 01101 A0 A1 B0 B1 --> B0 A1 B1 A0 01110 A0 A1 B0 B1 --> B0 A0 B1 A1 01111 A0 A1 B0 B1 --> B0 A0 A1 B1 10000 A0 A1 B0 B1 --> B0 B1 A1 A0 10001 A0 A1 B0 B1 --> B0 B1 A0 A1 10010 A0 A1 B0 B1 --> B1 A1 B0 A0 10011 A0 A1 B0 B1 --> B1 A1 A0 B0 10100 A0 A1 B0 B1 --> B1 B0 A0 A1 10101 A0 A1 B0 B1 --> B1 B0 A1 A0 10110 A0 A1 B0 B1 --> B1 A0 A1 B0 10111 A0 A1 B0 B1 --> B1 A0 B0 A1 Support for 1/2/4 lanes 00000 A0 A1 B0 B1 --> A0 A1 B0 B1 00001 A0 A1 B0 B1 --> A0 A1 B1 B0 00110 A0 A1 B0 B1 --> A1 A0 B0 B1 00111 A0 A1 B0 B1 --> A1 A0 B1 B0

NVCSI_PHY_3_NVCSI_CIL_BK_MODE_STATUS_0

Offset: 0x10709
Byte Offset: 0x41c24
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
1:0	X	**BK_MODE**: 00: two independent 2x bricks 01: one 4x brick, received clock from partition A is used. Clock from partition B is not used 10: one 4x brick, received clock from partition B is used. Clock from partition A is not used 11: illegal

NVCSI_PHY_3_NVCSI_CIL_TX_TIMING_0_0

Offset: 0x1070a
Byte Offset: 0x41c28
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00006030 (0bxxxx,xxxx,xxxx,xxxx,0110,0000,0011,0000)

Bit	Reset	Description
15:8	0x60	**THSEXIT**: THSEXIT length, in slow clock cycle number
7:0	0x30	**TLPX**: TLPX length, in slow clock cycle number

NVCSI_PHY_3_NVCSI_CIL_TX_TIMING_1_0

Offset: 0x1070b
Byte Offset: 0x41c2c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00101030 (0bxxxx,xxxx,0001,0000,0001,0000,0011,0000)

Bit	Reset	Description
23:16	0x10	**T3POST**: T3POST length, in slow clock cycle number
15:8	0x10	**T3PREBEGIN**: T3PREBEGIN length, in slow clock cycle number
7:0	0x30	**T3PREPARE**: T3PREPARE length, in slow clock cycle number

NVCSI_PHY_3_NVCSI_CIL_TX_TIMING_2_0

Offset: 0x1070c
Byte Offset: 0x41c30
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:8	0xfff	**T3CALALTSEQ**: T3CALALTSEQ length, in slow clock cycle number
7:0	0xff	**T3CALPREAMBLE**: T3CALPREAMBLE length, in slow clock cycle number

NVCSI_PHY_3_NVCSI_CIL_TX_CALIB_CTRL_0

Offset: 0x1070d
Byte Offset: 0x41c34
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000004 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0100)

Bit	Reset	Description
17:2	0x1	**CALIB_SEED**: Calibration sequence format 2 PRBS seed. Should set to a non-zero value.

Bit	Reset	Description
1	FMT_1	**CALIB_SEQ**: Calibration sequence format. **0** = FMT_1 **1** = FMT_2
0	DISABLE	**CALIB_EN**: Calibration sequence enable **0** = DISABLE **1** = ENABLE

NVCSI_PHY_3_NVCSI_CIL_TX_CTRL_0

Offset: 0x1070e
Byte Offset: 0x41c38
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000004 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0100)

Bit	Reset	Description
4	DISABLE	**TX_ENABLE**: Enable the TX SCIL. **0** = DISABLE **1** = ENABLE
3	SINGLE	**PAD_BK_MODE**: Set the pad to one stream or two independent stream usage. **0** = SINGLE **1** = DUAL
2:0	0x4	**LANE_NUM**: Valid value: 1/2/4

NVCSI_PHY_3_NVCSI_CIL_TX_SW_RESET_0_0

Offset: 0x1070f
Byte Offset: 0x41c3c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	ENABLE	**SW_RESET**: Reset the TX lane **0** = DISABLE **1** = ENABLE

NVCSI_PHY_3_NVCSI_CIL_CPHY_BIST_CONFIG_0_0

Offset: 0x10710
Byte Offset: 0x41c40
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000009 (0bxxxx,xxxx,xx00,0000,0000,0000,0000,1001)

Bit	Reset	Description
21	DISABLE	**REALIGN_EN**: If set to enable, the BIST logic will re-align the reference symbol to match the received symbol when there is clock slip by wire state error. **0** = DISABLE **1** = ENABLE
20:3	0x1	**PRBS_SEED**: Seed of the PRBS pattern. Should set to a non-zero value.
2:1	0x0	**PRBS_PATTERN**: PRBS pattern PRBS9: X0+X5+X9 PRBS11: X0+X9+X11 PRBS18: X0+X11+X18 **0** = PRBS9 **1** = PRBS11 **2** = PRBS18
0	MODE1	**BIST_MODE**: BIST control mode MODE0: Set the expected error number, BIST logic will keep comparing the received symbols and the reference PRBS generated symbols. When the expected error number got, BIST logic stop. And report how many word are totally compared. MODE11: Set the expected word number, BIST logic will keep comparing the received symbols and the reference PRBS generated symbols. When the expected words(symbols) are compared, BIST logic stop and report how many errors are found. **0** = MODE0 **1** = MODE1

NVCSI_PHY_3_NVCSI_CIL_CPHY_BIST_CONFIG_1_0

Offset: 0x10711
Byte Offset: 0x41c44
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000100 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx1,0000,0000)

Bit	Reset	Description
8	0x1	**ERR_TYPE**: The expecte error number type. SYMBOL_NUM: symbol error number. **0** = WORD_NUM :WORD_NUM: word error number. **1** = SYMBOL_NUM
7:0	0x0	**ERR_NUM**: The expected error number, used for BIST MODE 0.

NVCSI_PHY_3_NVCSI_CIL_CPHY_BIST_CONFIG_2_0

Offset: 0x10712
Byte Offset: 0x41c48
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WORD_NUM_LOW**: The low 32 bits of the expect word number, used for BIST MODE 1.

NVCSI_PHY_3_NVCSI_CIL_CPHY_BIST_CONFIG_3_0

Offset: 0x10713
Byte Offset: 0x41c4c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WORD_NUM_HIGH**: The higher 16 bits of the expected word number, used for BIST MODE 1.

NVCSI_PHY_3_NVCSI_CIL_IDAC_CALIB_START_0

Offset: 0x10714
Byte Offset: 0x41c50
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**IDAC_CAL_TRIGGER**: Trigger the IDAC calibration.

NVCSI_PHY_3_NVCSI_CIL_IDAC_CALIB_CTRL_0

Offset: 0x10715
Byte Offset: 0x41c54
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,0000,xxxx,x000,0000,0101)

Bit	Reset	Description
19	0x0	**IDAC_CAL_EN_TRIO1_B**: Enable IDAC calibration for trio B1
18	0x0	**IDAC_CAL_EN_TRIO0_B**: Enable IDAC calibration for trio B0
17	0x0	**IDAC_CAL_EN_TRIO1_A**: Enable IDAC calibration for trio A1
16	0x0	**IDAC_CAL_EN_TRIO0_A**: Enable IDAC calibration for trio A0

Bit	Reset	Description
10:0	0x5	**IDAC_SETTLE_TIME**: How many cycle for settle to sample the done from pad.

NVCSI_PHY_3_NVCSI_CIL_IDAC_CALIB_STATUS_0

Offset: 0x10716
Byte Offset: 0x41c58
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,0000,0000,0000,xxxx,0000,0000,0000)

Bit	Reset	Description
27	0x0	**IDAC_CALIB_MAXOUT_RXCA_TRIO1_B**: IDAC calibration maxout for trio B1 RXCA.
26	0x0	**IDAC_CALIB_MAXOUT_RXBC_TRIO1_B**: IDAC calibration maxout for trio B1 RXBC.
25	0x0	**IDAC_CALIB_MAXOUT_RXAB_TRIO1_B**: IDAC calibration maxout for trio B1 RXAB.
24	0x0	**IDAC_CALIB_MAXOUT_RXCA_TRIO0_B**: IDAC calibration maxout for trio B0 RXCA.
23	0x0	**IDAC_CALIB_MAXOUT_RXBC_TRIO0_B**: IDAC calibration maxout for trio B0 RXBC.
22	0x0	**IDAC_CALIB_MAXOUT_RXAB_TRIO0_B**: IDAC calibration maxout for trio B0 RXAB.
21	0x0	**IDAC_CALIB_MAXOUT_RXCA_TRIO1_A**: IDAC calibration maxout for trio A1 RXCA.
20	0x0	**IDAC_CALIB_MAXOUT_RXBC_TRIO1_A**: IDAC calibration maxout for trio A1 RXBC.
19	0x0	**IDAC_CALIB_MAXOUT_RXAB_TRIO1_A**: IDAC calibration maxout for trio A1 RXAB.
18	0x0	**IDAC_CALIB_MAXOUT_RXCA_TRIO0_A**: IDAC calibration maxout for trio A0 RXCA.
17	0x0	**IDAC_CALIB_MAXOUT_RXBC_TRIO0_A**: IDAC calibration maxout for trio A0 RXBC.
16	0x0	**IDAC_CALIB_MAXOUT_RXAB_TRIO0_A**: IDAC calibration maxout for trio A0 RXAB.
11	0x0	**IDAC_CALIB_DONE_RXCA_TRIO1_B**: IDAC calibration done for trio B1 RXCA.

Bit	Reset	Description
10	0x0	**IDAC_CALIB_DONE_RXBC_TRIO1_B**: IDAC calibration done for trio B1 RXBC.
9	0x0	**IDAC_CALIB_DONE_RXAB_TRIO1_B**: IDAC calibration done for trio B1 RXAB.
8	0x0	**IDAC_CALIB_DONE_RXCA_TRIO0_B**: IDAC calibration done for trio B0 RXCA.
7	0x0	**IDAC_CALIB_DONE_RXBC_TRIO0_B**: IDAC calibration done for trio B0 RXBC.
6	0x0	**IDAC_CALIB_DONE_RXAB_TRIO0_B**: IDAC calibration done for trio B0 RXAB.
5	0x0	**IDAC_CALIB_DONE_RXCA_TRIO1_A**: IDAC calibration done for trio A1 RXCA.
4	0x0	**IDAC_CALIB_DONE_RXBC_TRIO1_A**: IDAC calibration done for trio A1 RXBC.
3	0x0	**IDAC_CALIB_DONE_RXAB_TRIO1_A**: IDAC calibration done for trio A1 RXAB.
2	0x0	**IDAC_CALIB_DONE_RXCA_TRIO0_A**: IDAC calibration done for trio A0 RXCA.
1	0x0	**IDAC_CALIB_DONE_RXBC_TRIO0_A**: IDAC calibration done for trio A0 RXBC.
0	0x0	**IDAC_CALIB_DONE_RXAB_TRIO0_A**: IDAC calibration done for trio A0 RXAB.

NVCSI_PHY_3_NVCSI_CIL_IDAC_CALIB_STATUS1_0

Offset: 0x10717
Byte Offset: 0x41c5c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x21084210 (0bxx10,0001,0000,1000,0100,0010,0001,0000)

Bit	Reset	Description
29:25	0x10	**RXCA_IDACADJ_TRIO1_A**: IDAC calibration result for trio A1 RXCA
24:20	0x10	**RXBC_IDACADJ_TRIO1_A**: IDAC calibration result for trio A1 RXBC
19:15	0x10	**RXAB_IDACADJ_TRIO1_A**: IDAC calibration result for trio A1 RXAB

Bit	Reset	Description
14:10	0x10	**RXCA_IDACADJ_TRIO0_A**: IDAC calibration result for trio A0 RXCA
9:5	0x10	**RXBC_IDACADJ_TRIO0_A**: IDAC calibration result for trio A0 RXBC
4:0	0x10	**RXAB_IDACADJ_TRIO0_A**: IDAC calibration result for trio A0 RXAB

NVCSI_PHY_3_NVCSI_CIL_IDAC_CALIB_STATUS2_0

Offset: 0x10718
Byte Offset: 0x41c60
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x21084210 (0bxx10,0001,0000,1000,0100,0010,0001,0000)

Bit	Reset	Description
29:25	0x10	**RXCA_IDACADJ_TRIO1_B**: IDAC calibration result for trio B1 RXCA
24:20	0x10	**RXBC_IDACADJ_TRIO1_B**: IDAC calibration result for trio B1 RXBC
19:15	0x10	**RXAB_IDACADJ_TRIO1_B**: IDAC calibration result for trio B1 RXAB
14:10	0x10	**RXCA_IDACADJ_TRIO0_B**: IDAC calibration result for trio B0 RXCA
9:5	0x10	**RXBC_IDACADJ_TRIO0_B**: IDAC calibration result for trio B0 RXBC
4:0	0x10	**RXAB_IDACADJ_TRIO0_B**: IDAC calibration result for trio B0 RXAB

NVCSI_PHY_3_NVCSI_CIL_IDAC_OVERRIDE_A_0

Offset: 0x10719
Byte Offset: 0x41c64
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x84218421 (0b1000,0100,0010,0001,1000,0100,0010,0001)

Bit	Reset	Description
31:27	0x10	**IDAC_VALUE_RXCA_TRIO1_A**: The override value
26:22	0x10	**IDAC_VALUE_RXBC_TRIO1_A**: The override value
21:17	0x10	**IDAC_VALUE_RXAB_TRIO1_A**: The override value
16	0x1	**SW_SET_IDAC_TRIO1_A**: Enable the SW override of the IDAC value
15:11	0x10	**IDAC_VALUE_RXCA_TRIO0_A**: The override value
10:6	0x10	**IDAC_VALUE_RXBC_TRIO0_A**: The override value
5:1	0x10	**IDAC_VALUE_RXAB_TRIO0_A**: The override value
0	0x1	**SW_SET_IDAC_TRIO0_A**: Enable the SW override of the IDAC value

NVCSI_PHY_3_NVCSI_CIL_IDAC_OVERRIDE_B_0

Offset: 0x1071a
Byte Offset: 0x41c68
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x84218421 (0b1000,0100,0010,0001,1000,0100,0010,0001)

Bit	Reset	Description
31:27	0x10	**IDAC_VALUE_RXCA_TRIO1_B**: The override value
26:22	0x10	**IDAC_VALUE_RXBC_TRIO1_B**: The override value
21:17	0x10	**IDAC_VALUE_RXAB_TRIO1_B**: The override value
16	0x1	**SW_SET_IDAC_TRIO1_B**: Enable the SW override of the IDAC value
15:11	0x10	**IDAC_VALUE_RXCA_TRIO0_B**: The override value

Bit	Reset	Description
10:6	0x10	**IDAC_VALUE_RXBC_TRIO0_B**: The override value
5:1	0x10	**IDAC_VALUE_RXAB_TRIO0_B**: The override value
0	0x1	**SW_SET_IDAC_TRIO0_B**: Enable the SW override of the IDAC value

NVCSI_PHY_3_NVCSI_CIL_IDAC_CALIB_DEBUG_0

Offset: 0x1071b
Byte Offset: 0x41c6c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000XXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
11	X	**ZI_HSRX_RXCA_TRIO1_B**
10	X	**ZI_HSRX_RXBC_TRIO1_B**
9	X	**ZI_HSRX_RXAB_TRIO1_B**
8	X	**ZI_HSRX_RXCA_TRIO0_B**
7	X	**ZI_HSRX_RXBC_TRIO0_B**
6	X	**ZI_HSRX_RXAB_TRIO0_B**
5	X	**ZI_HSRX_RXCA_TRIO1_A**
4	X	**ZI_HSRX_RXBC_TRIO1_A**
3	X	**ZI_HSRX_RXAB_TRIO1_A**
2	X	**ZI_HSRX_RXCA_TRIO0_A**
1	X	**ZI_HSRX_RXBC_TRIO0_A**
0	X	**ZI_HSRX_RXAB_TRIO0_A**

NVCSI_PHY_3_NVCSI_CIL_TEST_CONTROL_0

Offset: 0x1071c
Byte Offset: 0x41c70

Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**HS_EN_DATA_LANE1_B**: Force to enable the e_input_hs of the data lane b1, used for impedance measurements of HS mode
4	0x0	**HS_EN_DATA_LANE0_B**: Force to enable the e_input_hs of the data lane b0, used for impedance measurements of HS mode
3	0x0	**HS_EN_DATA_LANE1_A**: Force to enable the e_input_hs of the data lane a1, used for impedance measurements of HS mode
2	0x0	**HS_EN_DATA_LANE0_A**: Force to enable the e_input_hs of the data lane a0, used for impedance measurements of HS mode
1	0x0	**HS_EN_CLK_LANE_B**: Force to enable the e_input_hs of the clock lane b, used for impedance measurements of HS mode
0	0x0	**HS_EN_CLK_LANE_A**: Force to enable the e_input_hs of the clock lane a, used for impedance measurements of HS mode

NVCSI_PHY_3_NVCSI_CIL_PULLDN_CONTROL_0

Offset: 0x1071d
Byte Offset: 0x41c74
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**E_PULLDN_IO1_B**
4	0x0	**E_PULLDN_IO0_B**
3	0x0	**E_PULLDN_IO1_A**
2	0x0	**E_PULLDN_IO0_A**
1	0x0	**E_PULLDN_CLK_B**

Bit	Reset	Description
0	0x0	**E_PULLDN_CLK_A**

NVCSI_PHY_3_NVCSI_CIL_SPARE_0

Spare register
Offset: 0x1071e
Byte Offset: 0x41c78
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**phy_reg**

NVCSI_PHY_3_NVCSI_CIL_A_SW_RESET_0

Offset: 0x1071f
Byte Offset: 0x41c7c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**SW_RESET1_A**: SOFT RESET FOR LANE A1. Active High **1** = ENABLE **0** = DISABLE
0	0x0	**SW_RESET0_A**: SOFT RESET FOR LANE A0. Active High **1** = ENABLE **0** = DISABLE

NVCSI_PHY_3_NVCSI_CIL_A_CLKEN_OVERRIDE_CTRL_0

Offset: 0x10720
Byte Offset: 0x41c80
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**CLKEN_OVERRIDE1_A**: CLKENABLE OVERRIDE FOR LANE A1. Active High **1** = ENABLE **0** = DISABLE
0	0x0	**CLKEN_OVERRIDE0_A**: CLKENABLE OVERRIDE FOR LANE A0. Active High **1** = ENABLE **0** = DISABLE

NVCSI_PHY_3_NVCSI_CIL_A_CTLE_CTRL_0

Offset: 0x10721
Byte Offset: 0x41c84
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000888 (0bxxxx,xxxx,0000,0000,0000,1000,1000,1000)

Bit	Reset	Description
23:20	0x0	**AFE_DCGAIN_IO1_A**: RX AFE DC gain control
19:16	0x0	**AFE_DCGAIN_IO0_A**: RX AFE DC gain control
15:12	0x0	**AFE_DCGAIN_CLK_A**: RX AFE DC gain control
11:8	0x8	**AFE_HFGAIN_IO1_A**: RX AFE_CTLE high frequency gain control, 16 curves, ac_gain=code*1db
7:4	0x8	**AFE_HFGAIN_IO0_A**: RX AFE_CTLE high frequency gain control, 16 curves, ac_gain=code*1db

Bit	Reset	Description
3:0	0x8	**AFE_HFGAIN_CLK_A**: RX AFE_CTLE high frequency gain control, 16 curves, ac_gain=code*1db

NVCSI_PHY_3_NVCSI_CIL_A_CTLE_CTRL1_0

Offset: 0x10722
Byte Offset: 0x41c88
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000554 (0bxxxx,xxxx,0000,0000,0000,0101,0101,0100)

Bit	Reset	Description
23:20	0x0	**AFE_CTRL_IO1_A**
19:16	0x0	**AFE_CTRL_IO0_A**
15:12	0x0	**AFE_CTRL_CLK_A**
11:10	0x1	**AFE_CURRENT_IO1_A**
9:8	0x1	**AFE_CURRENT_IO0_A**
7:6	0x1	**AFE_CURRENT_CLK_A**
5:4	0x1	**AFE_FREQBAND_IO1_A**: RX AFE bandwidth selection. 2'b00: <1.5GS/s, 2'b01: 1.5~3GS/s, 2'b10: 3~5GS/s, 2'b11: reserved.
3:2	0x1	**AFE_FREQBAND_IO0_A**: RX AFE bandwidth selection. 2'b00: <1.5GS/s, 2'b01: 1.5~3GS/s, 2'b10: 3~5GS/s, 2'b11: reserved.
1:0	0x0	**AFE_FREQBAND_CLK_A**: RX AFE bandwidth selection. 2'b00: <1.5GS/s, 2'b01: 1.5~3GS/s, 2'b10: 3~5GS/s, 2'b11: reserved.

NVCSI_PHY_3_NVCSI_CIL_A_PAD_CONFIG_0

Offset: 0x10723
Byte Offset: 0x41c8c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00070000 (0b0000,0000,0000,x111,xxxx,0000,0000,0000)

Bit	Reset	Description
31	0x0	**LP_RX_SELECT_IO1_A**: LP mode receiver select for partition A lane 1. DPHY mode: controls IO1P_A and IO1N_A CPHY mode: controls IO1P_A, IO1N_A and IOCLKN_A.
30	0x0	**LP_RX_SELECT_IO0_A**: LP mode receiver select for partition A lane 0. DPHY mode: controls IO0P_A and IO0N_A CPHY mode: controls IO0P_A, IO0N_A and IOCLKP_A.
29	0x0	**LP_RX_SELECT_CLK_A**: LP mode receiver select. 0: Schmitt receiver. 1: self-biased receiver. DPHY mode: controls IOCLKP_A and IOCLKN_A. CPHY mode: dummy.
28	0x0	**E_LPRX_HYS_N_IO1_A**: Enable LP-RX hysteresis. 0: enable hysteresis 1: disable hysteresis
27	0x0	**E_LPRX_HYS_N_IO0_A**: Enable LP-RX hysteresis. 0: enable hysteresis 1: disable hysteresis
26	0x0	**E_LPRX_HYS_N_CLK_A**: Enable LP-RX hysteresis. 0: enable hysteresis 1: disable hysteresis
25	0x0	**E_HS_TERM_N_IO1_A**: Enable the HS termination of Partition A Lane 1.
24	0x0	**E_HS_TERM_N_IO0_A**: Enable the HS termination of Partition A Lane 0.
23	0x0	**E_HS_TERM_N_CLK_A**: Enable the HS termination of Clock partition A.
22	0x0	**E_INPUT_LP_IO1_A**: Enable LP receiver of Partition A Lane 1 Applicable in both CPHY and DPHY case.
21	0x0	**E_INPUT_LP_IO0_A**: Enable LP receiver of Partition A Lane 0 Applicable in both CPHY and DPHY case.
20	0x0	**E_INPUT_LP_CLK_A**: enable LP receiver of Clock partition A Applicable in DPHY case. N/A for CPHY
18	0x1	**PD_CLK_A**: Power down for CLk of Partition A. Applicable in DPHY case. N/A for CPHY
17	0x1	**PD_IO1_A**: Power down for Trio 1 and Lane 1 of Partition A. Applicable in both CPHY and DPHY case.

Bit	Reset	Description
16	0x1	**PD_IO0_A**: Power down for Trio 0 and Lane 0 of Partition A. Applicable in both CPHY and DPHY case.
11:8	0x0	**SPARE_CLK_A**: Spare control bits for pad control. Bit 11 is used as E_PULLDN_CLK_A control (0 - Pull down of Clock Lane enabled, 1 - Pull down of Clock Lane enabled). Functionality of other bits currently not defined. Undefined register bits are reserved for future use. Pull down may be enabled when Clock Lane is unused to minimized interference.
7:4	0x0	**SPARE_IO1_A**: Spare control bits for pad control. Functionality of other bits currently not defined. Undefined register bits are reserved for future use. Pull down may be enabled when Data Lane-1 is unused to minimized interference.
3:0	0x0	**SPARE_IO0_A**: Spare control bits for pad control. Functionality of other bits currently not defined. Undefined register bits are reserved for future use. Pull down may be enabled when Data Lane-0 is unused to minimized interference.

NVCSI_PHY_3_NVCSI_CIL_A_PAD_CONFIG_1_0

Offset: 0x10724
Byte Offset: 0x41c90
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000002 (0b0000,0000,0000,0000,0000,0000,xx00,0010)

Bit	Reset	Description
31:24	0x0	**OUTADJ_IO1_A**: Trio 1 output delay trimmer, each tap delays 9ps
23:16	0x0	**OUTADJ_IO0_A**: Trio 0 output delay trimmer, each tap delays 9ps
15:8	0x0	**OUTADJ_CLK_A**: Clock bit output delay trimmer, each tap delays 9ps
5	0x0	**HS_BSO_A**: 1= power-on 0.4v-TX-regulator during LP-mode; 0= power-off 0.4v-TX-regulator during LP-mode;
4	0x0	**CPHY_MID_STRENGTH_TRIO1_A**: 0: when bump is driven to mid-voltage-level, both 100-ohm-pull-up and 100-ohm-pull-down is turned on to drive mid-voltage-level; 1: when bump is driven to mid-voltage-level, both pull-up and pull-down are turned off to become tri-stated;

Bit	Reset	Description
3	0x0	**CPHY_MID_STRENGTH_TRIO0_A**: 0: when bump is driven to mid-voltage-level, both 100-ohm-pull-up and 100-ohm-pull-down is turned on to drive mid-voltage-level; 1: when bump is driven to mid-voltage-level, both pull-up and pull-down are turned off to become tri-stated;
2	DISABLE	**REV_CLK_A**: Reverse clock polarity **0** = DISABLE **1** = ENABLE
1	ENABLE	**E_PREDRV_FLOP_RESET_A**: HS predriver flop reset **0** = DISABLE **1** = ENABLE
0	DISABLE	**FCZERO_A**: Force clk bit to drive differential 0 **0** = DISABLE **1** = ENABLE

NVCSI_PHY_3_NVCSI_CIL_A_PAD_CONFIG_2_0

Offset: 0x10725
Byte Offset: 0x41c94
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,0000,0000,0000,xxxx,0000,0000,0000)

Bit	Reset	Description
27:24	0x0	**PEMPD_IO1_A**: Enable trio 1 HS driver pull down pre-emphasis.
23:20	0x0	**PEMPD_IO0_A**: Enable trio 0 HS driver pull down pre-emphasis.
19:16	0x0	**PEMPD_CLK_A**: Enable clock bit HS driver pull down pre-emphasis.
11:8	0x0	**PEMPU_IO1_A**: Enable trio 1 HS driver pull up pre-emphasis.
7:4	0x0	**PEMPU_IO0_A**: Enable trio 0 HS driver pull up pre-emphasis.
3:0	0x0	**PEMPU_CLK_A**: Enable clock bit HS driver pull up pre-emphasis.

NVCSI_PHY_3_NVCSI_CIL_A_PAD_CONFIG_3_0

Offset: 0x10726
Byte Offset: 0x41c98
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x06660666 (0bxxxx,0110,0110,0110,xxxx,0110,0110,0110)

Bit	Reset	Description
27:24	0x6	**LPDNADJ_IO1_A**: Driver pull down impedance control
23:20	0x6	**LPDNADJ_IO0_A**: Driver pull down impedance control
19:16	0x6	**LPDNADJ_CLK_A**: Driver pull down impedance control
11:8	0x6	**LPUPADJ_IO1_A**: Driver pull up impedance control
7:4	0x6	**LPUPADJ_IO0_A**: Driver pull up impedance control
3:0	0x6	**LPUPADJ_CLK_A**: Driver pull up impedance control

NVCSI_PHY_3_NVCSI_CIL_A_PAD_CONFIG_4_0

Offset: 0x10727
Byte Offset: 0x41c9c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,0000,0000,0000,xxxx,0000,0000,0000)

Bit	Reset	Description
27:24	0x0	**SLEWDNADJ_IO1_A**: Pull down slew rate adjust
23:20	0x0	**SLEWDNADJ_IO0_A**: Pull down slew rate adjust
19:16	0x0	**SLEWDNADJ_CLK_A**: Pull down slew rate adjust

Bit	Reset	Description
11:8	0x0	**SLEWUPADJ_IO1_A**: Pull up slew rate adjust
7:4	0x0	**SLEWUPADJ_IO0_A**: Pull up slew rate adjust
3:0	0x0	**SLEWUPADJ_CLK_A**: Pull up slew rate adjust

NVCSI_PHY_3_NVCSI_CIL_A_PAD_CONFIG_5_0

Offset: 0x10728
Byte Offset: 0x41ca0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000110 (0bxxxx,xxxx,xxxx,xxxx,xxxx,x001,0001,0000)

Bit	Reset	Description
10:7	0x2	**CDDNADJ_IO1_A**: Level adjust on low limit of detection
6:3	0x2	**CDDNADJ_IO0_A**: Level adjust on low limit of detection
2	DISABLE	**E_CD_IO1_A**: Data bits contention detector enable **0** = DISABLE **1** = ENABLE
1	DISABLE	**E_CD_IO0_A**: Data bits contention detector enable **0** = DISABLE **1** = ENABLE
0	DISABLE	**E_CD_CLK_A**: Clock bit contention detector enable **0** = DISABLE **1** = ENABLE

NVCSI_PHY_3_NVCSI_CIL_A_PAD_CONFIG_6_0

Offset: 0x10729
Byte Offset: 0x41ca4
Read/Write: R/W

Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**E_DEEP_LPBK_A**: Enable deep loopback
0	0x0	**E_SHALLOW_LPBK_A**: Enalbe shallow loopback

NVCSI_PHY_3_NVCSI_CIL_A_PAD_CD_STATUS_0

Offset: 0x1072a
Byte Offset: 0x41ca8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**CDN_IO1_A**: trio1 n output, 1=contention detected.
4	0x0	**CDN_IO0_A**: trio0 n output, 1=contention detected.
3	0x0	**CDN_CLK_A**: Clock bit n output, 1=contention detected.
2	0x0	**CDP_IO1_A**: trio1 p output, 1=contention detected.
1	0x0	**CDP_IO0_A**: trio0 p output, 1=contention detected.
0	0x0	**CDP_CLK_A**: Clock bit p output, 1=contention detected.

NVCSI_PHY_3_NVCSI_CIL_A_DPHY_INADJ_CTRL_0

Offset: 0x1072b
Byte Offset: 0x41cac
Read/Write: R/W
Parity Protection: Y

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,x000,0000,x000,0000,x000,0000)

Bit	Reset	Description
22	0x0	**SW_SET_DPHY_INADJ_CLK_A**: Software set for clock input delay trimmer
21:16	0x0	**DPHY_INADJ_CLK_A**: Programmable value for CLK input delay trimmer,
14	0x0	**SW_SET_DPHY_INADJ_IO1_A**: Software override for bit 1 input delay trimmer 1 - SW programmed value is applied to the pads. 2 - Trimmer value from HW calibration state machine is applied to the pads
13:8	0x0	**DPHY_INADJ_IO1_A**: Programmable value for bit 1 input delay trimmer, each tap delay of 9 ps
6	0x0	**SW_SET_DPHY_INADJ_IO0_A**: Software override for bit 0 input delay trimmer 1 - SW programmed value is applied to the pads. 2 - Trimmer value from HW calibration state machine is applied to the pads
5:0	0x0	**DPHY_INADJ_IO0_A**: Programmable value for bit 0 input delay trimmer,

NVCSI_PHY_3_NVCSI_CIL_A_CLK_DESKEW_CTRL_0

Offset: 0x1072c
Byte Offset: 0x41cb0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000bf00 (0bxxxx,xxxx,xxxx,xxxx,1x11,1111,xx00,0000)

Bit	Reset	Description
15	ENABLE	**CLK_INADJ_SWEEP_CTRL_A**: Enable the clock trimmer sweep for D-PHY de-skew. **0** = DISABLE **1** = ENABLE
13:8	0x3f	**CLK_INADJ_LIMIT_HIGH_A**: The upper limit of the sweep range for trimmer sweep.
5:0	0x0	**CLK_INADJ_LIMIT_LOW_A**: The lower limit of the sweep range for trimmer sweep.

NVCSI_PHY_3_NVCSI_CIL_A_DATA_DESKEW_CTRL_0

Offset: 0x1072d
Byte Offset: 0x41cb4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x9f80bf00 (0b1xx1,1111,1x00,0000,1x11,1111,xx00,0000)

Bit	Reset	Description
31	0x1	**DATA_INADJ_SWEEP_CTRL1_A**: Enable the data lane A1 trimmer sweep for D-PHY de-skew.
28:23	0x3f	**DATA_INADJ_LIMIT_HIGH1_A**: The upper limit of the sweep range for trimmer sweep.
21:16	0x0	**DATA_INADJ_LIMIT_LOW1_A**: The lower limit of the sweep range for trimmer sweep.
15	0x1	**DATA_INADJ_SWEEP_CTRL0_A**: Enable the data lane A0 trimmer sweep for D-PHY de-skew.
13:8	0x3f	**DATA_INADJ_LIMIT_HIGH0_A**: The upper limit of the sweep range for trimmer sweep.
5:0	0x0	**DATA_INADJ_LIMIT_LOW0_A**: The lower limit of the sweep range for trimmer sweep.

NVCSI_PHY_3_NVCSI_CIL_A_DPHY_DESKEW_STATUS_0

Offset: 0x1072e
Byte Offset: 0x41cb8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000X0XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
15	X	**DPHY_CALIB_ERR_IO1_A**: calib error status
14	X	**DPHY_CALIB_DONE_IO1_A**: calib done status
7	X	**DPHY_CALIB_ERR_IO0_A**: calib error status

Bit	Reset	Description
6	X	**DPHY_CALIB_DONE_IO0_A**: calib done status
1	X	**DPHY_CALIB_ERR_CTRL_A**: calib error status
0	X	**DPHY_CALIB_DONE_CTRL_A**: calib done status

NVCSI_PHY_3_NVCSI_CIL_A_DPHY_DESKEW_DATA_CALIB_STATUS_LOW_0_0

Offset: 0x1072f
Byte Offset: 0x41cbc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_A**

NVCSI_PHY_3_NVCSI_CIL_A_DPHY_DESKEW_DATA_CALIB_STATUS_HIGH_0_0

Offset: 0x10730
Byte Offset: 0x41cc0
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_A**

NVCSI_PHY_3_NVCSI_CIL_A_DPHY_DESKEW_DATA_CALIB_STATUS_LOW_1_0

Offset: 0x10731
Byte Offset: 0x41cc4
Read/Write: RO
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_A**

NVCSI_PHY_3_NVCSI_CIL_A_DPHY_DESKEW_DATA_CALIB_STATUS_HIGH_1_0

Offset: 0x10732
Byte Offset: 0x41cc8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_A**

NVCSI_PHY_3_NVCSI_CIL_A_DPHY_DESKEW_CLK_CALIB_STATUS_LOW_0_0

Offset: 0x10733
Byte Offset: 0x41ccc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_A**

NVCSI_PHY_3_NVCSI_CIL_A_DPHY_DESKEW_CLK_CALIB_STATUS_HIGH_0_0

Offset: 0x10734
Byte Offset: 0x41cd0
Read/Write: RO
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_A**

NVCSI_PHY_3_NVCSI_CIL_A_DPHY_DESKEW_CLK_CALIB_STATUS_LOW_1_0

Offset: 0x10735
Byte Offset: 0x41cd4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_A**

NVCSI_PHY_3_NVCSI_CIL_A_DPHY_DESKEW_CLK_CALIB_STATUS_HIGH_1_0

Offset: 0x10736
Byte Offset: 0x41cd8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_A**

NVCSI_PHY_3_NVCSI_CIL_A_DPHY_DESKEW_RESULT_STATUS_0

Offset: 0x10737
Byte Offset: 0x41cdc
Read/Write: R/W
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	DONE	**DESKEW_RESULT_STATUS1_A**: For lane A1. The result status of the deskew calibration, HW check this field before start the trimmer sweep, if it's 1'b1, the trimmer sweep will not perform. HW set this bit to 1 when the calibration is done. SW should clear this field after it read out the 2 64 bits calibration status. **0** = DONE **1** = PENDING
0	DONE	**DESKEW_RESULT_STATUS0_A**: For lane A0. The result status of the deskew calibration, HW check this field before start the trimmer sweep, if it's 1'b1, the trimmer sweep will not perform. HW set this bit to 1 when the calibration is done. SW should clear this field after it read out the 2 64 bits calibration status. **0** = DONE **1** = PENDING

NVCSI_PHY_3_NVCSI_CIL_A_POLARITY_SWIZZLE_CTRL_0

Offset: 0x10738
Byte Offset: 0x41ce0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xx00,0000,xxxx,x000)

Bit	Reset	Description
13:11	0x0	**POLARITY_SWIZZLE_CPHY1_A**: Polarity Swizle control for Lane A1 in CPHY mode. valid only in CPHY mode 000 - A B C --> A B C 001 - A B C --> A C B 010 - A B C --> B C A 011 - A B C --> B A C 100 - A B C --> C A B 101 - A B C --> C B A
10:8	0x0	**POLARITY_SWIZZLE_CPHY0_A**: Polarity Swizle control for Lane A0 in CPHY mode. valid only in CPHY mode 000 - A B C --> A B C 001 - A B C --> A C B 010 - A B C --> B C A 011 - A B C --> B A C 100 - A B C --> C A B 101 - A B C --> C B A

Bit	Reset	Description
2	0x0	**POLARITY_SWIZZLE_CLK_A**: Polarity Swizzle control for clock lane A. Valid only in DPHY mode, need to work with ALLOW_FIRST_BIT_ON_CLK_NEGEDGE if enabled.
1	0x0	**POLARITY_SWIZZLE_DPHY1_A**: Polarity Swizzle control for data lane A1. Valid only in DPHY mode.
0	0x0	**POLARITY_SWIZZLE_DPHY0_A**: Polarity Swizzle control for data lane A0. Valid only in DPHY mode.

NVCSI_PHY_3_NVCSI_CIL_A_DESKEW_CONTROL_0

Offset: 0x10739
Byte Offset: 0x41ce4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00007411 (0bxxxx,xxxx,xxxx,xxxx,0111,0100,0001,0001)

Bit	Reset	Description
15:13	0x3	**DESKEW_DATA_DELAY**: How many cycles delay introduced by re-timing flops
12	0x1	**DESKEW_LANE_SKEW_CHECK_EN**: Check the skew between the lanes, this is used to handle the error case that error happen in the CSI link caused some data lanes detect the HS sync word (0xb8) while other data lanes detect the deskew sync word (0xff). If this bit is not set with the error case, the lane FIFO will overflow and reset is required. With this bit set, the FIFO will be auto flused and the FIFO overflow will not happen.
11:8	0x4	**DESKEW_LANE_SKEW_TOLERANCE**: The allowed skew between the data lane, only used in sync word search
7:4	0x1	**DESKEW_COMPARE**: Regsiter select to control the number of comparisons to be done for each trimmer setting during deskew calibration.
3:0	0x1	**DESKEW_SETTLE**: Regsiter select to control the number of byte clocks to wait before INADJ value settles. This is used during deskew calibration

NVCSI_PHY_3_NVCSI_CIL_A_CONTROL_0

Offset: 0x1073a
Byte Offset: 0x41ce8
Read/Write: R/W
Parity Protection: Y

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**BYPASS_LP_FILTER**: Bypass the glitch filter.
30:24	0x0	**CLK_SETTLE1_A**: Used only in CPHY mode, Settle time for clk start when moving A1 data lane from LP to HS (LP11->LP01->LP00).
23:17	0x0	**CLK_SETTLE0_A**: Used for both CPHY/DPY mode. For DPHY: Settle time for clk lane A0 when moving from LP to HS (LP11->LP01->LP00), this setting determines how many csicil clock cycles to wait after LP00. If the settle set to 0 or 8, the real setting is 1, otherwise, the real setting is (4+settle) For CPHY: Settle time for clk start when moving A0 data lane from LP to HS (LP11->LP01->LP00).
16:9	0x0	**THS_SETTLE1_A**: settle time for A1 data lane when moving from LP to HS (LP11->LP01->LP00), this setting determines how many LP clock cycles to wait, after LP00,before starting to look at the data.
8:1	0x0	**THS_SETTLE0_A**: settle time for A0 data lane when moving from LP to HS (LP11->LP01->LP00), this setting determines how many LP clock cycles to wait, after LP00,before starting to look at the data.
0	0x0	**BYPASS_LP_SEQ**: For DPHY: The clock lane LP signals should sequence through LP11->LP01->LP00 state, to indicate to CLOCK CIL about the mode switching to HS Rx mode. In case Camera is enabled earlier than CIL , it is highly likely that camera sends this control sequence sooner than cil can detect it. Enabling this bit allows the CLOCK CIL to overlook the LP control sequence and step in HS Rx mode directly looking at LP00 only. For CPHY: it's only used for debug purpose as it's embedded clock for CPHY, clock will always presents when the data lane toggle.

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_ERR_STATUS_0

Offset: 0x1073b
Byte Offset: 0x41cec
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
29:24	X	**LANE_DEMAPPER_ERR_MUX1_A**: Debug register, error detected in CPHY de-mapper mux.
23	X	**LANE_DEMAPPER_ERR_RXCTRL1_A**: Debug register, Error detected in CPHY de-mapper rx ctrl.
22:16	X	**LANE_DECODER_ERR1_A**: Debug register, Error detected in CPHY de-coder.
13:8	X	**LANE_DEMAPPER_ERR_MUX0_A**: Debug register, Error detected in CPHY de-mapper mux.
7	X	**LANE_DEMAPPER_ERR_RXCTRL0_A**: Debug register, Error detected in CPHY de-mapper rx ctrl.
6:0	X	**LANE_DECODER_ERR0_A**: Debug register, Error detected in CPHY de-coder.

NVCSI_PHY_3_NVCSI_CIL_A_ESCAPE_MODE_COMMAND_0_0

Offset: 0x1073c
Byte Offset: 0x41cf0
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_COMMAND0_A**: The received command from escape mode.

NVCSI_PHY_3_NVCSI_CIL_A_ESCAPE_MODE_DATA_0_0

Offset: 0x1073d
Byte Offset: 0x41cf4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_DATA0_A**: The received data from excape mode.

NVCSI_PHY_3_NVCSI_CIL_A_ESCAPE_MODE_COMMAND_1_0

Offset: 0x1073e
Byte Offset: 0x41cf8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_COMMAND1_A**: The received command from escape mode.

NVCSI_PHY_3_NVCSI_CIL_A_ESCAPE_MODE_DATA_1_0

Offset: 0x1073f
Byte Offset: 0x41cfc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_DATA1_A**: The received data from excape mode.

NVCSI_PHY_3_NVCSI_CIL_A_DPHY_SYNC_PATTERN_0

Offset: 0x10740
Byte Offset: 0x41d00
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01)

Bit	Reset	Description
1	0x0	**ALLOW_FIRST_BIT_ON_CLK_NEGEDGE_A**: Allow shift 1/3/5/7 bit to search the sync word. Normal case only shift 0/2/4/6 bit to search sync word.
0	0x1	**DISABLE_SB_ERR_IN_SYNC_A**: Allow one single bit error in sync word 6MSB

NVCSI_PHY_3_NVCSI_CIL_A_DPHY_DESKEW_SYNC_PATTERN_0

Offset: 0x10741
Byte Offset: 0x41d04
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**INVERT_DESKEW_PATTERN**: Invert the de-skew pattern.

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_SYNC_SEARCH_0

Offset: 0x10742
Byte Offset: 0x41d08
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01080108 (0bxxxx,xxx1,0000,1000,xxxx,xxx1,0000,1000)

Bit	Reset	Description
24	0x1	**COUNT_EN_TRIO1_A**: Check if the symbol is pre-amble before sync word.
23:16	0x8	**PERIOD_TRIO1_A**: For CPHY, how many cycles allowed for non pre-amble symbol before sync word.

Bit	Reset	Description
8	ENABLE	**COUNT_EN_TRIO0_A**: Check if the symbol is pre-amble before sync word. **0** = DISABLE **1** = ENABLE
7:0	0x8	**PERIOD_TRIO0_A**: For CPHY, how many cycles allowed for non pre-amble symbol before sync word.

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_START_0

Offset: 0x10743
Byte Offset: 0x41d0c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**EDGE_DELAY_CAL_TRIGGER_TRIO1_A**: Trigger the CPHY clock recover calibration for lane A1 by write 1'b1, when the calibration complete, HW will clear this bit to 1'b0. Write 1'b0 to this be take no effect.
0	0x0	**EDGE_DELAY_CAL_TRIGGER_TRIO0_A**: Trigger the CPHY clock recover calibration for lane A0 by write 1'b1, when the calibration complete, HW will clear this bit to 1'b0. Write 1'b0 to this be take no effect.

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_CTRL_0

Offset: 0x10744
Byte Offset: 0x41d10
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x12011210 (0b0001,0010,0000,0001,0001,0010,0001,0000)
PROD: 0x08000800 (0bxxx0,10xx,xxxx,xxxx,xxx0,10xx,xxxx,xxxx)

Bit	Reset	PROD	Description
31	0x0	_NONE_	**TRIM_CAL_E_NONSTOP_TRIO1_A**: Reserved. Trimmer calibration background monitoring control. 1: enable. The accumulator keeps updating.
30:29	0x0	_NONE_	**TRIM_CAL_GAIN_SEL_TRIO1_A**: Trimmer calibration loop gain selection [0]: gain control 0: coarse tune (counter1) gain = 1, fine tune (counter2) gain = 1/4. For example: coarse tune: 1 UP -> trim_cal_code + 1; fine tune: 4 UPs -> trim_cal_code + 1. 1: coarse tune (counter1) gain = 1/2, fine tune (counter2) gain = 1/8 [1]: reserved
28:26	0x4	0x2	**TRIM_CAL_CNT_SEL_TRIO1_A**: Trimmer calibration counter selection [0]: counter1 (coarse tune) clock cycle. 0: 64T (128UI) 1: 128T (256UI) [2:1]: counter2 (fine tune) clock cycle. 00: 32T (64UI) 01: 64T (128UI) 10: 96T (192UI) 11: 128T (256UI) For high data rate (>3.5GS/s): 64T(cnt1)+96T(cnt2)=160T(320ui) -> cnt_sel=3'b100 For low data rate (<=3.5GS/s): 128T(cnt1)+32T(cnt2)=160T(320ui) -> cnt_se=3'b001
25	0x1	_NONE_	**TRIM_CAL_CPHY_ED_LOOPMODE_TRIO1_A**: Trimmer calibration and cphy edge detector loop mode control 0: exclude trimmer delay in 2nd round of reset loop. 1: include trimmer delay in 2nd round of reset loop. For low data rate.
24:16	0x1	_NONE_	**CALIB_LENGTH_TRIO1_A**: //The length of the peramble which use to do caliration. Count by nvcsi core clock. It's used to make sure the calibration is doing inside the preamble phase. If this length set to a small value, the calibration may can't complete within one packet and need several packets to complete.
15	0x0	_NONE_	**TRIM_CAL_E_NONSTOP_TRIO0_A**: Reserved. Trimmer calibration background monitoring control. 1: enable. The accumulator keeps updating.
14:13	0x0	_NONE_	**TRIM_CAL_GAIN_SEL_TRIO0_A**: Trimmer calibration loop gain selection [0]: gain control 0: coarse tune (counter1) gain = 1, fine tune (counter2) gain = 1/4. For example: coarse tune: 1 UP -> trim_cal_code + 1; fine tune: 4 UPs -> trim_cal_code + 1. 1: coarse tune (counter1) gain = 1/2, fine tune (counter2) gain = 1/8 [1]: reserved

Bit	Reset	PROD	Description
12:10	0x4	0x2	**TRIM_CAL_CNT_SEL_TRIO0_A**: Trimmer calibration counter selection [0]: counter1 (coarse tune) clock cycle. 0: 64T (128UI) 1: 128T (256UI) [2:1]: counter2 (fine tune) clock cycle. 00: 32T (64UI) 01: 64T (128UI) 10: 96T (192UI) 11: 128T (256UI) For high data rate (>3.5GS/s): 64T(cnt1)+96T(cnt2)=160T(320ui) -> cnt_sel=3'b100 For low data rate (<=3.5GS/s): 128T(cnt1)+32T(cnt2)=160T(320ui) -> cnt_se=3'b001
9	0x1	_NONE_	**TRIM_CAL_CPHY_ED_LOOPMODE_TRIO0_A**: Trimmer calibration and cphy edge detector loop mode control 0: exclude trimmer delay in 2nd round of reset loop. 1: include trimmer delay in 2nd round of reset loop. For low data rate.
8:0	0x10	_NONE_	**CALIB_LENGTH_TRIO0_A**: The length of the peramble which use to do caliration. Count by nvcsi core clock. It's used to make sure the calibration is doing inside the preamble phase. If this length set to a small value, the calibration may can't complete within one packet and need several packets to complete.

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_CTRL1_0

Offset: 0x10745
Byte Offset: 0x41d14
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0c0e0c0e (0bx000,1100,0000,1110,x000,1100,0000,1110)

Bit	Reset	Description
30	0x0	**PERIODIC_CALIB_TRIO1_A**: Enable the periodic calibration.
29	0x0	**CALIB_APPLY_IN_POST_TRIO1_A**: 1: Apply the calibration result when detect post symbol. 0: Apply the calibration result immediately when calibration done.

Bit	Reset	Description
28:27	0x1	**CALIB_APPLY_SELECT_TRIO1_A**: Which calibration result is applied to pad for dual calibration. 2'b11: Reserved 2'b10: Apply the middle value of two calibration to pad if the difference within threshold. 2'b01: Apply the 2nd calibration result to pad if it's difference with previous latch result within threshold. 2'b00: Apply the 1st calibration result to pad if it's difference with previous latch result within threshold.
26	0x1	**DUAL_CALIB_TRIO1_A**: Two times calibratiton. 1: Two time calibration with one trigger. 0: Single time calibration with one trigger
25:18	0x3	**TEMP_EDGE_DELAY_VALUE_TRIO1_A**: The edge delay value while doing the calibration.
17	0x1	**TEMP_EDGE_DELAY_SEL_TRIO1_A**: When the selection of edge delay value when do edge delay calibration. 1: Use the temp value from register setting. 0: Use the currently using setting. For the first calibration after reset, it's still using the temp value from register setting.
16	0x0	**BYPASS_CALIB_PACKET_TRIO1_A**: 1: Drop the packet data when do the calibration. 0: Not drop the packet data when do the calibration.
14	0x0	**PERIODIC_CALIB_TRIO0_A**: Enable the periodic calibration.
13	0x0	**CALIB_APPLY_IN_POST_TRIO0_A**: 1: Apply the calibration result when detect post symbol. 0: Apply the calibration result immediately when calibration done.
12:11	0x1	**CALIB_APPLY_SELECT_TRIO0_A**: Which calibration result is applied to pad for dual calibration. 2'b11: Reserved 2'b10: Apply the middle value of two calibration to pad if the difference within threshold. 2'b01: Apply the 2nd calibration result to pad if it's difference with previous latch result within threshold. 2'b00: Apply the 1st calibration result to pad if it's difference with previous latch result within threshold.
10	0x1	**DUAL_CALIB_TRIO0_A**: Two times calibratiton. 1: Two time calibration with one trigger. 0: Single time calibration with one trigger
9:2	0x3	**TEMP_EDGE_DELAY_VALUE_TRIO0_A**: The edge delay value apply to pad while doing the calibration.
1	0x1	**TEMP_EDGE_DELAY_SEL_TRIO0_A**: When the selection of edge delay value when do edge delay calibration. 1: Use the temp value from register setting. 0: Use the currently using setting. For the first calibration after reset, it's still using the temp value from register setting.
0	0x0	**BYPASS_CALIB_PACKET_TRIO0_A**: 1: Drop the packet data when do the calibration. 0: Not drop the packet data when do the calibration.

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_CTRL2_0

Offset: 0x10746
Byte Offset: 0x41d18
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01000100 (0bxxxx,xxx1,0000,0000,xxxx,xxx1,0000,0000)

Bit	Reset	Description
24	ENABLE	**TRIM_CAL_PD_HOLD_TRIO1_A**: Hold in PD **0** = DISABLE **1** = ENABLE
23	DISABLE	**TRIM_CAL_E_FIXEDDLY_TRIO1_A**: Enable fixed delay in trim calibration unit. **0** = DISABLE **1** = ENABLE
22:16	0x0	**TRIM_CAL_FIXEDDLY_TRIO1_A**: Adjust fixed delay. Delay = code*5ps(TT)/7ps(SS)
8	ENABLE	**TRIM_CAL_PD_HOLD_TRIO0_A**: Hold in PD **0** = DISABLE **1** = ENABLE
7	DISABLE	**TRIM_CAL_E_FIXEDDLY_TRIO0_A**: Enable fixed delay in trim calibration unit. **0** = DISABLE **1** = ENABLE
6:0	0x0	**TRIM_CAL_FIXEDDLY_TRIO0_A**: Adjust fixed delay. Delay = code*5ps(TT)/7ps(SS)

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_CTRL3_0

Offset: 0x10747
Byte Offset: 0x41d1c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffffffff	**CALIB_PERIOD_TRIO0_A**: The period of the periodic edge delay calibration, in nvcsi clock cycle number.

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_CTRL4_0

Offset: 0x10748
Byte Offset: 0x41d20
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffffffff	**CALIB_PERIOD_TRIO1_A**: The period of the periodic edge delay calibration, in nvcsi clock cycle number.

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_CTRL5_0

Offset: 0x10749
Byte Offset: 0x41d24
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000ffff (0bxxxx,xxxx,xxxx,xxxx,1111,1111,1111,1111)

Bit	Reset	Description
15:8	0xff	**ERR_THRESHOLD_TRIO1_A**: The threshold of drop the calibration result between current result and previous result. For dual calibration, check between the two calibration result in the same trigger. For signal calibration, check between the current calibration result and previous calibration result.
7:0	0xff	**ERR_THRESHOLD_TRIO0_A**: The threshold of drop the calibration result between current result and previous resulti. For dual calibration, check between the two calibration result in the same trigger. For signal calibration, check between the current calibration result and previous calibration result.

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_STATUS_0

Offset: 0x1074a
Byte Offset: 0x41d28
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,xxxx)

Bit	Reset	Description
9	0x0	**EDGE_DELAY_MISMATCH_TRIO1_A**: When the CALIB_APPLY_SELECT_TRIO is 0/1 or dual time calibration is not enabled, the calibration result difference from the previous latched calibration result is larger than threshold, this time calibration result is not apply to pad. When the CALIB_APPLY_SELECT_TRIO is 2, the calibration result difference of the two times calibration result is larger than threshold, this time calibration result is not apply to pad.
8	0x0	**EDGE_DELAY_MISMATCH_TRIO0_A**: When the CALIB_APPLY_SELECT_TRIO is 0/1 or dual time calibration is not enabled, the calibration result difference from the previous latched calibration result is larger than threshold, this time calibration result is not apply to pad. When the CALIB_APPLY_SELECT_TRIO is 2, the calibration result difference of the two times calibration result is larger than threshold, this time calibration result is not apply to pad.
7	0x0	**CALIB_DONE_1_TRIO1_A**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.
6	0x0	**CALIB_DONE_1_TRIO0_A**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.
5	0x0	**CALIB_DONE_0_TRIO1_A**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.
4	0x0	**CALIB_DONE_0_TRIO0_A**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_EDGE_DELAY_CTRL_0

Offset: 0x1074b
Byte Offset: 0x41d2c
Read/Write: R/W
Parity Protection: See table below
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x40000000 (0b01xx,x000,0000,0000,xxxx,x000,0000,0000)

Bit	Parity Protection	Reset	Description
31:30	Y	0x1	**CPHY_CLKGEN_TRIGWIDTH_A**: Adjust AB/CB/CA pulse width in self-clock generator.
26	N	0x0	**UPDATE_OFFSET_TRIO1_A**: Trigger to apply the new offset value
25:17	Y	0x0	**DELAY_OFFSET_TRIO1_A**: Offset value of trio A1. Represents signed 2's complement number for calibration offset. Represents unsigned number if the field used as override value.
16	Y	0x0	**OVERRIDE_CAL_VAL_TRIO1_A**: Override for trio A1 0: Apply the DELAY_OFFSET as an offset to the calibration result. 1: Apply the DELAY_OFFSET value to the pad.
10	N	0x0	**UPDATE_OFFSET_TRIO0_A**: Trigger to apply the new offset value
9:1	Y	0x0	**DELAY_OFFSET_TRIO0_A**: Offset value of trio A0. Represents signed 2's complement number for calibration offset. Represents unsigned number if the field used as override value.
0	Y	0x0	**OVERRIDE_CAL_VAL_TRIO0_A**: Override for trio A0 0: Apply the DELAY_OFFSET as an offset to the calibration result. 1: Apply the DELAY_OFFSET value to the pad.

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_EDGE_DELAY_VALUE_0

Offset: 0x1074c
Byte Offset: 0x41d30
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000303 (0bxxxx,xxxx,xxxx,xxxx,0000,0011,0000,0011)

Bit	Reset	Description
15:8	0x3	**EDGE_DELAY_VALUE_TRIO1_A**: The edge delay value apply to pad finally
7:0	0x3	**EDGE_DELAY_VALUE_TRIO0_A**: The edge delay value apply to pad finally

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_EDGE_DELAY_CALIB_RESULT_0

Offset: 0x1074d
Byte Offset: 0x41d34
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**EDGE_DELAY_CALIB_RESULT_1_TRIO1_A**: The edge delay calibrate result, with offset.
23:16	0x0	**EDGE_DELAY_CALIB_RESULT_1_TRIO0_A**: The edge delay calibrate result, with offset.
15:8	0x0	**EDGE_DELAY_CALIB_RESULT_0_TRIO1_A**: The edge delay calibrate result, with offset.
7:0	0x0	**EDGE_DELAY_CALIB_RESULT_0_TRIO0_A**: The edge delay calibrate result, with offset.

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_INADJ_CALIB_CTRL_0

Offset: 0x1074e
Byte Offset: 0x41d38
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
16	0x0	**INADJ_CAL_TRIGGER_TRIO1_A**: Trigger INADJ calibration for trio0
0	0x0	**INADJ_CAL_TRIGGER_TRIO0_A**: Trigger INADJ calibration for trio1

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_INADJ_CALIB_CTRL1_0

Offset: 0x1074f
Byte Offset: 0x41d3c

Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00010001 (0b0000,0000,0000,0001,0000,0000,0000,0001)

Bit	Reset	Description
31:16	0x1	**PRBS9_SEED_TRIO1_A**: PRBS9 initial seed for trio0
15:0	0x1	**PRBS9_SEED_TRIO0_A**: PRBS9 initial seed for trio1

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_INADJ_CALIB_CTRL2_0

Offset: 0x10750
Byte Offset: 0x41d40
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0fc00fc0 (0bxxxx,1111,1100,0000,xxxx,1111,1100,0000)

Bit	Reset	Description
27:22	0x3f	**INADJ_LIMIT_HIGH_TRIO1_A**: The upper limit for the INADJ sweep.
21:16	0x0	**INADJ_LIMIT_LOW_TRIO1_A**: The lower limit for the INADJ sweep.
11:6	0x3f	**INADJ_LIMIT_HIGH_TRIO0_A**: The upper limit for the INADJ sweep.
5:0	0x0	**INADJ_LIMIT_LOW_TRIO0_A**: The lower limit for the INADJ sweep.

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_INADJ_CALIB_STATUS_0

Offset: 0x10751
Byte Offset: 0x41d44
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**INADJ_CALIB_DONE_TRIO1_A**: Trio a1 inadj calibration done.
0	0x0	**INADJ_CALIB_DONE_TRIO0_A**: Trio a0 inadj calibration done.

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_INADJ_CALIB_STATUS1_0

Offset: 0x10752
Byte Offset: 0x41d48
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_LOW_TRIO0_A**: Trio a0 PRBS check status for lower 32 INADJ value.

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_INADJ_CALIB_STATUS2_0

Offset: 0x10753
Byte Offset: 0x41d4c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_LOW_TRIO1_A**: Trio a1 PRBS check status for lower 32 INADJ value.

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_INADJ_CALIB_STATUS3_0

Offset: 0x10754
Byte Offset: 0x41d50

Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_HIGH_TRIO0_A**: Trio a0 PRBS check status for upper 32 INADJ value.

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_INADJ_CALIB_STATUS4_0

Offset: 0x10755
Byte Offset: 0x41d54
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_HIGH_TRIO1_A**: Trio a1 PRBS check status for upper 32 INADJ value.

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_INADJ_OVERRIDE_0

Offset: 0x10756
Byte Offset: 0x41d58
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00080008 (0bxxxx,xxx0,0000,1000,xxxx,xxx0,0000,1000)

Bit	Reset	Description
24:17	0x4	**CPHY_INADJ_TRIO1_A**: The SW override INADJ value.
16	DISABLE	**SW_SET_CPHY_INADJ_TRIO1_A**: Enable SW override of the INADJ. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
8:1	0x4	**CPHY_INADJ_TRIO0_A**: The SW override INADJ value.
0	DISABLE	**SW_SET_CPHY_INADJ_TRIO0_A**: Enable SW override of the INADJ. **0** = DISABLE **1** = ENABLE

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_BIST_CTRL_0

Offset: 0x10757
Byte Offset: 0x41d5c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	0x0	**STOP_BIST_TRIO1_A**: Manually stop the A1 BIST, write 1 to stop the RX bist.
2	0x0	**STOP_BIST_TRIO0_A**: Manually stop the A0 BIST, write 1 to stop the RX bist.
1	0x0	**TRIGGER_BIST_TRIO1_A**: Trigger the trio A1 BIST, write 1 to trigger the RX bist, HW will clear it to 0 when the BIST is done, write 0 to this field is with no effect.
0	0x0	**TRIGGER_BIST_TRIO0_A**: Trigger the trio A0 BIST, write 1 to trigger the RX bist, HW will clear it to 0 when the BIST is done, write 0 to this field is with no effect.

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_BIST_STATUS_0_0

Offset: 0x10758
Byte Offset: 0x41d60
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:8	0x0	**WORD_ERR_CNT_TRIO1_A**: The trio A1 number of word comparison failure, report for BIST mode 1.
7:0	0x0	**WORD_ERR_CNT_TRIO0_A**: The trio A0 number of word comparison failure, report for BIST mode 1.

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_BIST_STATUS_1_0

Offset: 0x10759
Byte Offset: 0x41d64
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:8	0x0	**SYM_ERR_CNT_TRIO1_A**: The trio A1 number of symbol compariosn failure, it's the sum of the symbol err count and the clock shift count, the real wire state error number is 1/2 of this field, report for BIST mode 1.
7:0	0x0	**SYM_ERR_CNT_TRIO0_A**: The trio A0 number of symbol compariosn failure, it's the sum of the symbol err count and the clock shift count, the real wire state error number is 1/2 of this field, report for BIST mode 1.

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_BIST_STATUS_2_0

Offset: 0x1075a
Byte Offset: 0x41d68
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WORD_CNT_LOW_TRIO0_A**: The trio A0 low 32 bits of the compared word number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_BIST_STATUS_3_0

Offset: 0x1075b
Byte Offset: 0x41d6c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WORD_CNT_HIGH_TRIO0_A**: The trio A0 high 16 bits of the compared word number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_BIST_STATUS_4_0

Offset: 0x1075c
Byte Offset: 0x41d70
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WORD_CNT_LOW_TRIO1_A**: The trio A1 low 32 bits of the compared word number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_BIST_STATUS_5_0

Offset: 0x1075d
Byte Offset: 0x41d74
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WORD_CNT_HIGH_TRIO1_A**: The trio A1 high 16 bits of the compared word number number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_3_NVCSI_CIL_A_CPHY_BIST_STATUS_6_0

Offset: 0x1075e
Byte Offset: 0x41d78
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,x000,0000,0000,xxxx,x000,0000,0000)

Bit	Reset	Description
26	0x0	**ERR_DETECT_FAIL_TRIO1_A**: Multi symbol error in one word (7 symbol) for trio A1, BIST logic is not able to detect the accurate wire state error. With this error, if not other errors detected, the word error is accurate.
25	0x0	**SHIFT_DETECT_FAIL_TRIO1_A**: Detect symbol error in two continuous word (14 symbols) for trio A1, BIST logic is not able to determine if symbol clock is slip. (if the symbol error count is not saturating, the result is still available with this error)
24	0x0	**SHIFT_OVERFLOW_TRIO1_A**: Too many clocks slip and the BIST logic is not able to re-align for trio A1. Typically, when this error is detect, the symbol error counter will saturate.
23:20	0x0	**DOUBLE_CLK_MISSED_CNT_TRIO1_A**: Number of the times for double symbol clock lost for trio A1.
19:16	0x0	**SINGLE_CLK_MISSED_CNT_TRIO1_A**: Number of the times for single symbol clock lost for trio A1.
10	0x0	**ERR_DETECT_FAIL_TRIO0_A**: Multi symbol error in one word (7 symbol) for trio A0, BIST logic is not able to detect the accurate wire state error. With this error, if not other errors detected, the word error is accurate.
9	0x0	**SHIFT_DETECT_FAIL_TRIO0_A**: Detect symbol error in two continuous word (14 symbols) for trio A0, BIST logic is not able to determine if symbol clock is slip. (if the symbol error count is not saturating, the result is still available with this error)
8	0x0	**SHIFT_OVERFLOW_TRIO0_A**: Too many clocks slip and the BIST logic is not able to re-align for trio A0. Typically, when this error is detect, the symbol error counter will saturate.
7:4	0x0	**DOUBLE_CLK_MISSED_CNT_TRIO0_A**: Number of the times for double symbol clock lost for trio A0.

Bit	Reset	Description
3:0	0x0	**SINGLE_CLK_MISSED_CNT_TRIO0_A**: Number of the times for single symbol clock lost for trio A0.

NVCSI_PHY_3_NVCSI_CIL_B_SW_RESET_0

Offset: 0x1075f
Byte Offset: 0x41d7c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**SW_RESET1_B**: SOFT RESET FOR LANE B1. Active High **1** = ENABLE **0** = DISABLE
0	0x0	**SW_RESET0_B**: SOFT RESET FOR LANE B0. Active High **1** = ENABLE **0** = DISABLE

NVCSI_PHY_3_NVCSI_CIL_B_CLKEN_OVERRIDE_CTRL_0

Offset: 0x10760
Byte Offset: 0x41d80
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**CLKEN_OVERRIDE1_B**: CLKENABLE OVERRIDE FOR LANE B1. Active High **1** = ENABLE **0** = DISABLE

Bit	Reset	Description
0	0x0	**CLKEN_OVERRIDE0_B**: CLKENABLE OVERRIDE FOR LANE B0. Active High **1** = ENABLE **0** = DISABLE

NVCSI_PHY_3_NVCSI_CIL_B_CTLE_CTRL_0

Offset: 0x10761
Byte Offset: 0x41d84
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000888 (0bxxxx,xxxx,0000,0000,0000,1000,1000,1000)

Bit	Reset	Description
23:20	0x0	**AFE_DCGAIN_IO1_B**: RX AFE DC gain control
19:16	0x0	**AFE_DCGAIN_IO0_B**: RX AFE DC gain control
15:12	0x0	**AFE_DCGAIN_CLK_B**: RX AFE DC gain control
11:8	0x8	**AFE_HFGAIN_IO1_B**: RX AFE_CTLE high frequency gain control, 16 curves, ac_gain=code*1db
7:4	0x8	**AFE_HFGAIN_IO0_B**: RX AFE_CTLE high frequency gain control, 16 curves, ac_gain=code*1db
3:0	0x8	**AFE_HFGAIN_CLK_B**: RX AFE_CTLE high frequency gain control, 16 curves, ac_gain=code*1db

NVCSI_PHY_3_NVCSI_CIL_B_CTLE_CTRL1_0

Offset: 0x10762
Byte Offset: 0x41d88
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000554 (0bxxxx,xxxx,0000,0000,0000,0101,0101,0100)

Bit	Reset	Description
23:20	0x0	**AFE_CTRL_IO1_B**
19:16	0x0	**AFE_CTRL_IO0_B**
15:12	0x0	**AFE_CTRL_CLK_B**
11:10	0x1	**AFE_CURRENT_IO1_B**
9:8	0x1	**AFE_CURRENT_IO0_B**
7:6	0x1	**AFE_CURRENT_CLK_B**
5:4	0x1	**AFE_FREQBAND_IO1_B**: RX AFE bandwidth selection. 2'b00: <1.5GS/s, 2'b01: 1.5~3GS/s, 2'b10: 3~5GS/s, 2'b11: reserved.
3:2	0x1	**AFE_FREQBAND_IO0_B**: RX AFE bandwidth selection. 2'b00: <1.5GS/s, 2'b01: 1.5~3GS/s, 2'b10: 3~5GS/s, 2'b11: reserved.
1:0	0x0	**AFE_FREQBAND_CLK_B**: RX AFE bandwidth selection. 2'b00: <1.5GS/s, 2'b01: 1.5~3GS/s, 2'b10: 3~5GS/s, 2'b11: reserved.

NVCSI_PHY_3_NVCSI_CIL_B_PAD_CONFIG_0

Offset: 0x10763
Byte Offset: 0x41d8c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00070000 (0b0000,0000,0000,x111,xxxx,0000,0000,0000)

Bit	Reset	Description
31	0x0	**LP_RX_SELECT_IO1_B**: LP mode receiver select for partition B lane 1. DPHY mode: controls IO1P_B and IO1N_B CPHY mode: controls IO1P_B, IO1N_B and IOCLKN_B.
30	0x0	**LP_RX_SELECT_IO0_B**: LP mode receiver select for partition B lane 0. DPHY mode: controls IO0P_B and IO0N_B CPHY mode: controls IO0P_B, IO0N_B and IOCLKP_B.
29	0x0	**LP_RX_SELECT_CLK_B**: LP mode receiver select. 0: Schmitt receiver. 1: self-biased receiver. DPHY mode: controls IOCLKP_B and IOCLKN_B. CPHY mode: dummy.

Bit	Reset	Description
28	0x0	**E_LPRX_HYS_N_IO1_B**: Enable LP-RX hysteresis. 0: enable hysteresis 1: disable hysteresis
27	0x0	**E_LPRX_HYS_N_IO0_B**: Enable LP-RX hysteresis. 0: enable hysteresis 1: disable hysteresis
26	0x0	**E_LPRX_HYS_N_CLK_B**: Enable LP-RX hysteresis. 0: enable hysteresis 1: disable hysteresis
25	0x0	**E_HS_TERM_N_IO1_B**: Enable the HS termination of Partition B Lane 1.
24	0x0	**E_HS_TERM_N_IO0_B**: Enable the HS termination of Partition B Lane 0.
23	0x0	**E_HS_TERM_N_CLK_B**: Enable the HS termination of Clock partition B.
22	0x0	**E_INPUT_LP_IO1_B**: Enable LP receiver of Partition B Lane 1 Applicable in both CPHY and DPHY case.
21	0x0	**E_INPUT_LP_IO0_B**: Enable LP receiver of Partition B Lane 0 Applicable in both CPHY and DPHY case.
20	0x0	**E_INPUT_LP_CLK_B**: enable LP receiver of Clock partition B Applicable in DPHY case. N/A for CPHY
18	0x1	**PD_CLK_B**: Power down for CLk of Partition B. Applicable in DPHY case. N/A for CPHY
17	0x1	**PD_IO1_B**: Power down for Trio 1 and Lane 1 of Partition B. Applicable in both CPHY and DPHY case.
16	0x1	**PD_IO0_B**: Power down for Trio 0 and Lane 0 of Partition B. Applicable in both CPHY and DPHY case.
11:8	0x0	**SPARE_CLK_B**: Spare control bits for pad control. Functionality of other bits currently not defined. Undefined register bits are reserved for future use. Pull down may be enabled when Clock Lane is unused to minimized interference.
7:4	0x0	**SPARE_IO1_B**: Spare control bits for pad control. Functionality of other bits currently not defined. Undefined register bits are reserved for future use. Pull down may be enabled when Data Lane-1 is unused to minimized interference.

Bit	Reset	Description
3:0	0x0	**SPARE_IO0_B**: Spare control bits for pad control. Functionality of other bits currently not defined. Undefined register bits are reserved for future use. Pull down may be enabled when Data Lane-0 is unused to minimized interference.

NVCSI_PHY_3_NVCSI_CIL_B_PAD_CONFIG_1_0

Offset: 0x10764
Byte Offset: 0x41d90
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000002 (0b0000,0000,0000,0000,0000,0000,xx00,0010)

Bit	Reset	Description
31:24	0x0	**OUTADJ_IO1_B**: Trio 1 output delay trimmer, each tap delays 9ps
23:16	0x0	**OUTADJ_IO0_B**: Trio 0 output delay trimmer, each tap delays 9ps
15:8	0x0	**OUTADJ_CLK_B**: Clock bit output delay trimmer, each tap delays 9ps
5	0x0	**HS_BSO_B**: 1= power-on 0.4v-TX-regulator during LP-mode; 0= power-off 0.4v-TX-regulator during LP-mode;
4	0x0	**CPHY_MID_STRENGTH_TRIO1_B**: 0: when bump is driven to mid-voltage-level, both 100-ohm-pull-up and 100-ohm-pull-down is turned on to drive mid-voltage-level; 1: when bump is driven to mid-voltage-level, both pull-up and pull-down are turned off to become tri-stated;
3	0x0	**CPHY_MID_STRENGTH_TRIO0_B**: 0: when bump is driven to mid-voltage-level, both 100-ohm-pull-up and 100-ohm-pull-down is turned on to drive mid-voltage-level; 1: when bump is driven to mid-voltage-level, both pull-up and pull-down are turned off to become tri-stated;
2	DISABLE	**REV_CLK_B**: Reverse clock polarity **0** = DISABLE **1** = ENABLE
1	ENABLE	**E_PREDRV_FLOP_RESET_B**: HS predriver flop reset **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
0	DISABLE	**FCZERO_B**: Force clk bit to drive differential 0 **0** = DISABLE **1** = ENABLE

NVCSI_PHY_3_NVCSI_CIL_B_PAD_CONFIG_2_0

Offset: 0x10765
Byte Offset: 0x41d94
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,0000,0000,0000,xxxx,0000,0000,0000)

Bit	Reset	Description
27:24	0x0	**PEMPD_IO1_B**: Enable trio 1 HS driver pull down pre-emphasis.
23:20	0x0	**PEMPD_IO0_B**: Enable trio 0 HS driver pull down pre-emphasis.
19:16	0x0	**PEMPD_CLK_B**: Enable clock bit HS driver pull down pre-emphasis.
11:8	0x0	**PEMPU_IO1_B**: Enable trio 1 HS driver pull up pre-emphasis.
7:4	0x0	**PEMPU_IO0_B**: Enable trio 0 HS driver pull up pre-emphasis.
3:0	0x0	**PEMPU_CLK_B**: Enable clock bit HS driver pull up pre-emphasis.

NVCSI_PHY_3_NVCSI_CIL_B_PAD_CONFIG_3_0

Offset: 0x10766
Byte Offset: 0x41d98
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x06660666 (0bxxxx,0110,0110,0110,xxxx,0110,0110,0110)

Bit	Reset	Description
27:24	0x6	**LPDNADJ_IO1_B**: Driver pull down impedance control
23:20	0x6	**LPDNADJ_IO0_B**: Driver pull down impedance control
19:16	0x6	**LPDNADJ_CLK_B**: Driver pull down impedance control
11:8	0x6	**LPUPADJ_IO1_B**: Driver pull up impedance control
7:4	0x6	**LPUPADJ_IO0_B**: Driver pull up impedance control
3:0	0x6	**LPUPADJ_CLK_B**: Driver pull up impedance control

NVCSI_PHY_3_NVCSI_CIL_B_PAD_CONFIG_4_0

Offset: 0x10767
Byte Offset: 0x41d9c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,0000,0000,0000,xxxx,0000,0000,0000)

Bit	Reset	Description
27:24	0x0	**SLEWDNADJ_IO1_B**: Pull down slew rate adjust
23:20	0x0	**SLEWDNADJ_IO0_B**: Pull down slew rate adjust
19:16	0x0	**SLEWDNADJ_CLK_B**: Pull down slew rate adjust
11:8	0x0	**SLEWUPADJ_IO1_B**: Pull up slew rate adjust
7:4	0x0	**SLEWUPADJ_IO0_B**: Pull up slew rate adjust
3:0	0x0	**SLEWUPADJ_CLK_B**: Pull up slew rate adjust

NVCSI_PHY_3_NVCSI_CIL_B_PAD_CONFIG_5_0

Offset: 0x10768

Byte Offset: 0x41da0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000110 (0bxxxx,xxxx,xxxx,xxxx,xxxx,x001,0001,0000)

Bit	Reset	Description
10:7	0x2	**CDDNADJ_IO1_B**: Level adjust on low limit of detection
6:3	0x2	**CDDNADJ_IO0_B**: Level adjust on low limit of detection
2	DISABLE	**E_CD_IO1_B**: Data bits contention detector enable **0** = DISABLE **1** = ENABLE
1	DISABLE	**E_CD_IO0_B**: Data bits contention detector enable **0** = DISABLE **1** = ENABLE
0	DISABLE	**E_CD_CLK_B**: Clock bit contention detector enable **0** = DISABLE **1** = ENABLE

NVCSI_PHY_3_NVCSI_CIL_B_PAD_CONFIG_6_0

Offset: 0x10769
Byte Offset: 0x41da4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**E_DEEP_LPBK_B**: Enable deep loopback
0	0x0	**E_SHALLOW_LPBK_B**: Enalbe shallow loopback

NVCSI_PHY_3_NVCSI_CIL_B_PAD_CD_STATUS_0

Offset: 0x1076a
Byte Offset: 0x41da8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**CDN_IO1_B**: trio1 n output, 1=contention detected.
4	0x0	**CDN_IO0_B**: trio0 n output, 1=contention detected.
3	0x0	**CDN_CLK_B**: Clock bit n output, 1=contention detected.
2	0x0	**CDP_IO1_B**: trio1 p output, 1=contention detected.
1	0x0	**CDP_IO0_B**: trio0 p output, 1=contention detected.
0	0x0	**CDP_CLK_B**: Clock bit p output, 1=contention detected.

NVCSI_PHY_3_NVCSI_CIL_B_DPHY_INADJ_CTRL_0

Offset: 0x1076b
Byte Offset: 0x41dac
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,x000,0000,x000,0000,x000,0000)

Bit	Reset	Description
22	0x0	**SW_SET_DPHY_INADJ_CLK_B**: Software set for clock input delay trimmer
21:16	0x0	**DPHY_INADJ_CLK_B**: Programmable value for CLK input delay trimmer,

Bit	Reset	Description
14	0x0	**SW_SET_DPHY_INADJ_IO1_B**: Software override for bit 1 input delay trimmer 1 - SW programmed value is applied to the pads. 2 - Trimmer value from HW calibration state machine is applied to the pads
13:8	0x0	**DPHY_INADJ_IO1_B**: Programmable value for bit 1 input delay trimmer, each tap delay of 9 ps
6	0x0	**SW_SET_DPHY_INADJ_IO0_B**: Software override for bit 0 input delay trimmer 1 - SW programmed value is applied to the pads. 2 - Trimmer value from HW calibration state machine is applied to the pads
5:0	0x0	**DPHY_INADJ_IO0_B**: Programmable value for bit 0 input delay trimmer,

NVCSI_PHY_3_NVCSI_CIL_B_CLK_DESKEW_CTRL_0

Offset: 0x1076c
Byte Offset: 0x41db0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000bf00 (0bxxxx,xxxx,xxxx,xxxx,1x11,1111,xx00,0000)

Bit	Reset	Description
15	ENABLE	**CLK_INADJ_SWEEP_CTRL_B**: Enable the clock trimmer sweep for D-PHY de-skew. **0** = DISABLE **1** = ENABLE
13:8	0x3f	**CLK_INADJ_LIMIT_HIGH_B**: The upper limit of the sweep range for trimmer sweep.
5:0	0x0	**CLK_INADJ_LIMIT_LOW_B**: The lower limit of the sweep range for trimmer sweep.

NVCSI_PHY_3_NVCSI_CIL_B_DATA_DESKEW_CTRL_0

Offset: 0x1076d
Byte Offset: 0x41db4
Read/Write: R/W
Parity Protection: Y
Shadow: N

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x9f80bf00 (0b1xx1,1111,1x00,0000,1x11,1111,xx00,0000)

Bit	Reset	Description
31	0x1	**DATA_INADJ_SWEEP_CTRL1_B**: Enable the data lane B1 trimmer sweep for D-PHY de-skew.
28:23	0x3f	**DATA_INADJ_LIMIT_HIGH1_B**: The upper limit of the sweep range for trimmer sweep.
21:16	0x0	**DATA_INADJ_LIMIT_LOW1_B**: The lower limit of the sweep range for trimmer sweep.
15	0x1	**DATA_INADJ_SWEEP_CTRL0_B**: Enable the data lane B0 trimmer sweep for D-PHY de-skew.
13:8	0x3f	**DATA_INADJ_LIMIT_HIGH0_B**: The upper limit of the sweep range for trimmer sweep.
5:0	0x0	**DATA_INADJ_LIMIT_LOW0_B**: The lower limit of the sweep range for trimmer sweep.

NVCSI_PHY_3_NVCSI_CIL_B_DPHY_DESKEW_STATUS_0

Offset: 0x1076e
Byte Offset: 0x41db8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000X0XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
15	X	**DPHY_CALIB_ERR_IO1_B**: calib error status
14	X	**DPHY_CALIB_DONE_IO1_B**: calib done status
7	X	**DPHY_CALIB_ERR_IO0_B**: calib error status
6	X	**DPHY_CALIB_DONE_IO0_B**: calib done status
1	X	**DPHY_CALIB_ERR_CTRL_B**: calib error status
0	X	**DPHY_CALIB_DONE_CTRL_B**: calib done status

NVCSI_PHY_3_NVCSI_CIL_B_DPHY_DESKEW_DATA_CALIB_STATUS_LOW_0_0

Offset: 0x1076f
Byte Offset: 0x41dbc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_B**

NVCSI_PHY_3_NVCSI_CIL_B_DPHY_DESKEW_DATA_CALIB_STATUS_HIGH_0_0

Offset: 0x10770
Byte Offset: 0x41dc0
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_B**

NVCSI_PHY_3_NVCSI_CIL_B_DPHY_DESKEW_DATA_CALIB_STATUS_LOW_1_0

Offset: 0x10771
Byte Offset: 0x41dc4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_B**

NVCSI_PHY_3_NVCSI_CIL_B_DPHY_DESKEW_DATA_CALIB_STATUS_HIGH_1_0

Offset: 0x10772
Byte Offset: 0x41dc8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_B**

NVCSI_PHY_3_NVCSI_CIL_B_DPHY_DESKEW_CLK_CALIB_STATUS_LOW_0_0

Offset: 0x10773
Byte Offset: 0x41dcc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_B**

NVCSI_PHY_3_NVCSI_CIL_B_DPHY_DESKEW_CLK_CALIB_STATUS_HIGH_0_0

Offset: 0x10774
Byte Offset: 0x41dd0
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO0_B**

NVCSI_PHY_3_NVCSI_CIL_B_DPHY_DESKEW_CLK_CALIB_STATUS_LOW_1_0

Offset: 0x10775
Byte Offset: 0x41dd4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_B**

NVCSI_PHY_3_NVCSI_CIL_B_DPHY_DESKEW_CLK_CALIB_STATUS_HIGH_1_0

Offset: 0x10776
Byte Offset: 0x41dd8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DPHY_CALIB_STATUS_IO1_B**

NVCSI_PHY_3_NVCSI_CIL_B_DPHY_DESKEW_RESULT_STATUS_0

Offset: 0x10777
Byte Offset: 0x41ddc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	DONE	**DESKEW_RESULT_STATUS1_B**: For lane B1. The result status of the deskew calibration, HW check this field before start the trimmer sweep, if it's 1'b1, the trimmer sweep will not perform. HW set this bit to 1 when the calibration is done. SW should clear this field after it read out the 2 64 bits calibration status. **0** = DONE **1** = PENDING
0	DONE	**DESKEW_RESULT_STATUS0_B**: For lane B0. The result status of the deskew calibration, HW check this field before start the trimmer sweep, if it's 1'b1, the trimmer sweep will not perform. HW set this bit to 1 when the calibration is done. SW should clear this field after it read out the 2 64 bits calibration status. **0** = DONE **1** = PENDING

NVCSI_PHY_3_NVCSI_CIL_B_POLARITY_SWIZZLE_CTRL_0

Offset: 0x10778
Byte Offset: 0x41de0
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xx00,0000,xxxx,x000)

Bit	Reset	Description
13:11	0x0	**POLARITY_SWIZZLE_CPHY1_B**: Polarity Swizle control for Lane B1 in CPHY mode. valid only in CPHY mode 000 - A B C --> A B C 001 - A B C --> A C B 010 - A B C --> B C A 011 - A B C --> B A C 100 - A B C --> C A B 101 - A B C --> C B A
10:8	0x0	**POLARITY_SWIZZLE_CPHY0_B**: Polarity Swizle control for Lane B0 in CPHY mode. valid only in CPHY mode 000 - A B C --> A B C 001 - A B C --> A C B 010 - A B C --> B C A 011 - A B C --> B A C 100 - A B C --> C A B 101 - A B C --> C B A
2	0x0	**POLARITY_SWIZZLE_CLK_B**: Polarity Swizzle control for clock lane B. Valid only in DPHY mode, need to work with ALLOW_FIRST_BIT_ON_CLK_NEGEDGE if enabled.

Bit	Reset	Description
1	0x0	**POLARITY_SWIZZLE_DPHY1_B**: Polarity Swizzle control for data lane B1. register bit valid only in DPHY mode.
0	0x0	**POLARITY_SWIZZLE_DPHY0_B**: Polarity Swizzle control for data lane B0. register bit valid only in DPHY mode.

NVCSI_PHY_3_NVCSI_CIL_B_DESKEW_CONTROL_0

Offset: 0x10779
Byte Offset: 0x41de4
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00007411 (0bxxxx,xxxx,xxxx,xxxx,0111,0100,0001,0001)

Bit	Reset	Description
15:13	0x3	**DESKEW_DATA_DELAY**: How many cycles delay introduced by re-timing flops
12	0x1	**DESKEW_LANE_SKEW_CHECK_EN**: Check the skew between the lanes. this is used to handle the error case that error happen in the CSI link caused some data lanes detect the HS sync word (0xb8) while other data lanes detect the deskew sync word (0xff). If this bit is not set with the error case, the lane FIFO will overflow and reset is required. With this bit set, the FIFO will be auto flused and the FIFO overflow will not happen.
11:8	0x4	**DESKEW_LANE_SKEW_TOLERANCE**: The allowed skew between the data lane, only used in sync word search
7:4	0x1	**DESKEW_COMPARE**: Regsiter select to control the number of comparisons to be done for each trimmer setting during deskew calibration.
3:0	0x1	**DESKEW_SETTLE**: Regsiter select to control the number of byte clocks to wait before INADJ value settles. This is used during deskew calibration

NVCSI_PHY_3_NVCSI_CIL_B_CONTROL_0

Offset: 0x1077a
Byte Offset: 0x41de8
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**BYPASS_LP_FILTER**: Bypass the glitch filter.
30:24	0x0	**CLK_SETTLE1_B**: Used only in CPHY mode, Settle time for clk start when moving B1 data lane from LP to HS (LP11->LP01->LP00).
23:17	0x0	**CLK_SETTLE0_B**: Used for both CPHY/DPY mode. For DPHY: Settle time for clk lane when moving from LP to HS (LP11->LP01->LP00), this setting determines how many csicil clock cycles to wait after LP00. If the settle set to 0 or 8, the real setting is 1, otherwise, the real setting is (4+settle) For CPHY: Settle time for clk start when moving B0 data lane from LP to HS (LP11->LP01->LP00).
16:9	0x0	**THS_SETTLE1_B**: settle time for B1 data lane when moving from LP to HS (LP11->LP01->LP00), this setting determines how many LP clock cycles to wait, after LP00, before starting to look at the data.
8:1	0x0	**THS_SETTLE0_B**: settle time for B0 data lane when moving from LP to HS (LP11->LP01->LP00), this setting determines how many LP clock cycles to wait, after LP00, before starting to look at the data.
0	0x0	**BYPASS_LP_SEQ**: For DPHY: The clock lane LP signals should sequence through LP11->LP01->LP00 state, to indicate to CLOCK CIL about the mode switching to HS Rx mode. In case Camera is enabled earlier than CIL , it is highly likely that camera sends this control sequence sooner than cil can detect it. Enabling this bit allows the CLOCK CIL to overlook the LP control sequence and step in HS Rx mode directly looking at LP00 only. For CPHY: it's only used for debug purpose as it's embedded clock for CPHY, clock will always presents when the data lane toggle.

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_ERR_STATUS_0

Offset: 0x1077b
Byte Offset: 0x41dec
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
29:24	X	**LANE_DEMAPPER_ERR_MUX1_B**: Debug register, error detected in CPHY de-mapper mux.
23	X	**LANE_DEMAPPER_ERR_RXCTRL1_B**: Debug register, error detected in CPHY de-mapper rx ctrl.
22:16	X	**LANE_DECODER_ERR1_B**: Debug register, error detected in CPHY de-coder.
13:8	X	**LANE_DEMAPPER_ERR_MUX0_B**: Debug register, error detected in CPHY de-mapper mux.
7	X	**LANE_DEMAPPER_ERR_RXCTRL0_B**: Debug register, error detected in CPHY de-mapper rx ctrl.
6:0	X	**LANE_DECODER_ERR0_B**: Debug register, error detected in CPHY de-coder.

NVCSI_PHY_3_NVCSI_CIL_B_ESCAPE_MODE_COMMAND_0_0

Offset: 0x1077c
Byte Offset: 0x41df0
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_COMMAND0_B**: The received command from escape mode.

NVCSI_PHY_3_NVCSI_CIL_B_ESCAPE_MODE_DATA_0_0

Offset: 0x1077d
Byte Offset: 0x41df4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_DATA0_B**: The received data from excape mode.

NVCSI_PHY_3_NVCSI_CIL_B_ESCAPE_MODE_COMMAND_1_0

Offset: 0x1077e
Byte Offset: 0x41df8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_COMMAND1_B**: The received command from escape mode.

NVCSI_PHY_3_NVCSI_CIL_B_ESCAPE_MODE_DATA_1_0

Offset: 0x1077f
Byte Offset: 0x41dfc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**ESCAPE_MODE_DATA1_B**: The received data from excape mode.

NVCSI_PHY_3_NVCSI_CIL_B_DPHY_SYNC_PATTERN_0

Offset: 0x10780
Byte Offset: 0x41e00
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01)

Bit	Reset	Description
1	0x0	**ALLOW_FIRST_BIT_ON_CLK_NEGEDGE_B**: Allow shift 1/3/5/7 bit to search the sync word. Normal case only shift 0/2/4/6 bit to search sync word.
0	0x1	**DISABLE_SB_ERR_IN_SYNC_B**: Allow one single bit error in sync word 6MSB.

NVCSI_PHY_3_NVCSI_CIL_B_DPHY_DESKEW_SYNC_PATTERN_0

Offset: 0x10781
Byte Offset: 0x41e04
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**INVERT_DESKEW_PATTERN**: Invert the de-skew pattern.

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_SYNC_SEARCH_0

Offset: 0x10782
Byte Offset: 0x41e08
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01080108 (0bxxxx,xxx1,0000,1000,xxxx,xxx1,0000,1000)

Bit	Reset	Description
24	0x1	**COUNT_EN_TRIO1_B**: Check if the symbol is pre-amble before sync word.
23:16	0x8	**PERIOD_TRIO1_B**: For CPHY, how many cycles allowed for non pre-amble symbol before sync word.
8	0x1	**COUNT_EN_TRIO0_B**: Check if the symbol is pre-amble before sync word.

Bit	Reset	Description
7:0	0x8	**PERIOD_TRIO0_B**: For CPHY, how many cycles allowed for non pre-amble symbol before sync word.

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_START_0

Offset: 0x10783
Byte Offset: 0x41e0c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**EDGE_DELAY_CAL_TRIGGER_TRIO1_B**: Trigger the CPHY clock recover calibration for lane B1 by write 1'b1, when the calibration complete, HW will clear this bit to 1'b0. Write 1'b0 to this be take no effect.
0	0x0	**EDGE_DELAY_CAL_TRIGGER_TRIO0_B**: Trigger the CPHY clock recover calibration for lane B0 by write 1'b1, when the calibration complete, HW will clear this bit to 1'b0. Write 1'b0 to this be take no effect.

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_CTRL_0

Offset: 0x10784
Byte Offset: 0x41e10
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x12011201 (0b0001,0010,0000,0001,0001,0010,0000,0001)
PROD: 0x08000800 (0bxxx0,10xx,xxxx,xxxx,xxx0,10xx,xxxx,xxxx)

Bit	Reset	PROD	Description
31	0x0	_NONE_	**TRIM_CAL_E_NONSTOP_TRIO1_B**: Reserved. Trimmer calibration background monitoring control. 1: enable. The accumulator keeps updating.

Bit	Reset	PROD	Description
30:29	0x0	_NONE_	**TRIM_CAL_GAIN_SEL_TRIO1_B**: Trimmer calibration loop gain selection [0]: gain control 0: coarse tune (counter1) gain = 1, fine tune (counter2) gain = 1/4. For example: coarse tune: 1 UP -> trim_cal_code + 1; fine tune: 4 UPs -> trim_cal_code + 1. 1: coarse tune (counter1) gain = 1/2, fine tune (counter2) gain = 1/8 [1]: reserved
28:26	0x4	0x2	**TRIM_CAL_CNT_SEL_TRIO1_B**: Trimmer calibration counter selection [0]: counter1 (coarse tune) clock cycle. 0: 64T (128UI) 1: 128T (256UI) [2:1]: counter2 (fine tune) clock cycle. 00: 32T (64UI) 01: 64T (128UI) 10: 96T (192UI) 11: 128T (256UI) For high data rate (>3.5GS/s): 64T(cnt1)+96T(cnt2)=160T(320ui) -> cnt_sel=3'b100 For low data rate (<=3.5GS/s): 128T(cnt1)+32T(cnt2)=160T(320ui) -> cnt_se=3'b001
25	0x1	_NONE_	**TRIM_CAL_CPHY_ED_LOOPMODE_TRIO1_B**: Trimmer calibration and cphy edge detector loop mode control 0: exclude trimmer delay in 2nd round of reset loop. 1: include trimmer delay in 2nd round of reset loop. For low data rate.
24:16	0x1	_NONE_	**CALIB_LENGTH_TRIO1_B**: The length of the peramble which use to do caliration. Count by nvcsi core clock. It's used to make sure the calibration is doing inside the preamble phase. If this length set to a small value, the calibration may can't complete within one packet and need several packets to complete.
15	0x0	_NONE_	**TRIM_CAL_E_NONSTOP_TRIO0_B**: Reserved. Trimmer calibration background monitoring control. 1: enable. The accumulator keeps updating.
14:13	0x0	_NONE_	**TRIM_CAL_GAIN_SEL_TRIO0_B**: Trimmer calibration loop gain selection [0]: gain control 0: coarse tune (counter1) gain = 1, fine tune (counter2) gain = 1/4. For example: coarse tune: 1 UP -> trim_cal_code + 1; fine tune: 4 UPs -> trim_cal_code + 1. 1: coarse tune (counter1) gain = 1/2, fine tune (counter2) gain = 1/8 [1]: reserved

Bit	Reset	PROD	Description
12:10	0x4	0x2	**TRIM_CAL_CNT_SEL_TRIO0_B**: Trimmer calibration counter selection [0]: counter1 (coarse tune) clock cycle. 0: 64T (128UI) 1: 128T (256UI) [2:1]: counter2 (fine tune) clock cycle. 00: 32T (64UI) 01: 64T (128UI) 10: 96T (192UI) 11: 128T (256UI) For high data rate (>3.5GS/s): 64T(cnt1)+96T(cnt2)=160T(320ui) -> cnt_sel=3'b100 For low data rate (<=3.5GS/s): 128T(cnt1)+32T(cnt2)=160T(320ui) -> cnt_se=3'b001
9	0x1	_NONE_	**TRIM_CAL_CPHY_ED_LOOPMODE_TRIO0_B**: Trimmer calibration and cphy edge detector loop mode control 0: exclude trimmer delay in 2nd round of reset loop. 1: include trimmer delay in 2nd round of reset loop. For low data rate.
8:0	0x1	_NONE_	**CALIB_LENGTH_TRIO0_B**: The length of the peramble which use to do caliration. Count by nvcsi core clock. It's used to make sure the calibration is doing inside the preamble phase. If this length set to a small value, the calibration may can't complete within one packet and need several packets to complete.

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_CTRL1_0

Offset: 0x10785
Byte Offset: 0x41e14
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0c0e0c0e (0bx000,1100,0000,1110,x000,1100,0000,1110)

Bit	Reset	Description
30	0x0	**PERIODIC_CALIB_TRIO1_B**: Enable the periodic calibration.
29	0x0	**CALIB_APPLY_IN_POST_TRIO1_B**: 1: Apply the calibration result when detect post symbol. 0: Apply the calibration result immediately when the calibration done.

Bit	Reset	Description
28:27	0x1	**CALIB_APPLY_SELECT_TRIO1_B**: Which calibration result is applied to pad for dual calibration. 2'b11: Reserved 2'b10: Apply the middle value of two calibration to pad if the difference within threshold. 2'b01: Apply the 2nd calibration result to pad if it's difference with previous latch result within threshold. 2'b00: Apply the 1st calibration result to pad if it's difference with previous latch result within threshold.
26	0x1	**DUAL_CALIB_TRIO1_B**: Two times calibratiton. 1: Two time calibration with one trigger. 0: Single time calibration with one trigger
25:18	0x3	**TEMP_EDGE_DELAY_VALUE_TRIO1_B**: The edge delay value while doing the calibration.
17	0x1	**TEMP_EDGE_DELAY_SEL_TRIO1_B**: When the selection of edge delay value when do edge delay calibration. 1: Use the temp value from register setting. 0: Use the currently using setting. For the first calibration after reset, it's still using the temp value from register setting.
16	0x0	**BYPASS_CALIB_PACKET_TRIO1_B**: 1: Drop the packet data when do the calibration. 0: Not drop the packet data when do the calibration.
14	0x0	**PERIODIC_CALIB_TRIO0_B**: Enable the periodic calibration.
13	0x0	**CALIB_APPLY_IN_POST_TRIO0_B**: 1: Apply the calibration result when detect post symbol. 0: Apply the calibration result immediately when the calibration done.
12:11	0x1	**CALIB_APPLY_SELECT_TRIO0_B**: Which calibration result is applied to pad for dual calibration. 2'b11: Reserved 2'b10: Apply the middle value of two calibration to pad if the difference within threshold. 2'b01: Apply the 2nd calibration result to pad if it's difference with previous latch result within threshold. 2'b00: Apply the 1st calibration result to pad if it's difference with previous latch result within threshold.
10	0x1	**DUAL_CALIB_TRIO0_B**: Two times calibratiton. 1: Two time calibration with one trigger. 0: Single time calibration with one trigger
9:2	0x3	**TEMP_EDGE_DELAY_VALUE_TRIO0_B**: The edge delay value apply to pad while doing the calibration.
1	0x1	**TEMP_EDGE_DELAY_SEL_TRIO0_B**: When the selection of edge delay value when do edge delay calibration. 1: Use the temp value from register setting. 0: Use the currently using setting. For the first calibration after reset, it's still using the temp value from register setting.
0	0x0	**BYPASS_CALIB_PACKET_TRIO0_B**: 1: Drop the packet data when do the calibration. 0: Not drop the packet data when do the calibration.

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_CTRL2_0

Offset: 0x10786
Byte Offset: 0x41e18
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01000100 (0bxxxx,xxx1,0000,0000,xxxx,xxx1,0000,0000)

Bit	Reset	Description
24	ENABLE	**TRIM_CAL_PD_HOLD_TRIO1_B**: Hold in PD **0** = DISABLE **1** = ENABLE
23	DISABLE	**TRIM_CAL_E_FIXEDDLY_TRIO1_B**: Enable fixed delay in trim calibration unit. **0** = DISABLE **1** = ENABLE
22:16	0x0	**TRIM_CAL_FIXEDDLY_TRIO1_B**: Adjust fixed delay. Delay = code*5ps(TT)/7ps(SS)
8	ENABLE	**TRIM_CAL_PD_HOLD_TRIO0_B**: Hold in PD **0** = DISABLE **1** = ENABLE
7	DISABLE	**TRIM_CAL_E_FIXEDDLY_TRIO0_B**: Enable fixed delay in trim calibration unit. **0** = DISABLE **1** = ENABLE
6:0	0x0	**TRIM_CAL_FIXEDDLY_TRIO0_B**: Adjust fixed delay. Delay = code*5ps(TT)/7ps(SS)

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_CTRL3_0

Offset: 0x10787
Byte Offset: 0x41e1c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffffffff	**CALIB_PERIOD_TRIO0_B**: The period of the periodic edge delay calibration, in nvcsi clock cycle number.

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_CTRL4_0

Offset: 0x10788
Byte Offset: 0x41e20
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffffffff	**CALIB_PERIOD_TRIO1_B**: The period of the periodic edge delay calibration, in nvcsi clock cycle number.

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_CTRL5_0

Offset: 0x10789
Byte Offset: 0x41e24
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0000ffff (0bxxxx,xxxx,xxxx,xxxx,1111,1111,1111,1111)

Bit	Reset	Description
15:8	0xff	**ERR_THRESHOLD_TRIO1_B**: The threshold of drop the calibration result between current result and previous result. For dual calibration, check between the two calibration result in the same trigger. For signal calibration, check between the current calibration result and previous calibration result.
7:0	0xff	**ERR_THRESHOLD_TRIO0_B**: The threshold of drop the calibration result between current result and previous result. For dual calibration, check between the two calibration result in the same trigger. For signal calibration, check between the current calibration result and previous calibration result.

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_STATUS_0

Offset: 0x1078a
Byte Offset: 0x41e28
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,xxxx)

Bit	Reset	Description
9	0x0	**EDGE_DELAY_MISMATCH_TRIO1_B**: When the CALIB_APPLY_SELECT_TRIO is 0/1 or dual time calibration is not enabled, the calibration result difference from the previous latched calibration result is larger than threshold, this time calibration result is not apply to pad. When the CALIB_APPLY_SELECT_TRIO is 2, the calibration result difference of the two times calibration result is larger than threshold, this time calibration result is not apply to pad.
8	0x0	**EDGE_DELAY_MISMATCH_TRIO0_B**: When the CALIB_APPLY_SELECT_TRIO is 0/1 or dual time calibration is not enabled, the calibration result difference from the previous latched calibration result is larger than threshold, this time calibration result is not apply to pad. When the CALIB_APPLY_SELECT_TRIO is 2, the calibration result difference of the two times calibration result is larger than threshold, this time calibration result is not apply to pad.
7	0x0	**CALIB_DONE_1_TRIO1_B**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.
6	0x0	**CALIB_DONE_1_TRIO0_B**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.
5	0x0	**CALIB_DONE_0_TRIO1_B**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.
4	0x0	**CALIB_DONE_0_TRIO0_B**: Maximum trimmer value reached without finding optimal trimmer setting for the given data rate.

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_EDGE_DELAY_CTRL_0

Offset: 0x1078b
Byte Offset: 0x41e2c
Read/Write: R/W
Parity Protection: See table below
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x40000000 (0b01xx,x000,0000,0000,xxxx,x000,0000,0000)

Bit	Parity Protection	Reset	Description
31:30	Y	0x1	**CPHY_CLKGEN_TRIGWIDTH_B**: Adjust AB/CB/CA pulse width in self-clock generator
26	N	0x0	**UPDATE_OFFSET_TRIO1_B**: Trigger to apply the a offset value. Used to update offset directly without trigger a new calibration.
25:17	Y	0x0	**DELAY_OFFSET_TRIO1_B**: Offset value of trio B1. Represents signed 2's complement number for calibration offset. Represents unsigned number if the field used as override value.
16	Y	0x0	**OVERRIDE_CAL_VAL_TRIO1_B**: Override for trio B1 0: Apply the DELAY_OFFSET as an offset to the calibration result. 1: Apply the DELAY_OFFSET value to the pad.
10	N	0x0	**UPDATE_OFFSET_TRIO0_B**: Trigger to apply the a offset value. Used to update offset directly without trigger a new calibration.
9:1	Y	0x0	**DELAY_OFFSET_TRIO0_B**: Offset value of trio B0. Represents signed 2's complement number for calibration offset. Represents unsigned number if the field used as override value.
0	Y	0x0	**OVERRIDE_CAL_VAL_TRIO0_B**: Override for trio B0 0: Apply the DELAY_OFFSET as an offset to the calibration result. 1: Apply the DELAY_OFFSET value to the pad.

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_EDGE_DELAY_VALUE_0

Offset: 0x1078c
Byte Offset: 0x41e30
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000303 (0bxxxx,xxxx,xxxx,xxxx,0000,0011,0000,0011)

Bit	Reset	Description
15:8	0x3	**EDGE_DELAY_VALUE_TRIO1_B**: The edge delay value apply to pad finally

Bit	Reset	Description
7:0	0x3	**EDGE_DELAY_VALUE_TRIO0_B**: The edge delay value apply to pad finally

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_EDGE_DELAY_CALIB_RESULT_0

Offset: 0x1078d
Byte Offset: 0x41e34
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**EDGE_DELAY_CALIB_RESULT_1_TRIO1_B**: The edge delay calibrate result, with offset.
23:16	0x0	**EDGE_DELAY_CALIB_RESULT_1_TRIO0_B**: The edge delay calibrate result, with offset.
15:8	0x0	**EDGE_DELAY_CALIB_RESULT_0_TRIO1_B**: The edge delay calibrate result, with offset.
7:0	0x0	**EDGE_DELAY_CALIB_RESULT_0_TRIO0_B**: The edge delay calibrate result, with offset.

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_INADJ_CALIB_CTRL_0

Offset: 0x1078e
Byte Offset: 0x41e38
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
16	0x0	**INADJ_CAL_TRIGGER_TRIO1_B**: Trigger INADJ calibration for trio0
0	0x0	**INADJ_CAL_TRIGGER_TRIO0_B**: Trigger INADJ calibration for trio1

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_INADJ_CALIB_CTRL1_0

Offset: 0x1078f
Byte Offset: 0x41e3c
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00010001 (0b0000,0000,0000,0001,0000,0000,0000,0001)

Bit	Reset	Description
31:16	0x1	**PRBS9_SEED_TRIO1_B**: PRBS9 initial seed for trio0
15:0	0x1	**PRBS9_SEED_TRIO0_B**: PRBS9 initial seed for trio1

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_INADJ_CALIB_CTRL2_0

Offset: 0x10790
Byte Offset: 0x41e40
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x0fc00fc0 (0bxxxx,1111,1100,0000,xxxx,1111,1100,0000)

Bit	Reset	Description
27:22	0x3f	**INADJ_LIMIT_HIGH_TRIO1_B**: The upper limit for the INADJ sweep.
21:16	0x0	**INADJ_LIMIT_LOW_TRIO1_B**: The lower limit for the INADJ sweep.
11:6	0x3f	**INADJ_LIMIT_HIGH_TRIO0_B**: The upper limit for the INADJ sweep.
5:0	0x0	**INADJ_LIMIT_LOW_TRIO0_B**: The lower limit for the INADJ sweep.

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_INADJ_CALIB_STATUS_0

Offset: 0x10791
Byte Offset: 0x41e44

Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**INADJ_CALIB_DONE_TRIO1_B**: Trio b1 inadj calibration done.
0	0x0	**INADJ_CALIB_DONE_TRIO0_B**: Trio b0 inadj calibration done.

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_INADJ_CALIB_STATUS1_0

Offset: 0x10792
Byte Offset: 0x41e48
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_LOW_TRIO0_B**: Trio b0 PRBS check status for lower 32 INADJ value.

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_INADJ_CALIB_STATUS2_0

Offset: 0x10793
Byte Offset: 0x41e4c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_LOW_TRIO1_B**: Trio b1 PRBS check status for lower 32 INADJ value.

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_INADJ_CALIB_STATUS3_0

Offset: 0x10794
Byte Offset: 0x41e50
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_HIGH_TRIO0_B**: Trio b0 PRBS check status for upper 32 INADJ value.

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_INADJ_CALIB_STATUS4_0

Offset: 0x10795
Byte Offset: 0x41e54
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**INADJ_CALIB_STATUS_HIGH_TRIO1_B**: Trio b1 PRBS check status for upper 32 INADJ value.

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_INADJ_OVERRIDE_0

Offset: 0x10796
Byte Offset: 0x41e58
Read/Write: R/W
Parity Protection: Y
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00080008 (0bxxxx,xxx0,0000,1000,xxxx,xxx0,0000,1000)

Bit	Reset	Description
24:17	0x4	**CPHY_INADJ_TRIO1_B**: The SW override INADJ value.
16	DISABLE	**SW_SET_CPHY_INADJ_TRIO1_B**: Enable SW override of the INADJ. **0** = DISABLE **1** = ENABLE
8:1	0x4	**CPHY_INADJ_TRIO0_B**: The SW override INADJ value.
0	DISABLE	**SW_SET_CPHY_INADJ_TRIO0_B**: Enable SW override of the INADJ. **0** = DISABLE **1** = ENABLE

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_BIST_CTRL_0

Offset: 0x10797
Byte Offset: 0x41e5c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	0x0	**STOP_BIST_TRIO1_B**: Manually stop the B1 BIST, write 1 to stop the RX bist.
2	0x0	**STOP_BIST_TRIO0_B**: Manually stop the B0 BIST, write 1 to stop the RX bist.
1	0x0	**TRIGGER_BIST_TRIO1_B**: Trigger the trio B1 BIST, write 1 to trigger the RX bist, HW will clear it to 0 when the BIST is done, write 0 to this field is with no effect.
0	0x0	**TRIGGER_BIST_TRIO0_B**: Trigger the trio B0 BIST, write 1 to trigger the RX bist, HW will clear it to 0 when the BIST is done, write 0 to this field is with no effect.

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_BIST_STATUS_0_0

Offset: 0x10798
Byte Offset: 0x41e60
Read/Write: RO
Parity Protection: N

Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:8	0x0	**WORD_ERR_CNT_TRIO1_B**: The trio B1 number of word comparison failure, report for BIST mode 1.
7:0	0x0	**WORD_ERR_CNT_TRIO0_B**: The trio B0 number of word comparison failure, report for BIST mode 1.

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_BIST_STATUS_1_0

Offset: 0x10799
Byte Offset: 0x41e64
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:8	0x0	**SYM_ERR_CNT_TRIO1_B**: The trio B1 number of symbol compariosn failure, it's the sum of the symbol err count and the clock shift count, the real wire state error number is 1/2 of this field, report for BIST mode 1.
7:0	0x0	**SYM_ERR_CNT_TRIO0_B**: The trio B0 number of symbol compariosn failure, it's the sum of the symbol err count and the clock shift count, the real wire state error number is 1/2 of this field, report for BIST mode 1.

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_BIST_STATUS_2_0

Offset: 0x1079a
Byte Offset: 0x41e68
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WORD_CNT_LOW_TRIO0_B**: The trio B0 low 32 bits of the compared word number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_BIST_STATUS_3_0

Offset: 0x1079b
Byte Offset: 0x41e6c
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WORD_CNT_HIGH_TRIO0_B**: The trio B0 high 16 bits of the compared word number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_BIST_STATUS_4_0

Offset: 0x1079c
Byte Offset: 0x41e70
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**WORD_CNT_LOW_TRIO1_B**: The trio B1 low 32 bits of the compared word number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_BIST_STATUS_5_0

Offset: 0x1079d
Byte Offset: 0x41e74
Read/Write: RO
Parity Protection: N
Shadow: N

Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WORD_CNT_HIGH_TRIO1_B**: The trio B1 high 16 bits of the compared word number number when detect the expected error number, report for BIST mode 0.

NVCSI_PHY_3_NVCSI_CIL_B_CPHY_BIST_STATUS_6_0

Offset: 0x1079e
Byte Offset: 0x41e78
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,x000,0000,0000,xxxx,x000,0000,0000)

Bit	Reset	Description
26	0x0	**ERR_DETECT_FAIL_TRIO1_B**: Multi symbol error in one word (7 symbol) for trio B1, BIST logic is not able to detect the accurate wire state error. With this error, if not other errors detected, the word error is accurate.
25	0x0	**SHIFT_DETECT_FAIL_TRIO1_B**: Detect symbol error in two continuous word (14 symbols) for trio B1, BIST logic is not able to determine if symbol clock is slip. (if the symbol error count is not saturating, the result is still available with this error)
24	0x0	**SHIFT_OVERFLOW_TRIO1_B**: Too many clocks slip and the BIST logic is not able to re-align for trio B1. Typically, when this error is detect, the symbol error counter will saturate.
23:20	0x0	**DOUBLE_CLK_MISSED_CNT_TRIO1_B**: Number of the times for double symbol clock lost for trio B1.
19:16	0x0	**SINGLE_CLK_MISSED_CNT_TRIO1_B**: Number of the times for single symbol clock lost for trio B1.
10	0x0	**ERR_DETECT_FAIL_TRIO0_B**: Multi symbol error in one word (7 symbol) for trio B0, BIST logic is not able to detect the accurate wire state error. With this error, if not other errors detected, the word error is accurate.
9	0x0	**SHIFT_DETECT_FAIL_TRIO0_B**: Detect symbol error in two continuous word (14 symbols) for trio B0, BIST logic is not able to determine if symbol clock is slip. (if the symbol error count is not saturating, the result is still available with this error)
8	0x0	**SHIFT_OVERFLOW_TRIO0_B**: Too many clocks slip and the BIST logic is not able to re-align for trio B0. Typically, when this error is detect, the symbol error counter will saturate.

Bit	Reset	Description
7:4	0x0	**DOUBLE_CLK_MISSED_CNT_TRIO0_B**: Number of the times for double symbol clock lost for trio B0.
3:0	0x0	**SINGLE_CLK_MISSED_CNT_TRIO0_B**: Number of the times for single symbol clock lost for trio B0.

NVCSI_PHY_GLOBAL_DEBUG_CONTROL_0_0

Offset: 0x12800
Byte Offset: 0x4a000
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,xxxx,x000)

Bit	Reset	Description
15:8	0x0	**DEBUG_0_EVENT_SEL**: Value ranges from 0 to 65, Selects the corresponding
2:0	NO_PHY	**DEBUG_0_PHY_SEL**: 0x1 to 0x4 = Selects PHY 0 to PHY 3 **0** = NO_PHY **1** = PHY_0 **2** = PHY_1 **3** = PHY_2 **4** = PHY_3

NVCSI_PHY_GLOBAL_DEBUG_CONTROL_1_0

Offset: 0x12801
Byte Offset: 0x4a004
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,xxxx,x000)

Bit	Reset	Description
15:8	0x0	**DEBUG_1_EVENT_SEL**: Value ranges from 0 to 65, Selects the corresponding

Bit	Reset	Description
2:0	NO_PHY	**DEBUG_1_PHY_SEL**: 0x1 to 0x4 = Selects PHY 0 to PHY 3 **0** = NO_PHY **1** = PHY_0 **2** = PHY_1 **3** = PHY_2 **4** = PHY_3

NVCSI_PHY_GLOBAL_DEBUG_CONTROL_2_0

Offset: 0x12802
Byte Offset: 0x4a008
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,xxxx,x000)

Bit	Reset	Description
15:8	0x0	**DEBUG_2_EVENT_SEL**: Value ranges from 0 to 65, Selects the corresponding
2:0	NO_PHY	**DEBUG_2_PHY_SEL**: 0x1 to 0x4 = Selects PHY 0 to PHY 3 **0** = NO_PHY **1** = PHY_0 **2** = PHY_1 **3** = PHY_2 **4** = PHY_3

NVCSI_PHY_GLOBAL_DEBUG_CONTROL_3_0

Offset: 0x12803
Byte Offset: 0x4a00c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,xxxx,x000)

Bit	Reset	Description
15:8	0x0	**DEBUG_3_EVENT_SEL**: Value ranges from 0 to 65, Selects the corresponding

Bit	Reset	Description
2:0	NO_PHY	**DEBUG_3_PHY_SEL**: 0x1 to 0x4 = Selects PHY 0 to PHY 3 **0** = NO_PHY **1** = PHY_0 **2** = PHY_1 **3** = PHY_2 **4** = PHY_3

NVCSI_PHY_GLOBAL_DEBUG_COUNTER_0_0

Offset: 0x12804
Byte Offset: 0x4a010
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DEBUG_COUNTER_0**: 32 bit debug counter for counting stream related events for debug. Writing "0x1" to this register clears DEBUG_COUNTER_0

NVCSI_PHY_GLOBAL_DEBUG_COUNTER_1_0

Offset: 0x12805
Byte Offset: 0x4a014
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DEBUG_COUNTER_1**: 32 bit debug counter for counting stream related events for debug. Writing "0x1" to this register clears DEBUG_COUNTER_1

NVCSI_PHY_GLOBAL_DEBUG_COUNTER_2_0

Offset: 0x12806

Byte Offset: 0x4a018
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DEBUG_COUNTER_2**: 32 bit debug counter for counting stream related events for debug. Writing "0x1" to this register clears DEBUG_COUNTER_2

NVCSI_PHY_GLOBAL_DEBUG_COUNTER_3_0

Offset: 0x12807
Byte Offset: 0x4a01c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DEBUG_COUNTER_3**: 32 bit debug counter for counting stream related events for debug. Writing "0x1" to this register clears DEBUG_COUNTER_3

NVCSI_PHY_GLOBAL_PLL_BASE_0

Offset: 0x12808
Byte Offset: 0x4a020
Read/Write: See table below
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00003302 (0bxxxx,xxxx,xxxx,xxxx,0011,0011,xx00,0010)

Bit	R/W	Reset	Description
15:8	RW	0x33	**PLL_MDIV**: Effective only when PLL_DIV_RATIO_OVERRIDE=1 PLL_MDIV[3:0] MIPI_PLL update rate 0 CLOCKIN / 1 1-15 CLOCKIN / PLL_MDIV[3:0]

Bit	R/W	Reset	Description
5	RW	0x0	**PLL_DIV_RATIO_OVERRIDE**: 0 -> Normal operation, M-divier/P-divider/N-divider/ CLKGEN_MODE/PDIV_MODE are controlled by SEL_DPHY_CPHY_PLL. 1 -> Enable the manual control of PLL_MDIV/P-divider/PLL_NDIV/ CLKGEN_MODE/PDIV_MODE. P-divider and N-divider are programmable through PLL_PDIV_RATIO and PLL_NDIV_RATIO.
4	RO	0x0	**PLL_LOCK**: PLL lock detector output (1 = locked, 0 = unlocked) De-assert when dynamic frequency change is initiated
3	RO	0x0	**PLL_FREQLOCK**: Frequency lock indicator FLL lock detector output (1 = locked, 0 = unlocked)
2	RW	0x0	**PLL_RESETB**: Digital force reset (Active low)
1	RW	0x1	**PLL_BYPASS**: 0 -> normal operation; 1 -> enable PLL bypass mode, allowing the reference clock to drive the output driver directly.
0	RW	0x0	**PLL_SEL_DPHY_CPHY**: CPHY/DPHY mode.

NVCSI_PHY_GLOBAL_PLL_MISC_0_0

Offset: 0x12809
Byte Offset: 0x4a024
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000007 (0b0000,0000,0000,0000,0000,0000,0000,0111)

Bit	Reset	Description
31:30	0x0	**PLL_LOOP_CTRL**: MIPI_PLL operation mode 0: FLL acquisition, PLL tracking 1: PLL acquisition, PLL tracking 2: FLL acquisition, PLL disabled 3: External DCO control (EXT_FRU[15:0])
29:24	0x0	**PLL_PDIV**: Effective only when PLL_DIV_RATIO_OVERRIDE=1 P-divider's division ratio: x000 -> divide-by-1; x001 -> divide-by-2; x010 -> divide-by-4; x011 -> divide-by-8. x1xx -> divide-by-16;

Bit	Reset	Description
23:8	0x0	**PLL_NDIV_FRAC**: Feedback divider fractional value (signed) Divide by NDIVeff = NDIV_INT[7:0] + (1/8192) x NDIV_FRAC[15:0] (maximum 1nS skew across all bits of NDIV_INT and NDIV_FRAC)
7:0	0x7	**PLL_NDIV_INT**: Effective only when PLL_DIV_RATIO_OVERRIDE=1 Feedback divider integer value. NDIV_INT[7:0] Divide value 0 1 1-255 1-255 (maximum 1nS skew across all bits of NDIV_INT and NDIV_FRAC)

NVCSI_PHY_GLOBAL_PLL_MISC_1_0

Offset: 0x1280a
Byte Offset: 0x4a028
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000008 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,1000)

Bit	Reset	Description
15:0	0x8	**PLL_EXT_FRU**: External frequency control value LOOP_CTRL = 0,1,2: ENABLE = 0 -> 1 causes DCO to be initialized to EXT_FRU[15:0] LOOP_CTRL = 3: ENABLE = 0 -> 1 causes DCO to free-run with control set to EXT_FRU[15:0]

NVCSI_PHY_GLOBAL_PLL_MISC_2_0

Offset: 0x1280b
Byte Offset: 0x4a02c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,xxxx,0000)

Bit	Reset	Description
31:24	0x0	**PLL_LOCKDET_MEM**: PLL error signal observation length for PLL lock detect PLL_LD_MEM[7:0]=0 disables the process that checks whether PLL is locked or not. After some clock delay, it forces PLL_LOCK=1 without ensuring if PLL has acquired phase and frequency.
23:16	0x0	**PLL_FRUG**: PLL frequency update gain PLL frequency gain = 2^(PLL_FRUG[7:0] - 4)
15:8	0x0	**PLL_FLL_LOCKDET_MEM**: FLL error signal observation length for FLL lock detect FLL_LD_MEM[7:0]=0 disables the process that checks whether FLL is locked or not. After some clock delay, it forces PLL_FREQLOCK=1 without ensuring if FLL has acquired the frequency.
3	0x0	**PLL_FLL_DIV**: FLL update rate divider 0: divide by 2 1: divide by 4
2:0	0x0	**PLL_FLL_FRUG**: FLL frequency update gain FLL gain = 2^(FLL_FRUG[2:0] - 7)

NVCSI_PHY_GLOBAL_PLL_MISC_3_0

Offset: 0x1280c
Byte Offset: 0x4a030
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,xx00,0000)

Bit	Reset	Description
29:27	0x0	**PLL_KP_STEP_TIMER**: KP Gear shifting timer.
26:24	0x0	**PLL_FRAC_STEP_TIMER**: Time between fractional steps. Value of zero is illegal
23:8	0x0	**PLL_FRAC_STEP**: Fractional step size (signed)
5:4	0x0	**PLL_DSM_DIV**: Control DSM running frequency, (divided-down version of dco frequency, fdco) 0: fdco / 2 1: fdco 2: fdco / 4 3: fdco / 8 Default = 0 (en_ssc = 0) or 1 (en_ssc = 1)

Bit	Reset	Description
3:0	0x0	**PLL_LOCKDET_TOL**: PLL error signal tolerance level for PLL phase and frequency lock condition

NVCSI_PHY_GLOBAL_PLL_MISC_4_0

Offset: 0x1280d
Byte Offset: 0x4a034
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,xx00,0000)

Bit	Reset	Description
23:8	0x0	**PLL_SETUP**
5:3	0x0	**PLL_KP_LO**: Analog phase control gain (PLL bandwidth definition) 0: 5% CCO current consumption 1: 10% CCO current consumption 2: 15% CCO current consumption 3: 20% CCO current consumption 4: 25% CCO current consumption 5: 30% CCO current consumption 6: 35% CCO current consumption 7: 40% CCO current consumption
2:0	0x0	**PLL_KP_HI**: Analog phase control gain (PLL bandwidth definition) 0: 5% CCO current consumption 1: 10% CCO current consumption 2: 15% CCO current consumption 3: 20% CCO current consumption 4: 25% CCO current consumption 5: 30% CCO current consumption 6: 35% CCO current consumption 7: 40% CCO current consumption

NVCSI_PHY_GLOBAL_PLL_MISC_5_0

Offset: 0x1280e
Byte Offset: 0x4a038
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected

SCR Protection: 0
Reset: 0x00003c00 (0b0000,0000,0000,0000,0011,1100,0000,0000)

Bit	Reset	Description
31:0	0x3c00	**PLL_CTRL**

NVCSI_PHY_GLOBAL_PLL_MISC_6_0
Offset: 0x1280f
Byte Offset: 0x4a03c
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,xx00,0000)

Bit	Reset	Description
11:8	0x0	**PLL_TESTMUX**: Mux select - analog/digital output seen on RDN
5:4	0x0	**PLL_REG_AVDD09_LOAD**: Internal regulated 0.9v extra loading for stability control bits
3:0	0x0	**PLL_REG_AVDD09_LEVEL**: Internal regulated 1.0v voltage level control bits: X000 -> 0.720v (0.9v-20%); X001 -> 0.765v (0.9v-15%); X010 -> 0.810v (0.9v-10%); X011 -> 0.855v (0.9v-5%); X100 -> 0.900v; X101 -> 0.945v (0.9v+5%); X110 -> 0.990v (0.9v+10%); X111 -> 1.035v (0.9v+15%).

7.2.1.3.9 NVCSI ERR Registers

NVCSI_ERR_COLLATOR_FEATURE_0

GLOBAL SLICE

Offset: 0x12c00
Byte Offset: 0x4b000
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00190001 (0b0000,0000,0001,1001,xxxx,xxxx,xx00,0001)

Bit	Reset	Description
31:16	0x19	**NUM_ERR**: Number of errors connected to this collator. This is passed as a build time option to the plugin
5:0	0x1	**NUM_ERR_SLICES**: Number of error slices supported by this error collator, does not include the GlobalSpace and is derived by ceil (NUM_ERR/32). SW shall first read this register to determine the number of slices and read the required number of Error_Status registers .

NVCSI_ERR_COLLATOR_SWRESET_0

Offset: 0x12c01
Byte Offset: 0x4b004
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SWRST**: 1'b1 : Issue a SW reset to the Error Collator. This will reset all the registers(Except SCR), counters and logic of the Error Collator. SW can use this bit to flush errors logged into the error collator for ex, after Boot, SC7/8 exit. 1'b0 : Do nothing, reset value. This bit is auto-cleared.

NVCSI_ERR_COLLATOR_MISSIONERR_TYPE_0

Offset: 0x12c02
Byte Offset: 0x4b008
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0101)

Bit	Reset	Description
5:0	0x5	**CODE**: This register indicates the fault code of the error line based on the value of MISSIONERR_INDEX Register. This can be used by a fault handling agent to triage an error without requiring device-specific code.The possible values of this field are: 6'd0 : None 6'd1 : Parity Error on internal data path 6'd2 : ECC SEC Error on internal data path 6'd3 : ECC DED Error on internal data path 6'd4 : Comparator Error 6'd5 : Register Parity Error 6'd6 : Parity Error from on-chip SRAM/FIFO 6'd7 : ECC SEC Error from on-chip SRAM/FIFO 6'd8 : ECC DED Error from on-chip SRAM/FIFO 6'd9 : Clock Monitor Error 6'd10 : Voltage Error 6'd11 : Temperature Error 6'd12 : CRC Error on internal data path 6'd16 : SW Corrected Error 6'd17 : SW Uncorrected Error 6'd18 : SW Generic Error 6'd20 : ECC SED Error on internal data path 6'd21 : ECC SED Error from on-chip SRAM/FIFO 6'd32 : Other HW Corrected Error 6'd33 : Other HW Uncorrected Error All other values : Reserved for future use.

NVCSI_ERR_COLLATOR_CURRENT_COUNTER_VALUE_0

Offset: 0x12c03
Byte Offset: 0x4b00c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
8:0	0x0	**VALUE**: Provides the current value of the counter corresponding to the error in MissionErr_Index Register. Default provides the value of error 0 counter. Bit[8] is the ovrerflow bit post which the counter saturates and does not counter further.

NVCSI_ERR_COLLATOR_MISSIONERR_USERVALUE_0

Offset: 0x12c04
Byte Offset: 0x4b010
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0xdead0000 (0b1101,1110,1010,1101,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0xdead0000	**VALUE**: Reflects the value on the error_<i>_user when error_<i>_pulse = MissionError. Only the user signals corresponding to the first error_<i>_pulse only will b elatched.Subsequent values will be dropped. The signals will get latched only when ERRSLICE<n>_MISSIONERR_STATUS is CLEAR. IPs must tie off the unused bits of this signals to 0, if unused.

NVCSI_ERR_COLLATOR_MISSIONERR_INDEX_0

Offset: 0x12c05
Byte Offset: 0x4b014
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4:0	0x0	**IDX**: BINARY Encoded. For error number 32, register should be programmed with value 0x20. Write to this register with Error number will update: - MISSIONERR_TYPE Register with the Error-Code for the Error. - CURRENT_COUNTER_VALUE Register with the error's SEC/DED Counter. - MISSIONERR_USERVALUE with value of the first error_<i>_user signal. SW can use this to trage the error. number shall update the MISSIONERR_TYPE register with the error code and the Current_Counter_Value register with the value of the errors SEC/DED counter. SW can use this register to triage the error.

NVCSI_ERR_COLLATOR_CORRECTABLE_THRESHOLD_0

Offset: 0x12c06
Byte Offset: 0x4b018
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**COUNT**: Threshold value for all SEC Fault Reporting Units connected to this error collator. SEC Errors are logged once the threshold is reached and the overflow bit is set. 7'b0 : Log SEC error after receiving 1 Error. 7'b1 : Log SEC error after receiving 2 Errors. ... 7'bFF : Log SEC error after receiving 256 Errors.

NVCSI_ERR_COLLATOR_MISSIONERR_INJECT_UNLOCK_0

Offset: 0x12c07
Byte Offset: 0x4b01c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Writes to ERRSLICE_XXX_MISSIONERR_INJECT registers are disabled until this register is written with a value of 0xE1. This is to prevent an inadvertent safety error injection in the safety plugin due to: 1. A fault on ERRSLICE_XXX_MISSIONERR_INJECT register itself. 2. Erroneous SW. The register shall be written with a value of 0x0 to reestablish the lock after user has completed the error injection testing. 0xE1 : Unlock the MISSIONERR_INJECT Register 0x0 : Lock the MISSIONERR_INJECT Register **0** = LOCK **225** = UNLOCK

NVCSI_ERR_COLLATOR_ERRSLICE0_MISSIONERR_ENABLE_0

ERROR SLICE - 0

Offset: 0x12c0c
Byte Offset: 0x4b030
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: 0
Reset: 0x03ffffff (0bxxxx,xx11,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
25	0x1	**ERR25**: 1'b1 -> Enable Mission Error Reporting for SW Generic Error from NV_nvcsi_top.u_NV_nvcsi.u_safety 1'b0 -> Disable Mission Error Reporting for SW Generic Error from NV_nvcsi_top.u_NV_nvcsi.u_safety **0** = DISABLE **1** = ENABLE
24	0x1	**ERR24**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from NV_nvcsi_top.u_NV_nvcsicil.u_regblk.u_arreggen 1'b0 -> Disable Mission Error Reporting for Register Parity Error from NV_nvcsi_top.u_NV_nvcsicil.u_regblk.u_arreggen **0** = DISABLE **1** = ENABLE
23	0x1	**ERR23**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from NV_nvcsi_top.u_NV_nvcsi.u_nvcsi_host1x.u_regblk.u_arreggen 1'b0 -> Disable Mission Error Reporting for Register Parity Error from NV_nvcsi_top.u_NV_nvcsi.u_nvcsi_host1x.u_regblk.u_arreggen **0** = DISABLE **1** = ENABLE
22	0x1	**ERR22**: 1'b1 -> Enable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_HOST1X_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Disable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_HOST1X_OTHER_UNCORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
21	0x1	**ERR21**: 1'b1 -> Enable Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_HOST1X_OTHER_CORRECTABLE_ERROR 1'b0 -> Disable Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_HOST1X_OTHER_CORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
20	0x1	**ERR20**: 1'b1 -> Enable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY3_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Disable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY3_OTHER_UNCORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
19	0x1	**ERR19**: 1'b1 -> Enable Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY3_OTHER_CORRECTABLE_ERROR 1'b0 -> Disable Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY3_OTHER_CORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
18	0x1	**ERR18**: 1'b1 -> Enable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY2_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Disable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY2_OTHER_UNCORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
17	0x1	**ERR17**: 1'b1 -> Enable Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY2_OTHER_CORRECTABLE_ERROR 1'b0 -> Disable Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY2_OTHER_CORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
16	0x1	**ERR16**: 1'b1 -> Enable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY1_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Disable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY1_OTHER_UNCORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
15	0x1	**ERR15**: 1'b1 -> Enable Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY1_OTHER_CORRECTABLE_ERROR 1'b0 -> Disable Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY1_OTHER_CORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
14	0x1	**ERR14**: 1'b1 -> Enable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY0_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Disable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY0_OTHER_UNCORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
13	0x1	**ERR13**: 1'b1 -> Enable Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY0_OTHER_CORRECTABLE_ERROR 1'b0 -> Disable Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY0_OTHER_CORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
12	0x1	**ERR12**: 1'b1 -> Enable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM5_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Disable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM5_OTHER_UNCORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
11	0x1	**ERR11**: 1'b1 -> Enable Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM5_OTHER_CORRECTABLE_ERROR 1'b0 -> Disable Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM5_OTHER_CORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
10	0x1	**ERR10**: 1'b1 -> Enable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM4_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Disable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM4_OTHER_UNCORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
9	0x1	**ERR9**: 1'b1 -> Enable Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM4_OTHER_CORRECTABLE_ERROR 1'b0 -> Disable Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM4_OTHER_CORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
8	0x1	**ERR8**: 1'b1 -> Enable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM3_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Disable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM3_OTHER_UNCORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
7	0x1	**ERR7**: 1'b1 -> Enable Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM3_OTHER_CORRECTABLE_ERROR 1'b0 -> Disable Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM3_OTHER_CORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
6	0x1	**ERR6**: 1'b1 -> Enable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM2_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Disable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM2_OTHER_UNCORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
5	0x1	**ERR5**: 1'b1 -> Enable Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM2_OTHER_CORRECTABLE_ERROR 1'b0 -> Disable Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM2_OTHER_CORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
4	0x1	**ERR4**: 1'b1 -> Enable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM1_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Disable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM1_OTHER_UNCORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
3	0x1	**ERR3**: 1'b1 -> Enable Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM1_OTHER_CORRECTABLE_ERROR 1'b0 -> Disable Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM1_OTHER_CORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
2	0x1	**ERR2**: 1'b1 -> Enable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM0_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Disable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM0_OTHER_UNCORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM0_OTHER_CORRECTABLE_ERROR 1'b0 -> Disable Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM0_OTHER_CORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from NV_NVCSI_SAFETY_err_collator 1'b0 -> Disable Mission Error Reporting for Register Parity Error from NV_NVCSI_SAFETY_err_collator **0** = DISABLE **1** = ENABLE

NVCSI_ERR_COLLATOR_ERRSLICE0_MISSIONERR_FORCE_0

Offset: 0x12c0d
Byte Offset: 0x4b034
Read/Write: WO

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ERR25**: 1'b1 -> Force Assertion of Mission Error Reporting for SW Generic Error from NV_nvcsi_top.u_NV_nvcsi.u_safety 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
24	0x0	**ERR24**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from NV_nvcsi_top.u_NV_nvcsicil.u_regblk.u_arreggen 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
23	0x0	**ERR23**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from NV_nvcsi_top.u_NV_nvcsi.u_nvcsi_host1x.u_regblk.u_arreggen 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
22	0x0	**ERR22**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_HOST1X_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
21	0x0	**ERR21**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_HOST1X_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
20	0x0	**ERR20**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY3_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
19	0x0	**ERR19**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY3_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
18	0x0	**ERR18**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY2_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
17	0x0	**ERR17**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY2_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
16	0x0	**ERR16**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY1_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
15	0x0	**ERR15**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY1_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
14	0x0	**ERR14**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY0_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
13	0x0	**ERR13**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY0_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
12	0x0	**ERR12**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM5_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
11	0x0	**ERR11**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM5_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
10	0x0	**ERR10**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM4_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
9	0x0	**ERR9**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM4_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
8	0x0	**ERR8**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM3_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
7	0x0	**ERR7**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM3_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
6	0x0	**ERR6**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM2_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
5	0x0	**ERR5**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM2_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
4	0x0	**ERR4**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM1_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
3	0x0	**ERR3**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM1_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM0_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM0_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from NV_NVCSI_SAFETY_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

NVCSI_ERR_COLLATOR_ERRSLICE0_MISSIONERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of MissionError_Enable register, to avoid silent dropping of errors.
Offset: 0x12c0e
Byte Offset: 0x4b038
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ERR25**: 1'b1 -> Error_25_pulse[1:0] for SW Generic Error from NV_nvcsi_top.u_NV_nvcsi.u_safety was equal to 2'b10. 1'b0 -> Error_25_pulse[1:0] for SW Generic Error from NV_nvcsi_top.u_NV_nvcsi.u_safety was equal to 2'b01.
24	0x0	**ERR24**: 1'b1 -> Error_24_pulse[1:0] for Register Parity Error from NV_nvcsi_top.u_NV_nvcsicil.u_regblk.u_arreggen was equal to 2'b10. 1'b0 -> Error_24_pulse[1:0] for Register Parity Error from NV_nvcsi_top.u_NV_nvcsicil.u_regblk.u_arreggen was equal to 2'b01.
23	0x0	**ERR23**: 1'b1 -> Error_23_pulse[1:0] for Register Parity Error from NV_nvcsi_top.u_NV_nvcsi.u_nvcsi_host1x.u_regblk.u_arreggen was equal to 2'b10. 1'b0 -> Error_23_pulse[1:0] for Register Parity Error from NV_nvcsi_top.u_NV_nvcsi.u_nvcsi_host1x.u_regblk.u_arreggen was equal to 2'b01.
22	0x0	**ERR22**: 1'b1 -> Error_22_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_HOST1X_OTHER_UNCORRECTABLE_ERROR was equal to 2'b10. 1'b0 -> Error_22_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_HOST1X_OTHER_UNCORRECTABLE_ERROR was equal to 2'b01.
21	0x0	**ERR21**: 1'b1 -> Error_21_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSI_HOST1X_OTHER_CORRECTABLE_ERROR was equal to 2'b10. 1'b0 -> Error_21_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSI_HOST1X_OTHER_CORRECTABLE_ERROR was equal to 2'b01.
20	0x0	**ERR20**: 1'b1 -> Error_20_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY3_OTHER_UNCORRECTABLE_ERROR was equal to 2'b10. 1'b0 -> Error_20_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY3_OTHER_UNCORRECTABLE_ERROR was equal to 2'b01.
19	0x0	**ERR19**: 1'b1 -> Error_19_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY3_OTHER_CORRECTABLE_ERROR was equal to 2'b10. 1'b0 -> Error_19_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY3_OTHER_CORRECTABLE_ERROR was equal to 2'b01.
18	0x0	**ERR18**: 1'b1 -> Error_18_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY2_OTHER_UNCORRECTABLE_ERROR was equal to 2'b10. 1'b0 -> Error_18_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY2_OTHER_UNCORRECTABLE_ERROR was equal to 2'b01.

Bit	Reset	Description
17	0x0	**ERR17**: 1'b1 -> Error_17_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY2_OTHER_CORRECTABLE_ERROR was equal to 2'b10. 1'b0 -> Error_17_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY2_OTHER_CORRECTABLE_ERROR was equal to 2'b01.
16	0x0	**ERR16**: 1'b1 -> Error_16_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY1_OTHER_UNCORRECTABLE_ERROR was equal to 2'b10. 1'b0 -> Error_16_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY1_OTHER_UNCORRECTABLE_ERROR was equal to 2'b01.
15	0x0	**ERR15**: 1'b1 -> Error_15_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY1_OTHER_CORRECTABLE_ERROR was equal to 2'b10. 1'b0 -> Error_15_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY1_OTHER_CORRECTABLE_ERROR was equal to 2'b01.
14	0x0	**ERR14**: 1'b1 -> Error_14_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY0_OTHER_UNCORRECTABLE_ERROR was equal to 2'b10. 1'b0 -> Error_14_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY0_OTHER_UNCORRECTABLE_ERROR was equal to 2'b01.
13	0x0	**ERR13**: 1'b1 -> Error_13_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY0_OTHER_CORRECTABLE_ERROR was equal to 2'b10. 1'b0 -> Error_13_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY0_OTHER_CORRECTABLE_ERROR was equal to 2'b01.
12	0x0	**ERR12**: 1'b1 -> Error_12_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM5_OTHER_UNCORRECTABLE_ERROR was equal to 2'b10. 1'b0 -> Error_12_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM5_OTHER_UNCORRECTABLE_ERROR was equal to 2'b01.
11	0x0	**ERR11**: 1'b1 -> Error_11_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM5_OTHER_CORRECTABLE_ERROR was equal to 2'b10. 1'b0 -> Error_11_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM5_OTHER_CORRECTABLE_ERROR was equal to 2'b01.
10	0x0	**ERR10**: 1'b1 -> Error_10_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM4_OTHER_UNCORRECTABLE_ERROR was equal to 2'b10. 1'b0 -> Error_10_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM4_OTHER_UNCORRECTABLE_ERROR was equal to 2'b01.

Bit	Reset	Description
9	0x0	**ERR9**: 1'b1 -> Error_9_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM4_OTHER_CORRECTABLE_ERROR was equal to 2'b10. 1'b0 -> Error_9_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM4_OTHER_CORRECTABLE_ERROR was equal to 2'b01.
8	0x0	**ERR8**: 1'b1 -> Error_8_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM3_OTHER_UNCORRECTABLE_ERROR was equal to 2'b10. 1'b0 -> Error_8_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM3_OTHER_UNCORRECTABLE_ERROR was equal to 2'b01.
7	0x0	**ERR7**: 1'b1 -> Error_7_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM3_OTHER_CORRECTABLE_ERROR was equal to 2'b10. 1'b0 -> Error_7_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM3_OTHER_CORRECTABLE_ERROR was equal to 2'b01.
6	0x0	**ERR6**: 1'b1 -> Error_6_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM2_OTHER_UNCORRECTABLE_ERROR was equal to 2'b10. 1'b0 -> Error_6_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM2_OTHER_UNCORRECTABLE_ERROR was equal to 2'b01.
5	0x0	**ERR5**: 1'b1 -> Error_5_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM2_OTHER_CORRECTABLE_ERROR was equal to 2'b10. 1'b0 -> Error_5_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM2_OTHER_CORRECTABLE_ERROR was equal to 2'b01.
4	0x0	**ERR4**: 1'b1 -> Error_4_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM1_OTHER_UNCORRECTABLE_ERROR was equal to 2'b10. 1'b0 -> Error_4_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM1_OTHER_UNCORRECTABLE_ERROR was equal to 2'b01.
3	0x0	**ERR3**: 1'b1 -> Error_3_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM1_OTHER_CORRECTABLE_ERROR was equal to 2'b10. 1'b0 -> Error_3_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM1_OTHER_CORRECTABLE_ERROR was equal to 2'b01.
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM0_OTHER_UNCORRECTABLE_ERROR was equal to 2'b10. 1'b0 -> Error_2_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM0_OTHER_UNCORRECTABLE_ERROR was equal to 2'b01.

Bit	Reset	Description
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM0_OTHER_CORRECTABLE_ERROR was equal to 2'b10. 1'b0 -> Error_1_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM0_OTHER_CORRECTABLE_ERROR was equal to 2'b01.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from NV_NVCSI_SAFETY_err_collator was equal to 2'b10. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from NV_NVCSI_SAFETY_err_collator was equal to 2'b01.

NVCSI_ERR_COLLATOR_ERRSLICE0_MISSIONERR_INJECT_0

Offset: 0x12c0f
Byte Offset: 0x4b03c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
24	0x0	**ERR24**: 1'b1 -> Assert the inject_error_24 output for Register Parity Error to NV_nvcsi_top.u_NV_nvcsicil.u_regblk.u_arreggen to allow for error injection. 1'b0 -> De-Asseert inject_error_24 output. **0** = DISABLE **1** = ENABLE
23	0x0	**ERR23**: 1'b1 -> Assert the inject_error_23 output for Register Parity Error to NV_nvcsi_top.u_NV_nvcsi.u_nvcsi_host1x.u_regblk.u_arreggen to allow for error injection. 1'b0 -> De-Asseert inject_error_23 output. **0** = DISABLE **1** = ENABLE
22	0x0	**ERR22**: 1'b1 -> Assert the inject_error_22 output for Other Uncorrected Error to FAULT_UDF_TAG_NVCSI_HOST1X_OTHER_UNCORRECTABLE_ERROR to allow for error injection. 1'b0 -> De-Asseert inject_error_22 output. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
21	0x0	**ERR21**: 1'b1 -> Assert the inject_error_21 output for Other Corrected Error to FAULT_UDF_TAG_NVCSI_HOST1X_OTHER_CORRECTABLE_ERROR to allow for error injection. 1'b0 -> De-Asseert inject_error_21 output. **0** = DISABLE **1** = ENABLE
20	0x0	**ERR20**: 1'b1 -> Assert the inject_error_20 output for Other Uncorrected Error to FAULT_UDF_TAG_NVCSICIL_PHY3_OTHER_UNCORRECTABLE_ERROR to allow for error injection. 1'b0 -> De-Asseert inject_error_20 output. **0** = DISABLE **1** = ENABLE
19	0x0	**ERR19**: 1'b1 -> Assert the inject_error_19 output for Other Corrected Error to FAULT_UDF_TAG_NVCSICIL_PHY3_OTHER_CORRECTABLE_ERROR to allow for error injection. 1'b0 -> De-Asseert inject_error_19 output. **0** = DISABLE **1** = ENABLE
18	0x0	**ERR18**: 1'b1 -> Assert the inject_error_18 output for Other Uncorrected Error to FAULT_UDF_TAG_NVCSICIL_PHY2_OTHER_UNCORRECTABLE_ERROR to allow for error injection. 1'b0 -> De-Asseert inject_error_18 output. **0** = DISABLE **1** = ENABLE
17	0x0	**ERR17**: 1'b1 -> Assert the inject_error_17 output for Other Corrected Error to FAULT_UDF_TAG_NVCSICIL_PHY2_OTHER_CORRECTABLE_ERROR to allow for error injection. 1'b0 -> De-Asseert inject_error_17 output. **0** = DISABLE **1** = ENABLE
16	0x0	**ERR16**: 1'b1 -> Assert the inject_error_16 output for Other Uncorrected Error to FAULT_UDF_TAG_NVCSICIL_PHY1_OTHER_UNCORRECTABLE_ERROR to allow for error injection. 1'b0 -> De-Asseert inject_error_16 output. **0** = DISABLE **1** = ENABLE
15	0x0	**ERR15**: 1'b1 -> Assert the inject_error_15 output for Other Corrected Error to FAULT_UDF_TAG_NVCSICIL_PHY1_OTHER_CORRECTABLE_ERROR to allow for error injection. 1'b0 -> De-Asseert inject_error_15 output. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
14	0x0	**ERR14**: 1'b1 -> Assert the inject_error_14 output for Other Uncorrected Error to FAULT_UDF_TAG_NVCSICIL_PHY0_OTHER_UNCORRECTABLE_ERROR to allow for error injection. 1'b0 -> De-Asseert inject_error_14 output. **0** = DISABLE **1** = ENABLE
13	0x0	**ERR13**: 1'b1 -> Assert the inject_error_13 output for Other Corrected Error to FAULT_UDF_TAG_NVCSICIL_PHY0_OTHER_CORRECTABLE_ERROR to allow for error injection. 1'b0 -> De-Asseert inject_error_13 output. **0** = DISABLE **1** = ENABLE
12	0x0	**ERR12**: 1'b1 -> Assert the inject_error_12 output for Other Uncorrected Error to FAULT_UDF_TAG_NVCSI_STREAM5_OTHER_UNCORRECTABLE_ERROR to allow for error injection. 1'b0 -> De-Asseert inject_error_12 output. **0** = DISABLE **1** = ENABLE
11	0x0	**ERR11**: 1'b1 -> Assert the inject_error_11 output for Other Corrected Error to FAULT_UDF_TAG_NVCSI_STREAM5_OTHER_CORRECTABLE_ERROR to allow for error injection. 1'b0 -> De-Asseert inject_error_11 output. **0** = DISABLE **1** = ENABLE
10	0x0	**ERR10**: 1'b1 -> Assert the inject_error_10 output for Other Uncorrected Error to FAULT_UDF_TAG_NVCSI_STREAM4_OTHER_UNCORRECTABLE_ERROR to allow for error injection. 1'b0 -> De-Asseert inject_error_10 output. **0** = DISABLE **1** = ENABLE
9	0x0	**ERR9**: 1'b1 -> Assert the inject_error_9 output for Other Corrected Error to FAULT_UDF_TAG_NVCSI_STREAM4_OTHER_CORRECTABLE_ERROR to allow for error injection. 1'b0 -> De-Asseert inject_error_9 output. **0** = DISABLE **1** = ENABLE
8	0x0	**ERR8**: 1'b1 -> Assert the inject_error_8 output for Other Uncorrected Error to FAULT_UDF_TAG_NVCSI_STREAM3_OTHER_UNCORRECTABLE_ERROR to allow for error injection. 1'b0 -> De-Asseert inject_error_8 output. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
7	0x0	**ERR7**: 1'b1 -> Assert the inject_error_7 output for Other Corrected Error to FAULT_UDF_TAG_NVCSI_STREAM3_OTHER_CORRECTABLE_ERROR to allow for error injection. 1'b0 -> De-Asseert inject_error_7 output. **0** = DISABLE **1** = ENABLE
6	0x0	**ERR6**: 1'b1 -> Assert the inject_error_6 output for Other Uncorrected Error to FAULT_UDF_TAG_NVCSI_STREAM2_OTHER_UNCORRECTABLE_ERROR to allow for error injection. 1'b0 -> De-Asseert inject_error_6 output. **0** = DISABLE **1** = ENABLE
5	0x0	**ERR5**: 1'b1 -> Assert the inject_error_5 output for Other Corrected Error to FAULT_UDF_TAG_NVCSI_STREAM2_OTHER_CORRECTABLE_ERROR to allow for error injection. 1'b0 -> De-Asseert inject_error_5 output. **0** = DISABLE **1** = ENABLE
4	0x0	**ERR4**: 1'b1 -> Assert the inject_error_4 output for Other Uncorrected Error to FAULT_UDF_TAG_NVCSI_STREAM1_OTHER_UNCORRECTABLE_ERROR to allow for error injection. 1'b0 -> De-Asseert inject_error_4 output. **0** = DISABLE **1** = ENABLE
3	0x0	**ERR3**: 1'b1 -> Assert the inject_error_3 output for Other Corrected Error to FAULT_UDF_TAG_NVCSI_STREAM1_OTHER_CORRECTABLE_ERROR to allow for error injection. 1'b0 -> De-Asseert inject_error_3 output. **0** = DISABLE **1** = ENABLE
2	0x0	**ERR2**: 1'b1 -> Assert the inject_error_2 output for Other Uncorrected Error to FAULT_UDF_TAG_NVCSI_STREAM0_OTHER_UNCORRECTABLE_ERROR to allow for error injection. 1'b0 -> De-Asseert inject_error_2 output. **0** = DISABLE **1** = ENABLE
1	0x0	**ERR1**: 1'b1 -> Assert the inject_error_1 output for Other Corrected Error to FAULT_UDF_TAG_NVCSI_STREAM0_OTHER_CORRECTABLE_ERROR to allow for error injection. 1'b0 -> De-Asseert inject_error_1 output. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
0	0x0	**ERR0**: 1'b1 -> Assert the inject_error_0 output for Register Parity Error to NV_NVCSI_SAFETY_err_collator to allow for error injection. 1'b0 -> De-Asseert inject_error_0 output. **0** = DISABLE **1** = ENABLE

NVCSI_ERR_COLLATOR_ERRSLICE0_LATENTERR_ENABLE_0

Offset: 0x12c10
Byte Offset: 0x4b040
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x03ffffff (0bxxxx,xx11,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
25	0x1	**ERR25**: 1'b1 -> Enable Latent Error Reporting for SW Generic Error from NV_nvcsi_top.u_NV_nvcsi.u_safety 1'b0 -> Disable Latent Error Reporting for SW Generic Error from NV_nvcsi_top.u_NV_nvcsi.u_safety **0** = DISABLE **1** = ENABLE
24	0x1	**ERR24**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from NV_nvcsi_top.u_NV_nvcsicil.u_regblk.u_arreggen 1'b0 -> Disable Latent Error Reporting for Register Parity Error from NV_nvcsi_top.u_NV_nvcsicil.u_regblk.u_arreggen **0** = DISABLE **1** = ENABLE
23	0x1	**ERR23**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from NV_nvcsi_top.u_NV_nvcsi.u_nvcsi_host1x.u_regblk.u_arreggen 1'b0 -> Disable Latent Error Reporting for Register Parity Error from NV_nvcsi_top.u_NV_nvcsi.u_nvcsi_host1x.u_regblk.u_arreggen **0** = DISABLE **1** = ENABLE
22	0x1	**ERR22**: 1'b1 -> Enable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_HOST1X_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Disable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_HOST1X_OTHER_UNCORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
21	0x1	**ERR21**: 1'b1 -> Enable Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_HOST1X_OTHER_CORRECTABLE_ERROR 1'b0 -> Disable Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_HOST1X_OTHER_CORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
20	0x1	**ERR20**: 1'b1 -> Enable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY3_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Disable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY3_OTHER_UNCORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
19	0x1	**ERR19**: 1'b1 -> Enable Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY3_OTHER_CORRECTABLE_ERROR 1'b0 -> Disable Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY3_OTHER_CORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
18	0x1	**ERR18**: 1'b1 -> Enable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY2_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Disable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY2_OTHER_UNCORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
17	0x1	**ERR17**: 1'b1 -> Enable Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY2_OTHER_CORRECTABLE_ERROR 1'b0 -> Disable Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY2_OTHER_CORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
16	0x1	**ERR16**: 1'b1 -> Enable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY1_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Disable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY1_OTHER_UNCORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
15	0x1	**ERR15**: 1'b1 -> Enable Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY1_OTHER_CORRECTABLE_ERROR 1'b0 -> Disable Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY1_OTHER_CORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
14	0x1	**ERR14**: 1'b1 -> Enable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY0_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Disable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY0_OTHER_UNCORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
13	0x1	**ERR13**: 1'b1 -> Enable Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY0_OTHER_CORRECTABLE_ERROR 1'b0 -> Disable Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY0_OTHER_CORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
12	0x1	**ERR12**: 1'b1 -> Enable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM5_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Disable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM5_OTHER_UNCORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
11	0x1	**ERR11**: 1'b1 -> Enable Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM5_OTHER_CORRECTABLE_ERROR 1'b0 -> Disable Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM5_OTHER_CORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
10	0x1	**ERR10**: 1'b1 -> Enable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM4_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Disable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM4_OTHER_UNCORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
9	0x1	**ERR9**: 1'b1 -> Enable Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM4_OTHER_CORRECTABLE_ERROR 1'b0 -> Disable Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM4_OTHER_CORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
8	0x1	**ERR8**: 1'b1 -> Enable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM3_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Disable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM3_OTHER_UNCORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
7	0x1	**ERR7**: 1'b1 -> Enable Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM3_OTHER_CORRECTABLE_ERROR 1'b0 -> Disable Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM3_OTHER_CORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
6	0x1	**ERR6**: 1'b1 -> Enable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM2_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Disable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM2_OTHER_UNCORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
5	0x1	**ERR5**: 1'b1 -> Enable Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM2_OTHER_CORRECTABLE_ERROR 1'b0 -> Disable Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM2_OTHER_CORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
4	0x1	**ERR4**: 1'b1 -> Enable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM1_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Disable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM1_OTHER_UNCORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
3	0x1	**ERR3**: 1'b1 -> Enable Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM1_OTHER_CORRECTABLE_ERROR 1'b0 -> Disable Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM1_OTHER_CORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
2	0x1	**ERR2**: 1'b1 -> Enable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM0_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Disable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM0_OTHER_UNCORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM0_OTHER_CORRECTABLE_ERROR 1'b0 -> Disable Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM0_OTHER_CORRECTABLE_ERROR **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
0	0x1	**ERR0**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from NV_NVCSI_SAFETY_err_collator 1'b0 -> Disable Latent Error Reporting for Register Parity Error from NV_NVCSI_SAFETY_err_collator **0** = DISABLE **1** = ENABLE

NVCSI_ERR_COLLATOR_ERRSLICE0_LATENTERR_FORCE_0

Offset: 0x12c11
Byte Offset: 0x4b044
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ERR25**: 1'b1 -> Force Assertion of Latent Error Reporting for SW Generic Error from NV_nvcsi_top.u_NV_nvcsi.u_safety 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
24	0x0	**ERR24**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from NV_nvcsi_top.u_NV_nvcsicil.u_regblk.u_arreggen 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
23	0x0	**ERR23**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from NV_nvcsi_top.u_NV_nvcsi.u_nvcsi_host1x.u_regblk.u_arreggen 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
22	0x0	**ERR22**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_HOST1X_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
21	0x0	**ERR21**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_HOST1X_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
20	0x0	**ERR20**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY3_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
19	0x0	**ERR19**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY3_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
18	0x0	**ERR18**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY2_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
17	0x0	**ERR17**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY2_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
16	0x0	**ERR16**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY1_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
15	0x0	**ERR15**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY1_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
14	0x0	**ERR14**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY0_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
13	0x0	**ERR13**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY0_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
12	0x0	**ERR12**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM5_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
11	0x0	**ERR11**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM5_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
10	0x0	**ERR10**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM4_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
9	0x0	**ERR9**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM4_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
8	0x0	**ERR8**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM3_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
7	0x0	**ERR7**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM3_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
6	0x0	**ERR6**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM2_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
5	0x0	**ERR5**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM2_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
4	0x0	**ERR4**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM1_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
3	0x0	**ERR3**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM1_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM0_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM0_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from NV_NVCSI_SAFETY_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

NVCSI_ERR_COLLATOR_ERRSLICE0_LATENTERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of LatentError_Enable register, to avoid silent dropping of errors.
Offset: 0x12c12
Byte Offset: 0x4b048
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ERR25**: 1'b1 -> Error_25_pulse[1:0] for SW Generic Error from NV_nvcsi_top.u_NV_nvcsi.u_safety was equal to 2'b00 or 2'b11. 1'b0 -> Error_25_pulse[1:0] for SW Generic Error from NV_nvcsi_top.u_NV_nvcsi.u_safety was equal to 2'b01 or 2'b10, but no latent error.
24	0x0	**ERR24**: 1'b1 -> Error_24_pulse[1:0] for Register Parity Error from NV_nvcsi_top.u_NV_nvcsicil.u_regblk.u_arreggen was equal to 2'b00 or 2'b11. 1'b0 -> Error_24_pulse[1:0] for Register Parity Error from NV_nvcsi_top.u_NV_nvcsicil.u_regblk.u_arreggen was equal to 2'b01 or 2'b10, but no latent error.
23	0x0	**ERR23**: 1'b1 -> Error_23_pulse[1:0] for Register Parity Error from NV_nvcsi_top.u_NV_nvcsi.u_nvcsi_host1x.u_regblk.u_arreggen was equal to 2'b00 or 2'b11. 1'b0 -> Error_23_pulse[1:0] for Register Parity Error from NV_nvcsi_top.u_NV_nvcsi.u_nvcsi_host1x.u_regblk.u_arreggen was equal to 2'b01 or 2'b10, but no latent error.
22	0x0	**ERR22**: 1'b1 -> Error_22_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_HOST1X_OTHER_UNCORRECTABLE_ERROR was equal to 2'b00 or 2'b11. 1'b0 -> Error_22_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_HOST1X_OTHER_UNCORRECTABLE_ERROR was equal to 2'b01 or 2'b10, but no latent error.

Bit	Reset	Description
21	0x0	**ERR21**: 1'b1 -> Error_21_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSI_HOST1X_OTHER_CORRECTABLE_ERROR was equal to 2'b00 or 2'b11. 1'b0 -> Error_21_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSI_HOST1X_OTHER_CORRECTABLE_ERROR was equal to 2'b01 or 2'b10, but no latent error.
20	0x0	**ERR20**: 1'b1 -> Error_20_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY3_OTHER_UNCORRECTABLE_ERROR was equal to 2'b00 or 2'b11. 1'b0 -> Error_20_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY3_OTHER_UNCORRECTABLE_ERROR was equal to 2'b01 or 2'b10, but no latent error.
19	0x0	**ERR19**: 1'b1 -> Error_19_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY3_OTHER_CORRECTABLE_ERROR was equal to 2'b00 or 2'b11. 1'b0 -> Error_19_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY3_OTHER_CORRECTABLE_ERROR was equal to 2'b01 or 2'b10, but no latent error.
18	0x0	**ERR18**: 1'b1 -> Error_18_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY2_OTHER_UNCORRECTABLE_ERROR was equal to 2'b00 or 2'b11. 1'b0 -> Error_18_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY2_OTHER_UNCORRECTABLE_ERROR was equal to 2'b01 or 2'b10, but no latent error.
17	0x0	**ERR17**: 1'b1 -> Error_17_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY2_OTHER_CORRECTABLE_ERROR was equal to 2'b00 or 2'b11. 1'b0 -> Error_17_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY2_OTHER_CORRECTABLE_ERROR was equal to 2'b01 or 2'b10, but no latent error.
16	0x0	**ERR16**: 1'b1 -> Error_16_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY1_OTHER_UNCORRECTABLE_ERROR was equal to 2'b00 or 2'b11. 1'b0 -> Error_16_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY1_OTHER_UNCORRECTABLE_ERROR was equal to 2'b01 or 2'b10, but no latent error.
15	0x0	**ERR15**: 1'b1 -> Error_15_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY1_OTHER_CORRECTABLE_ERROR was equal to 2'b00 or 2'b11. 1'b0 -> Error_15_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY1_OTHER_CORRECTABLE_ERROR was equal to 2'b01 or 2'b10, but no latent error.
14	0x0	**ERR14**: 1'b1 -> Error_14_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY0_OTHER_UNCORRECTABLE_ERROR was equal to 2'b00 or 2'b11. 1'b0 -> Error_14_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY0_OTHER_UNCORRECTABLE_ERROR was equal to 2'b01 or 2'b10, but no latent error.

Bit	Reset	Description
13	0x0	**ERR13**: 1'b1 -> Error_13_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY0_OTHER_CORRECTABLE_ERROR was equal to 2'b00 or 2'b11. 1'b0 -> Error_13_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY0_OTHER_CORRECTABLE_ERROR was equal to 2'b01 or 2'b10, but no latent error.
12	0x0	**ERR12**: 1'b1 -> Error_12_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM5_OTHER_UNCORRECTABLE_ERROR was equal to 2'b00 or 2'b11. 1'b0 -> Error_12_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM5_OTHER_UNCORRECTABLE_ERROR was equal to 2'b01 or 2'b10, but no latent error.
11	0x0	**ERR11**: 1'b1 -> Error_11_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM5_OTHER_CORRECTABLE_ERROR was equal to 2'b00 or 2'b11. 1'b0 -> Error_11_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM5_OTHER_CORRECTABLE_ERROR was equal to 2'b01 or 2'b10, but no latent error.
10	0x0	**ERR10**: 1'b1 -> Error_10_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM4_OTHER_UNCORRECTABLE_ERROR was equal to 2'b00 or 2'b11. 1'b0 -> Error_10_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM4_OTHER_UNCORRECTABLE_ERROR was equal to 2'b01 or 2'b10, but no latent error.
9	0x0	**ERR9**: 1'b1 -> Error_9_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM4_OTHER_CORRECTABLE_ERROR was equal to 2'b00 or 2'b11. 1'b0 -> Error_9_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM4_OTHER_CORRECTABLE_ERROR was equal to 2'b01 or 2'b10, but no latent error.
8	0x0	**ERR8**: 1'b1 -> Error_8_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM3_OTHER_UNCORRECTABLE_ERROR was equal to 2'b00 or 2'b11. 1'b0 -> Error_8_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM3_OTHER_UNCORRECTABLE_ERROR was equal to 2'b01 or 2'b10, but no latent error.
7	0x0	**ERR7**: 1'b1 -> Error_7_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM3_OTHER_CORRECTABLE_ERROR was equal to 2'b00 or 2'b11. 1'b0 -> Error_7_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM3_OTHER_CORRECTABLE_ERROR was equal to 2'b01 or 2'b10, but no latent error.
6	0x0	**ERR6**: 1'b1 -> Error_6_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM2_OTHER_UNCORRECTABLE_ERROR was equal to 2'b00 or 2'b11. 1'b0 -> Error_6_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM2_OTHER_UNCORRECTABLE_ERROR was equal to 2'b01 or 2'b10, but no latent error.

Bit	Reset	Description
5	0x0	**ERR5**: 1'b1 -> Error_5_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM2_OTHER_CORRECTABLE_ERROR was equal to 2'b00 or 2'b11. 1'b0 -> Error_5_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM2_OTHER_CORRECTABLE_ERROR was equal to 2'b01 or 2'b10, but no latent error.
4	0x0	**ERR4**: 1'b1 -> Error_4_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM1_OTHER_UNCORRECTABLE_ERROR was equal to 2'b00 or 2'b11. 1'b0 -> Error_4_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM1_OTHER_UNCORRECTABLE_ERROR was equal to 2'b01 or 2'b10, but no latent error.
3	0x0	**ERR3**: 1'b1 -> Error_3_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM1_OTHER_CORRECTABLE_ERROR was equal to 2'b00 or 2'b11. 1'b0 -> Error_3_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM1_OTHER_CORRECTABLE_ERROR was equal to 2'b01 or 2'b10, but no latent error.
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM0_OTHER_UNCORRECTABLE_ERROR was equal to 2'b00 or 2'b11. 1'b0 -> Error_2_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM0_OTHER_UNCORRECTABLE_ERROR was equal to 2'b01 or 2'b10, but no latent error.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM0_OTHER_CORRECTABLE_ERROR was equal to 2'b00 or 2'b11. 1'b0 -> Error_1_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM0_OTHER_CORRECTABLE_ERROR was equal to 2'b01 or 2'b10, but no latent error.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from NV_NVCSI_SAFETY_err_collator was equal to 2'b00 or 2'b11. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from NV_NVCSI_SAFETY_err_collator was equal to 2'b01 or 2'b10, but no latent error.

NVCSI_ERR_COLLATOR_ERRSLICE0_COUNTER_RELOAD_0

Offset: 0x12c14
Byte Offset: 0x4b050
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ERR25**: 1'b1 -> Reload Error Counter for SW Generic Error from NV_nvcsi_top.u_NV_nvcsi.u_safety 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
24	0x0	**ERR24**: 1'b1 -> Reload Error Counter for Register Parity Error from NV_nvcsi_top.u_NV_nvcsicil.u_regblk.u_arreggen 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
23	0x0	**ERR23**: 1'b1 -> Reload Error Counter for Register Parity Error from NV_nvcsi_top.u_NV_nvcsi.u_nvcsi_host1x.u_regblk.u_arreggen 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
22	0x0	**ERR22**: 1'b1 -> Reload Error Counter for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_HOST1X_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
21	0x0	**ERR21**: 1'b1 -> Reload Error Counter for Other Corrected Error from FAULT_UDF_TAG_NVCSI_HOST1X_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
20	0x0	**ERR20**: 1'b1 -> Reload Error Counter for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY3_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
19	0x0	**ERR19**: 1'b1 -> Reload Error Counter for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY3_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
18	0x0	**ERR18**: 1'b1 -> Reload Error Counter for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY2_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

Bit	Reset	Description
17	0x0	**ERR17**: 1'b1 -> Reload Error Counter for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY2_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
16	0x0	**ERR16**: 1'b1 -> Reload Error Counter for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY1_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
15	0x0	**ERR15**: 1'b1 -> Reload Error Counter for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY1_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
14	0x0	**ERR14**: 1'b1 -> Reload Error Counter for Other Uncorrected Error from FAULT_UDF_TAG_NVCSICIL_PHY0_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
13	0x0	**ERR13**: 1'b1 -> Reload Error Counter for Other Corrected Error from FAULT_UDF_TAG_NVCSICIL_PHY0_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
12	0x0	**ERR12**: 1'b1 -> Reload Error Counter for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM5_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
11	0x0	**ERR11**: 1'b1 -> Reload Error Counter for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM5_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
10	0x0	**ERR10**: 1'b1 -> Reload Error Counter for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM4_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

Bit	Reset	Description
9	0x0	**ERR9**: 1'b1 -> Reload Error Counter for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM4_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
8	0x0	**ERR8**: 1'b1 -> Reload Error Counter for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM3_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
7	0x0	**ERR7**: 1'b1 -> Reload Error Counter for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM3_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
6	0x0	**ERR6**: 1'b1 -> Reload Error Counter for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM2_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
5	0x0	**ERR5**: 1'b1 -> Reload Error Counter for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM2_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
4	0x0	**ERR4**: 1'b1 -> Reload Error Counter for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM1_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
3	0x0	**ERR3**: 1'b1 -> Reload Error Counter for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM1_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
2	0x0	**ERR2**: 1'b1 -> Reload Error Counter for Other Uncorrected Error from FAULT_UDF_TAG_NVCSI_STREAM0_OTHER_UNCORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

Bit	Reset	Description
1	0x0	**ERR1**: 1'b1 -> Reload Error Counter for Other Corrected Error from FAULT_UDF_TAG_NVCSI_STREAM0_OTHER_CORRECTABLE_ERROR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
0	0x0	**ERR0**: 1'b1 -> Reload Error Counter for Register Parity Error from NV_NVCSI_SAFETY_err_collator 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

7.2.2 Video Input (VI)

7.2.2.1 Overview

The Video Input controller is the sixth generation of this controller, and hence is referred to as VI6, or simply VI. This unit receives data from the CSI (CSI) and routes it to system memory. VI6 is designed to fit a larger number of simultaneous cameras transmitting a larger number of exposures with higher bit depths while supporting additional bandwidth and allowing the system to process it all efficiently. VI6 also introduces some safety enhancements for failure detection and diagnostic.

Similar to previous versions, VI6 is designed to be as orthogonal in its configuration as possible. VI6 still uses a single pipeline, which is time-multiplexed into a number of channels; VI6 can choose which channel each pixel packet is associated with, based on a set of packet properties, and then can choose how to process each channel individually. VI6 has tight integration of the hardware with an NVIDIA-designed Falcon real-time microcontroller, and uses Falcon software to offload work that would otherwise require specific programming options for individual features, allowing the capabilities to be expanded.

7.2.2.1.1 Pixel Conveyances

There are many different stages in which pixels are conveyed; the terminology for each of these is specific, and sometimes not obvious. The following terms describe various types of pixel conveyances.

Table 7.14 Pixel Conveyance Terms

Term	Definition
Lane	A single 2-wire (D-PHY) or 3-wire (C-PHY) interface used for high-speed serial clock or data transmission over the MIPI CSI link.
Link	An aggregation of one, two, three (C-PHY only), or four MIPI CSI lanes from a common data source that conveys pixels. Link is sometimes also used to refer to an aggregation of two smaller links (i.e., an x4 link bonded with another x4 link is referred to as an x8 link, even though both smaller links independently follow the MIPI CSI protocol, and convey some pixels on their own).
Brick	One instantiation of NVCSI; the hardware-managing lanes and links that are used. The D-PHY/C-PHY bricks have four lanes that can individually be configured into an x4 stream, an x2/x2 stream pair, an x2/x1 stream pair, or an x1/x1 stream pair. There are two D-PHY/C-PHY bricks that can use both streams in a pair, and two D-PHY/C-PHY bricks that have only one stream connected.
Stream	One output interface from an NVCSI brick. There are up to two streams per D-PHY/C-PHY brick. Conveys pixel packets. A stream can also be configured as the aggregation of two smaller streams, through the CSIMUX de-interleaver.
Pixel packet	The quantum of information flowing through the VI pipeline.
Atom	The quantum of information flowing out of VI into the memory interface.
Virtual channel	A four-bit field in the MIPI CSI packet header, for multiple time-multiplexed (line-interleaved) virtual links over one physical link. Each virtual channel has a frame start and frame end bracketing each frame sent. All packets arrive by way of a numbered virtual channel, even if only one virtual channel is in use on the link.
Data-type	A six-bit field in the MIPI CSI packet header determining what data format is being sent in any given packet. Multiple data-types can be interleaved in a single frame.
Channel	A virtual pipeline in VI. There are two VI instances and each VI instance supports up to 36 channels.
Surface	A location in memory and set of resources needed for sending an independent stream of data. VI6 has three surfaces per channel.

7.2.2.1.2 Dynamic Frequency Scaling

VI6 supports Dynamic Frequency Scaling: the VI clock can be changed while VI is processing data traffic, as long as it meets bandwidth requirement for input CSI streams.

7.2.2.1.3 Performance

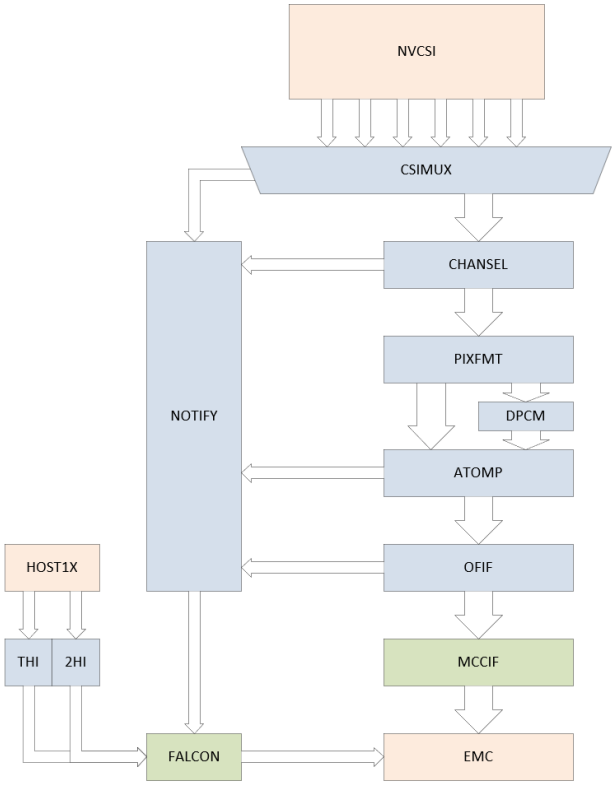
NVCSI supports up to 4.5 Gsym/s C-PHY link speed, therefore with 16 trios the maximum pixel rate is 10.3 Gpix/s in 16bpp case. Some variants can have two instances of VI to support high input bandwidth requirements.

7.2.2.1.4 Architectural Overview

VI6 receives one packet at a time from an NVCSI source and passes it through a fixed pipeline before dispatching it to memory. The pipeline is channel-multiplexed with each stage maintaining a cache of its required state for each channel. VI6 additionally has a generic notification log

mechanism, which can log fixed-sized messages to a built-in buffer for the host application to parse at its convenience. A high-level block diagram for VI6 is shown below.

Figure 7.10 VI6 Top-Level Block Diagram



Each block of the VI6 top-level block diagram is described:

- CSI Multiplexer (CSIMUX) - The CSIMUX functions as the frontend to the VI pipeline, serving as a transport between the NVCSI inputs and VI itself. Since VI can only handle a pixel packet from one CSI stream at a time, CSIMUX contains a small buffer for each input to handle peak bandwidth conditions. If a source camera does not provide a frame ID, CSIMUX can emit an auto-incrementing frame ID per virtual channel.
  - The external interfaces have the capability of producing a peak of one packet every cycle; each packet consists of either four or eight pixels. (This is also the internal packet width within VI6.) Each input is synchronized down into the VI clock domain through an asynchronous FIFO, and is prioritized and multiplexed by an arbiter and multiplexer.
- Channel Selector (CHANSEL) - Since the VI6 pipeline is fixed, each stage is time-multiplexed into channels; every pixel packet is assigned to a set of channels as it flows through the pipe. (A pixel packet can be reasonably assigned to zero channels, if it matches nothing; one channel, if it contains pixel data; or many channels, if it contains an NVCSI FS/FE event.) CHANSEL has the task of inspecting each incoming pixel packet and pattern-matching it against each channel's configuration.

  - CHANSEL is capable of matching based on any combination of physical channel (i.e., which CSI channel), data-type, virtual channel ID, and frame ID.
  - CHANSEL also maintains the per-frame state for each channel and emits notifications to NOTIFY for start-of-frame, specific-line-completion, and end-of-frame events. Embedded data is also routed appropriately in CHANSEL.
- Pixel Formatter (PIXFMT) - CSI2 provides a very minimal set of pixel formats through the bus interface. VI6's PIXFMT module unpacks the input pixel formats into formats that are advantageous for the host system to read, and formats that ISP can handle natively. (Much of the behavior of the pixel formatter is retained from prior versions of VI.)
- DPCM Encoder - To save memory bandwidth, VI6 provides a compression mechanism for data traveling between VI and memory. Compression is implemented using a two-pixel differential PCM encoding, wrapped in a Huffman code to get bit-rate reduction. The DPCM encoder stage packs the (odd-width) bitstream into a sequence of bytes and emits whole byte quantities to ATOMP to align into atoms.
  - For strip-mode operation, the DPCM encoder needs to periodically insert encoding restart points; otherwise, the decoder is not capable of recovering enough state to start in the middle of a line. Since this form of compression exists to save memory bandwidth, not memory quantity, restart points are realigned to known boundaries; the result is (in the best case) an extremely sparse memory image.
  - DPCM is available for RAW10, RAW12, logarithmic RAW16, RAW20, and run-length-encoded RAW12 data; all other formats (or cases in which DPCM is disabled) use a bypass path that leaves the pixel data unmodified, but kept in lock-step with the DPCM pipeline.
- Memory Atom Packer (ATOMP) - ATOMP is the final stage of the VI6 alignment pipeline; it takes pixel packets of varying widths and produces memory requests. ATOMP is responsible for packing pixels into appropriate transaction sizes (memory atoms), generating memory addresses, and managing write acknowledgments. When write acknowledgments return for requested end-of-line or end-of-frame events, ATOMP returns them to NOTIFY to be timestamped and output.
- Output Water-level Buffer (OFIF) - Atoms would ordinarily be sent directly to MCCIF for output to memory, but MCCIF is incapable of providing an early warning that it is running low on buffer capacity. To provide back pressure within the VI6 pipeline, a buffer capable of providing a high-water-level indication is necessary. When the buffer 'water level' reaches a programmable level, notifications are issued to NOTIFY, and ATOMP begins dropping pixel packets from channels programmed to be less critical.
- Notification Engine (NOTIFY) - NOTIFY serves to receive messages from throughout VI6, and to enqueue them for usage by Falcon. NOTIFY contains FIFOs to buffer events from each source and one master output FIFO to buffer events; these events are later read out from a CPU and dispatched appropriately.
  - NOTIFY receives messages from CSI (CSI beginning-of-line and end-of-line messages); CHANSEL (start-of-frame, start-of-line, unrecognized channel, frame dropped, incorrect size); ATOMP (end-of-frame, end of line); OFIF (packet dropped, buffer full); and the VGPIO block (VGPIO state changed). The messages are timestamped using the system timestamp counter upon arrival to allow synchronization to the rest of the system. This timestamp counter is also synchronized with the Falcon microcontroller.

- Falcon microcontroller - VI6 abstracts its programming interface through a Falcon microcontroller. The Falcon has individual state for each channel in the VI system, and has a work list for each channel in the event of failure or successful completion. Since the Falcon's latency is small and easily bounded, it can guarantee notification read performance, effectively closing the loop from frame end to reprogramming.
- Miscellaneous hardware - VI6 also contains: a GPIO block, to interface with flashes, mechanical shutters, and user buttons; a PWM unit, to drive sensor I/Os on a periodic time basis; and a timestamp unit, tracking system time.

7.2.2.2 Functional Description

The previous section presented an overview of the VI6 architecture; in this section, we provide a detailed description of the behavior of VI6. The detailed functional description proceeds by block with each section containing a description of the block's operation and interface to software.

7.2.2.2.1 CSIMUX

CSIMUX's core is a multiplexer that takes multiple inputs from the NVCSI clock domain and synchronizes one at a time to the VI6 internal clock. Internally, it synchronizes through asynchronous FIFOs. It outputs one-pixel packet every cycle, and no backpressure is permitted throughout the VI6 pipeline. CSIMUX is also the point at which some 8-lane sensors are de-interleaved.

There is one VI_CSIMUX_CONFIG_STREAM register set for each NVCSI stream on the system.

CSIMUX maintains a 16-bit frame ID counter for each virtual channel on each NVCSI stream. It is incremented with each start of frame from that virtual channel; if the FRAMEIDGEN bit for that virtual channel is set in VI_CSIMUX_STREAM_CONFIG for a stream, then the generated frame ID replaces the value from NVCSI. (Note that the initial value of the frame counter is, surprisingly, 1 and not 0!) The frame ID generator's counter is reset whenever FRAMEIDGEN is disabled. The value range of the generated frame ID is [1, 32768], inclusively. The value range works fine with any 15-bit frame ID mask.

The VPR bit in VI_CSIMUX_STREAM_CONFIG indicates whether output must be written to VPR RAM. (Specifically, when a packet with the VPR bit set reaches ATOMP, the AXI bus signal awuser[10] shall be set.) The VPR bit in the register for a brick comes from a sticky bit in the PMC specific to that NVCSI brick. This allows for VPR support that is permanently bound to an NVCSI brick and can survive VI reset. The 8-lane de-interleaver's VPR bit comes from the OR of the VPR bits of the bricks it is currently attached to.

Error Handling and Recovery

CSIMUX has a multitude of events that it can detect, both in 8-lane de-interleave mode and in normal operation modes. Loosely, these are grouped into three types: stream events, frame faults, and frame status.

Frame status events are events that can happen during normal operation. In the case of normal streams, the events available are FE and FS, which reflect FS and FE events on a virtual channel from NVCSI. If a fault occurs at the same time as an FS or FE event, then the tag changes to CSIMUX_FRAME, and a frame error is reported instead. If a fault occurs during a frame, the first fault is immediately reported, and any additional faults are deferred until frame end; the FE notification, in either such case, is changed to a CSIMUX_FRAME and accumulated errors are reported.

Stream events are events that are error conditions; they should not occur in normal operation. In the case of normal streams, the events available are STREAM_BADPKT, which indicates that an illegal packet has been sent from NVCSI; STREAM_OVERFLOW indicates that the FIFO for this stream has overflowed; STREAM_LOF indicates that a frame start packet has been lost because of a FIFO overflow; or STREAM_SPURIOUS, which reports that the next packet after a frame end was not a frame start.

In the 8-lane modes, these events are DEINT_{LO,HI}*BADPKT, DEINT*{LO,HI}*OVERFLOW, DEINT*{LO,HI}*LOF, and DEINT*{LO,HI}_SPURIOUS. Stream events can occur on multiple streams at once and are reported in parallel through the CSIMUX_STREAM notification tag.

Frame fault events are events that are error conditions and occur because there is an issue with the data that came from the sensor. They are as follows (listed as normal / 8-lane):

- FE_CSI_FAULT/DEINT_FE_CSI_{HI, LO}_FAULT: NVCSI has reported an error in the received frame. Error bits from NVCSI are individually masked with the CSI_STREAM.FAULT_MASK registers; one mask register exists for each stream in the system.
- DEINT_DATA_FAULT: A general data format exception occurred while deinterleaving. A subcode is given in the DATA_FAULT_CODE field of the payload; errors that can occur are mismatched or unexpected line-starts, mismatched or unexpected line-ends, minimum line width violations, or line size violations (for instance, having insufficient pixels to fill a de-interleaved packet).
- FS_FAULT / DEINT_FS_FAULT: A frame start was found for a virtual channel before the previous frame was concluded with a frame end. In such an event, CSIMUX additionally injects a frame end packet; the FS_FAULT is reported synchronously to the injected FE packet.
- DEINT_FE_FAULT: A frame end arrived in the low stream, but there was no frame end nearby in the high stream.
- FORCE_FE_FAULT/DEINT_FORCE_FE_FAULT: A frame end was injected either by the FEINJECT mechanism or by the frame timer.
- DEINT_FS_FRAME_ID_FAULT: The frame IDs do not match between the high and low streams.

- FE_FRAME_ID_FAULT/DEINT_FE_FRAME_ID_FAULT: The frame ID from the FE packet does not match the frame ID from the FS packet.
- DEINT_DTYPE_FAULT: The data-types do not match between the high and low streams.
- DEINT_VC_FAULT: The virtual channels do not match between the high and low streams.
- PXL_ENABLE_FAULT / DEINT_{HI, LO}_PXL_ENABLE_FAULT: An illegal pixel encoding has been detected (for example, a non-full pixel packet without an end-of-frame identifier).
- DEINT_PXL_ENABLE_FAULT: The pixel packet encoding after merging would result in an illegal pixel set of pixel enables.
- SPILL_FAULT: The low stream in a de-interleaved mode has more pixels than the high stream, or there is an unexpected packet skew between the line end of the HI and LO streams.
- All events are routed to notifications when they occur, but can be overridden by the mask registers VI_CSIMUX_NOTIFY_MASK_STREAM[...] and VI_CSIMUX_EIGHT_LANE_DEINT_NOTIFY_MASK. Additionally, when a frame fault event or a stream event occurs, the STICKYFAULT bit is set in VI_CSIMUX_CONFIG_STREAM.

The case in which a sensor fails to send a frame-end could potentially cause the downstream pipeline to consume resources that it may not be able to release until the frame-end arrives. To combat this, CSIMUX provides a mechanism to either manually inject a frame-end event, or automatically to inject frame-end events based on a timeout. The FEMAXTIME and FESHORTTIMER registers in _CONFIG_STREAM configure the behavior of this timeout; if FEMAXTIME is set to 0, the timeout is disabled. (FESHORTTIMER is provided to reduce the timeout for more expeditious verification.)

A mechanism to force a frame end is also available through the FORCE_FE trigger bit. (FORCE_FE does not generate a frame end if there is not currently a matching frame start.) FORCE_FE should be used whenever a stream is reset, or if a stream appears to be hung (and the timeout is not desired); additionally, FORCE_FE can be used as part of hardware testing to inject short-frame errors.

To reset a stream for any reason, use this specific sequence:

1. Set CONFIG_STREAM.QBLOCK. This disables the input FIFO and flushes any pending data from it, ensuring that an overflow does not occur while the reset process is taking place, and ensuring that the stream resumes operation from a known state if the sensor is disabled.
2. Trigger CONFIG_STREAM.FEINJECT. This terminates any frames that are in flight and marks them with an error.
3. Trigger CONFIG_STREAM.SRESET. This resets any overflow detection and spurious data state.
4. For each channel linked to the stream, perform a channel reset on CHANSEL and ISPBUF.
5. Clear CONFIG_STREAM.QBLOCK. This enables the stream for normal operation once again.

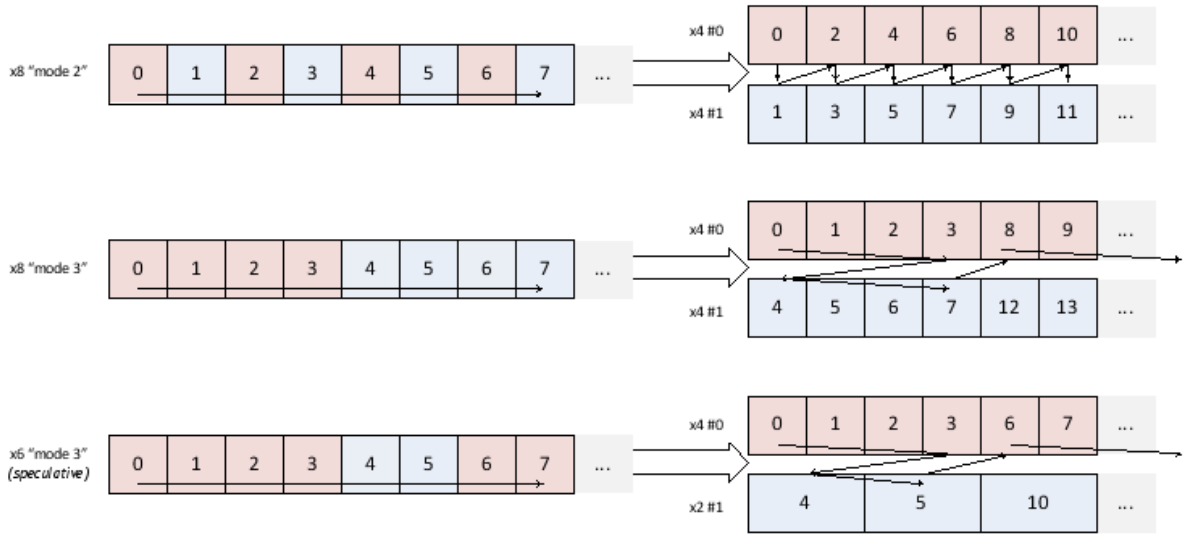
Not all error conditions mandate a stream reset; the majority of these events are recoverable by the hardware. It is recommended to reset a stream in the event of a LOF error; however, in such an event it is not possible to guarantee how many frame starts have been lost, and the frame starts may have been lost from multiple virtual channels.

Notifications

CSIMUX is capable of emitting four tags of NOTIFY events. In normal operation, CSIMUX emits an FS and an FE event. In error conditions, CSIMUX also emits CSIMUX_FRAME and CSIMUX_STREAM events.

The FS, FE, and CSIMUX_FRAME events are similar in nature, and have their payloads formatted the same way. The channel number is set to the concatenation of the NVCSI data-type with the NVCSI virtual channel number, and the payload contains the stream ID and any appropriate errors, as described preceding.

Figure 7.11 8-Lane Modes Supported in VI



Eight-Lane Deinterleaving

Some sensors support a special 8-lane mode that spreads pixels across two 4-lane CSI links. CSIMUX supports deinterleaving these 8-lane sensors, however it imposes some limitations on the conditions in which this is possible. Specifically, for CSIMUX to de-interleave 8-lane modes:

- Both CSI links must come from adjacent bricks: link pairs cannot be arbitrarily matched up.
- Both CSI links must be on channel A of their respective bricks.
- Virtual channels must not be used because there must be only one virtual channel on each link.
- The sensor must be running in a mode that outputs as eight PPC from CSI–that is to say, a RAW mode, rather than an RGB or YUV mode.
- Lines must be at least nine pixels wide. (Mode 2, in x8 mode, supports lines as short as three pixels wide.)

The de-interleaver fits in after the FIFOs for both links, but before arbitration. When the de-interleaver is activated (by setting EIGHT_LANE_DEINT_CONFIG.MODE to a setting other than DISABLE), the de-interleaver takes the place of _both streams from the master brick, and the primary stream from the slave brick. (That is, if MUXSEL is set to PAIR_MS_02, then streams 0, 1, and 2 are deactivated.)

Deactivated streams have their weight set to zero, and the de-interleaver is activated on the arbiter with the weight specified in _EIGHT_LANE_DEINT_CONFIG.WT. The FIFOs for both links, in deinterleaving mode, are no longer controlled by the arbitration logic.

Input Swizzling

To facilitate debugging of the de-interleave logic, and to support use cases that require a single input to be routed to multiple channels simultaneously, CSIMUX provides an input swizzling circuit. The input swizzling allows any stream output into CSIMUX to take its input from any physical stream. On a micro-architectural level, the input swizzler exists in the NVCSI clock domain, before any processing (including deinterleaving). The swizzler is disabled by hardware and reset to an identity mapping if any VI6 security bits (CSI[AB|CD|EF|GH]_SECURE_ENABLE or CHANNEL_SECURE_ENABLE) are activated.

Table 7.15 Swizzler Registers

Register	Field	Bits	Reset	Description
VI_CSIMUX_ STREAM_SWIZZLER	STREAM5_SWIZZLE_SRC	22:20	5	Physical stream number for output stream 5.
	STREAM4_SWIZZLE_SRC	18:16	4	Physical stream number for output stream 4.
	STREAM3_SWIZZLE_SRC	14:12	3	Physical stream number for output stream 3.
	STREAM2_SWIZZLE_SRC	10:8	2	Physical stream number for output stream 2.
	STREAM1_SWIZZLE_SRC	6:4	1	Physical stream number for output stream 1.
	STREAM0_SWIZZLE_SRC	2:0	0	Physical stream number for output stream 0.

Table 7.16 CSIMUX Input Select Registers

Register	Field	Bits	Reset	Description
VI_CSIMUX_ INPUT_STREAM_MUX	STREAM0	1:0	0	0: NVCSI stream 0 1: TERMINATE
	STREAM1	3:2	0	0: NVCSI stream 1 1: TERMINATE
	STREAM2	5:4	0	0: NVCSI stream 2 1: TERMINATE
	STREAM3	7:6	0	0: NVCSI stream 3 1: TERMINATE

Register	Field	Bits	Reset	Description
	STREAM4	9:8	0	0: NVCSI stream 4 1: TERMINATE
	STREAM5	11:10	0	0: NVCSI stream 5 1: TERMINATE

Host1x Injection Path

To support validation of VI, it may be valuable to run a software-defined test pattern generator. CSIMUX enables this with the addition of an 'injection' path from Host1x, in which software can synthesize CSI-to-VI packets and inject them into VI as if they came directly from a CSI stream. Such packets are injected as close to the top of the VI pipeline as possible, allowing even de-interleaver logic to be tested. The injected packet source corresponding CSI[AB|CD|EF|GH]_SECURE_ENABLE as VPR flag.

The injection pathway is designed to be able to synthesize any logical pattern of CSI packets down any sequence of streams, which can be used to test functionality that current sensors are unable to produce. The injection pathway can also synthesize error cases, enabling the validation of both software and hardware error handling. The constraint that the Host1x interface presents is that throughput through this interface is very limited; for that reason, testing timing-driven test cases can be validated only in limited circumstances using the injection path. (Use of the switch, accumulate, and stall options can allow certain small back-to-back sequences to be injected.)

The injection interface is a set of staging registers that mirror the CSI-to-VI packet specification, along with an injection trigger register that sends a packet down one or more streams. This register set is specified below.

Table 7.17 CSIMUX Packet Injection Registers

Register	Field	Bits	Reset	Description
VI_CSIMUX_INJECT_DATA_0	PPC4_PXL0	31:0	0	Pixel 0 for 4 PPC mode.
	PPC8_RAWPXL0	15:0	0	Pixel 0 for 8 PPC mode.
	PPC8_RAWPXL1	31:16	0	Pixel 1 for 8 PPC mode.
	FRAME_ID	15:0	0	Frame ID.
VI_CSIMUX_INJECT_DATA_1	PPC4_PXL1	31:0	0	Pixel 1 for 4 PPC mode.
	PPC8_RAWPXL2	15:0	0	Pixel 2 for 8 PPC mode.

Register	Field	Bits	Reset	Description
	PPC8_RAW PXL3	31:16	0	Pixel 3 for 8 PPC mode.
VI_CSIMUX_INJECT_DATA_2	PPC4_PXL2	31:0	0	Pixel 2 for 4 PPC mode.
	PPC8_RAW PXL4	15:0	0	Pixel 4 for 8 PPC mode.
	PPC8_RAW PXL5	31:16	0	Pixel 5 for 8 PPC mode.
VI_CSIMUX_INJECT_DATA_3	PPC4_PXL3	31:0	0	Pixel 3 for 4 PPC mode.
	PPC8_RAW PXL6	15:0	0	Pixel 6 for 8 PPC mode.
	PPC8_RAW PXL7	31:16	0	Pixel 7 for 8 PPC mode.
VI_CSIMUX_INJECT_HEADER	PXL_ENABLE	7:0	0	Valid pixels for data packets.
	DTYPE	13:8	0	CSI data-type.
	FE_ERROR	13:8	0	Errors reported at FE.
	VC	17:14	0	Virtual channel number.
	CTYPE	21:18	0	Packet type.
	PPC_FLAG	22:22	0	Pixels per cycle, or pixels per packet. 1 for 8 PPC, 0 for 4 PPC.
VI_CSIMUX_INJECT	STREAM	6	*(trigger)*	Many-hot set of streams on which to inject the packet loaded into the staging registers.
VI_CSIMUX_INJECT_CFG	STREAM_SWITCH	6	0	Multiplexer enable per stream: set a bit to 0 to take input from CSI, or to 1 to take input from the Host1x injection stream.
	ACCUMULATE	1	0	When set, accumulates packets from Host1x injection, to be released in batch when unset.
	PIPE_STALL	1	0	When set, disables CSIMUX pipeline to test overflow functionality. *(debug feature)*
	FIFO_RD_COUNT	6	0	Number of packets in injection FIFO waiting to be consumed.
	FIFO_WR_COUNT	6	0	As FIFO_RD_COUNT, but from write side of FIFO. Use the larger of the two.
	FIFO_DEPTH	6	32 (R/O)	Size of injection FIFO, in number of packets available to be held.

PPC Calculation

VI6 adds PPC information to the NVCSI-to-VI packet. CSIMUX and CHANSEL can use this info to properly handle user-defined data-types. The register, PPC_CALCULATION, provides for backward compatibility with prior VI programming model. When USE_NVCSI2VI_BUS is 0, the VI pipeline's internal pixel count representation is derived from the reported data-type. However, when USE_NVCSI2VI_BUS is set to 1, PPC info from the NVCSI-to-VI packet is used. The default value of USE_NVCSI2VI_BUS is 1; this is the standard software configuration, unless the PPC info from NVCSI results in unexpected behavior from VI. Note that this register applies to all CSIMUX streams and VI channels.

Table 7.18 PPC Calculation Register

Register	Field	Bits	Reset	Description
PPC_CALCULATION	USE_NVCSI2VI_BUS	0:0	1	0: PPC derived from data-type 1: PPC from NVCSI-to-VI packet

7.2.2.2.2 CHANSEL

CHANSEL is a pattern-matching engine wired up to a priority encoder. Each channel also has a small amount of configuration state associated with it.

The matching engine is capable of matching on: input stream ID, virtual channel ID, data-type, and frame ID. (Frame ID is specified as a field and mask to permit matching on every nth frame, for *n* as a power of 2.) As described previously, CHANSEL can match any number of channels on an input packet from NVCSI. The rules for matching are as follows:

- An NVCSI packet that is an FS or FE matches all channels that have a matching input stream ID, virtual channel ID, and frame ID. The data-type match is ignored for FS or FE packets. The packet is then broadcast down the pipeline as an NVCSI FS/FE packet.
- An NVCSI packet that contains pixels with an embedded data data-type matches the first channel that has a matching input stream ID, virtual channel ID, and frame ID. The data-type match is ignored for embedded data packets. (For clarity, pixel data or embedded data can only be transmitted down one VI channel at a time: if a packet matches multiple channels, then the lowest-numbered channel takes precedence.) Embedded data packets use the embedded data line counter to determine when ATOMP shall send an "embedded data complete" notification.
- An NVCSI packet that contains pixels with any other data-type matches the first channel that has a matching input stream ID, virtual channel ID, frame ID, and data-type. Data packets use the height counter to determine when an EOF is generated in the pipeline; their widths are also validated, and an optional number of pixel packets are skipped at the beginning of each line.
- If the first nonembedded data packet after an NVCSI FS packet from an input stream does not match any channels at all, a "no-match" notification is emitted from CHANSEL.

The configuration state contains a width and height to permit CHANSEL to generate notifications for and discard frames that have incorrect sizes, and to set control registers within the pipeline at the beginning and end of lines and frames. CHANSEL also has counters for embedded data and not only pixel data.

DOL Header Match Mode

Some multiexposure sensors prepend a four-pixel header to each line with a "line information" word describing which lines belong to which exposures. To permit the remainder of the VI pipeline to appropriately route data based on these exposures, CHANSEL has a DOL header match mode. At the beginning of each line a selectable pixel from the first pixel packet is latched and is referred to for the rest of the line; this pixel is the DOL header. All 16 bits of the first pixel are stored and matched on, but the DOL header often appears as follows:

- Bit 4 of the first pixel in the line is a frame number identifier (FSET).
- Bits 3 through 0 of the first pixel in the line are a one-hot encoding of which exposure the line represents (FID3-FID0).

The user should note that in this mode, output data still includes the line information pixels that are cropped at ISP as necessary.

Note: DOL header mode is defined only for RAW6 through RAW14 modes.

Notifications

CHANSEL emits a handful of notifications. The normal notifications are start-of-frame and end-of-frame (as well as embedded data start-of-frame and end-of-frame), and a line timer notification. CHANSEL also emits two classes of error notifications: channel-specific faults (i.e., faults that apply to exactly one channel, and happen because of data that have been received in the channel), and many-channel faults (i.e., faults that can apply to zero, one, or many channels). The notifications are as described below:

- PXL_SOF / PXL_EOF / EMBED_SOF / EMBED_EOF (normal) are emitted under their eponymous conditions.
- NLINES (normal) is emitted when the LINE_TIMER expires.
- FAULT / FAULT_FE (one-channel fault) is a master bitmask notification emitted when a data-specific fault occurs on a channel. (The _FE version is emitted when CHANSEL is forced to insert an FE packet to terminate a frame; this can happen, for instance, during a channel reset.)
- NOMATCH (many-channel) is emitted because of the "no-match" condition described preceding.
- COLLISION (many-channel) is emitted when a start-of-frame matches a channel that is already in frame. "Stale frame" errors also appear through a COLLISION notification, but in a separate payload field; they occur for frames that did not receive a LOAD command before an FS packet arrived.

- SHORT_FRAME (many-channel) is emitted when a Frame End appears from NVCSI before the normal number of pixels has appeared (or when zero pixels have appeared, and therefore no channel could be selected). An additional payload field indicates that too few embedded lines have been received.
- LOAD_FRAMED (many-channel) is emitted when a LOAD command is received for a channel while that channel is currently in a frame.

The FAULT and FAULT_FE notifications deserve slightly more explanation. The faults that CHANSEL can detect are:

- {PIXEL,EMBED}_MISSING_LE: Two line-start packets were detected without a line-end packet between them.
- {PIXEL,EMBED}_RUNAWAY: More lines were received than expected.
- {PIXEL,EMBED}_SPURIOUS: A pixel data packet was received without a line-start packet first.
- {PIXEL,EMBED}_LONG_LINE: More pixels (or embedded data bytes) were received than expected in a line.
- PIXEL_SHORT_LINE: Fewer pixels were received than expected in a line.
- EMBED_INFRINGE: Embedded data was received on a frame for which no embedded data was expected. Note that in such a case, the embedded data is still sent to downstream VI units.
- DTYPE_MISMATCH: The pixel data-type changed in the middle of the line (or there is another data-type error).
- FORCE_FE: CHANSEL needed to force a frame to end for some reason.

Programming Notes

Note that both FRAME* and CROP* registers exist; but pixels are no longer transmitted after CROP, but the actual frame size is compared against FRAME_X.WIDTH. Therefore, if the line from the sensor is longer or shorter than _FRAME_X.WIDTH, an error is generated; if the line width is exactly _FRAME_X.WIDTH, a line of _CROP_X.WIDTH is output downstream. (This allows support for sensors that generate CSI padding pixels.) Note also that _CROP_X is specified in pixels, although _SKIP_X is specified in packets. If _CROP is longer than the image size, this is not an error; however, the appropriate sizes must be programmed into OUT* to ensure consistent behavior.

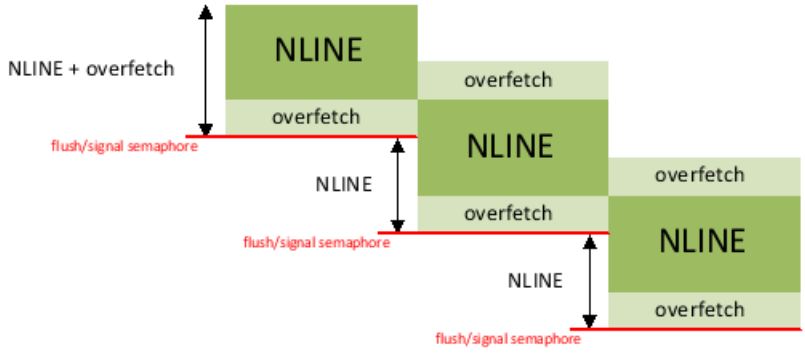
Note that _EMBED_Y.EXPECT does not control whether data is transmitted downstream, although _EMBED_Y.LINES and _EMBED_Y.ENABLE do. This means that even if EXPECT is set to 0, it is still possible that an embedded data line can travel downstream to ATOMP, which proceeds to write it out, if ENABLE is set. If ENABLE is unset, but EXPECT is set, then notifications are still generated for missing embedded data, but no atoms are generated for PIXFMT or ATOMP.

The _FLUSH and _LINE_TIMER registers are very similar; they differ in their behavior when the timer is triggered. A line timer event results only in a notification, intended for software to timestamp a specific line, and react. This may be valuable in triggering an external flash or focuser at a particular point in the image scanout. By comparison, a flush event sends a request to ATOMP to flush out all lines that have accumulated until this point, and to provide a write acknowledgment when they have been written to memory; this allows "early start" processing on a GPU (or other external

chunking-capable engine). The flush event is restricted to lines within the crop region, whereas the line timer is usable outside of the crop. (Lines are numbered from 1 in these registers.)

To better support subframe processing, the _FLUSH_FIRST and _LINE_TIMER_FIRST registers are introduced. These exist because subframe processing requires overfetch regions, and so the first sub-frame must wait for more data to be written than the following subframes.

Figure 7.12 FLUSH/LINE_TIMER



Finally, there are some additional considerations when using interleaving multiple data-types on a single channel. Data-type interleaving is not permitted for data-types that toggle between eight PPC and four PPC lines, such as the YUV420 10-bit modes. The data-type override should not be programmed to a user data-type, since a default path in PIXFMT does not exist to write those data-types out. When data-type interleaving is in use, all data-types have the same line width; if they do not, the maximum line width is programmed into _FRAME_X.WIDTH, and PIXEL_SHORT_LINE faults are masked off. Lastly, the _FRAME_Y.HEIGHT register is programmed to the total number of lines expected across all data-types sent to a channel.

7.2.2.2.3 Channel Programming Model

VI provides a double-buffered per-channel control scheme as its primary programming model. Each channel in VI has two sets of registers exposed–an active set and a shadow set. All registers that are marked as *(paged)* exist twice per channel; the register VI_CHANSEL_PARAM_SIZE exists internally as VI_CHANSEL_PARAM_SIZE_0, VI_CHANSEL_PARAM_SIZE_0_NS, VI_CHANSEL_PARAM_SIZE_1, VI_CHANSEL_ PARAM_SIZE_1_NS, and so on. Each channel is configured through a control register, VI_CHANNEL_CONTROL, which is programmed as described in the table below. For debugging purposes, the CHANNEL_COMMAND interface presents the RD_MUX_SEL and WR_ACT_SEL registers that allow direct access to the active register set; these registers should never be enabled in normal operation.

Table 7.19 Channel Control Register

Register	Field	Bits	Reset	Description
VI_CH*n ENABLE _(per-channel, but does not have shadow copy).*	ENABLE	1	1'b0	Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are alternate-surface atoms remaining to be written in ATOMP, they are still written even if the channel is disabled.
VI_CH*n CONTROL _(per-channel, but does not have shadow copy).*	SINGLESHOT	1	1'b0	1: if this channel should automatically set ENABLE to 0 when a frame completes at CHANSEL.
	SINGLESHOT_AUTO	1	1'b0	1: if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
	EARLY_ABORT	1	1'b0	1: if a LOAD should abort a frame that is in progress; 0 if the LOAD should gate on FE.
	POST_RUNAWAY_PIXEL	1	1'b1	1: if a RUNAWAY error should be flagged even if other errors are detected in a frame. (It may be advisable to disable this bit if excessive NOTIFY pressure is expected; it is safe to leave it enabled as a default.)
	POST_RUNAWAY_EMBED	1	1'b1	As POST_RUNAWAY_PIXEL, but for embedded data.
VI_CH*n CHANNEL_COMMAND _(per-channel, but does not have shadow copy).*	LOAD	1	RAZ	Atomically loads all channel state from shadow registers to active registers. (In hardware, LOAD is deferred to the end of the frame, if a frame is active.) Write 1 to load; always reads as zero.
	AUTOLOAD	1	RAZ	1: if this channel should automatically load a new configuration (as if LOAD were set) at the time that an FE is received at the last pipeline stage that reads from configuration registers, after a frame is complete.
	RD_MUX_SEL	1	0	0: host reads active register set. 1: host reads shadow register set. For debug use only.
	WR_ACT_SEL	1	0	0: host writes shadow register set. 1: host writes both shadow and active register sets. For debug use only.

The AUTOLOAD bit behaves as described, but should be used with caution. Since the autoload functionality has no provision for error detection and handling, it is generally not the preferred mechanism with which to build a robust system. The primary expected use case for AUTOLOAD is in conditions that require extremely low Vblank times, in which it becomes impossible even for a

Falcon to reprogram a channel between frames. Additionally, when a channel is in frame and enabled, LOAD commands are deferred until the frame completes; this can also be used as an alternative to autoload mode.

The SINGLESHOT bit should be set in most use cases. Continuous retriggering of a VI frame, if reprogramming does not complete in time, could result in corrupting an already-completed frame (even though a STALE_FRAME error is generated). As such, STALE_FRAME errors should never be masked in continuous mode since they always indicate that a race between configuration and frame start has occurred.

Using SINGLESHOT_AUTO and clear SINGLESHOT after shadow registers are programmed is the preferred programming model in continuous capture mode.

Initial Programming and Channel Reset

In the ideal case, a VI channel should be enabled before a sensor begins streaming into VI. Failing to do this could result in partial frames being transferred from VI, with a NOMATCH notification from the first pixel after a missing frame-start. In cases in which VI must be started after the sensor has been enabled, the first frame can be treated as an error frame, and the system can begin programming the next frame in the vertical-blanking interval (as determined by the CSIMUX FE event) as usual.

The combination of disabling, reloading, and then reenabling a channel functions as a reset for a stuck channel. Transitioning a channel from disabled to enabled is defined as resetting all channel state that is not contained in configuration registers. To reset a channel, follow the following procedure:

1. Disable the channel by clearing CONTROL.ENABLE.
2. Issue a LOAD command to the channel. The LOAD operation forces an FE to be sent down the channel if a frame was open at the time that the channel was disabled.
3. If the channel was configured to feed into FMLite, reset FMLite by asserting VI_FMLITE_VFM_CTRL.TDS_SW_RESET for at least 100 VI clock cycles.
4. If appropriate, reconfigure the channel for a new programming, and issue another LOAD command to the channel.
5. If appropriate, reenable the channel by setting CONTROL.ENABLE.

7.2.2.2.4 PIXFMT

CSI2 provides a very minimal set of pixel formats through the bus interface. VI's PIXFMT module unpacks the input NVCSI pixel formats into formats that are advantageous for the host system to read. The input to PIXFMT is formatted exactly as NVCSI presents it to VI. PIXFMT contains logic to reformat these pixels into the memory output format in preparation for ATOMP, and as such, the output format is designed to be conducive for easy atom packing.

Input Data Formats

NVCSI provides input to VI in one of several formats. NVCSI can transmit the following pixel formats with the following bus widths:

- RAW 8ppc: RAW6, RAW7, RAW8, RAW10, RAW12, RAW14, RAW16, embedded-8
- RAW 4ppc: RAW20
- RGB 8ppc: RGB444, RGB555, RGB565
- RGB 4ppc: RGB666, RGB888
- YUV 8ppc: YUV420-8 legacy, YUV420-8 CSPS, YUV422-8
- YUV 4ppc: YUV422-10
- YUV mixed: YUV420-10 CSPS

Merged RAW Data Input

HDR applications can present linearized wide data in formats that are too wide to transmit over CSI in any native fashion. For instance, in addition to the native CSI RAW16 format, RAW16 can be implemented by merging adjacent RAW8 pixels, and RAW20 is implemented by merging adjacent RAW10 pixels. PIXFMT supports this by taking eight PPC packets in these two modes, merging adjacent pixels, and treating them as four PPC packets. This is controlled by the VI_PIXFMT_WIDE register. Note that VI_PIXFMT_WIDE does not need to be enabled for CSI2 v2.0 native high-bit-depth modes.

Note that pixel merge is a PIXFMT function, which works behind CHANSEL. CHANSEL functions related to pixel coordinates, e.g., size checking and cropping, are still regarding to CSI input data format.

Functions related to pixel coordinates after PIXFMT merge, for instances PDAF extraction and cropping, are in merged wide data format.

Companding Module

Some HDR sensors have modes that compand higher-resolution raw samples. PIXFMT provides a companding engine with a programmable look-up table, allowing a multitude of sensor modes to be supported. Additionally, since the companding engine supports fractional scale factors, it can be used to expand as well as compress; this allows HDR linear data from a sensor to be converted to compressed-space for processing in ISP.

PIXFMT's compand unit implements a piecewise linear function with 10 knee points. All 10 knee points are tested for their base point in parallel, and the lowest matching comparator selects the LUT entry; afterwards, the base is subtracted from the pixel, a scaling function is multiplied in, and an offset is added back in to produce the output.

Only one set of compand knee points is supported across all VI channels. However, although all channels share the same companding table, each channel can have companding enabled and

disabled independently. The compand knee points are defined in the VI_PIXFMT_COMPAND_KNEE_ registers; the compand enable is present in the VI_PIXFMT_ENABLE register.

The compand scaling registers are sized to allow for a compression range of 1x to 1/128, and an expansion range of 1x to 1023x.

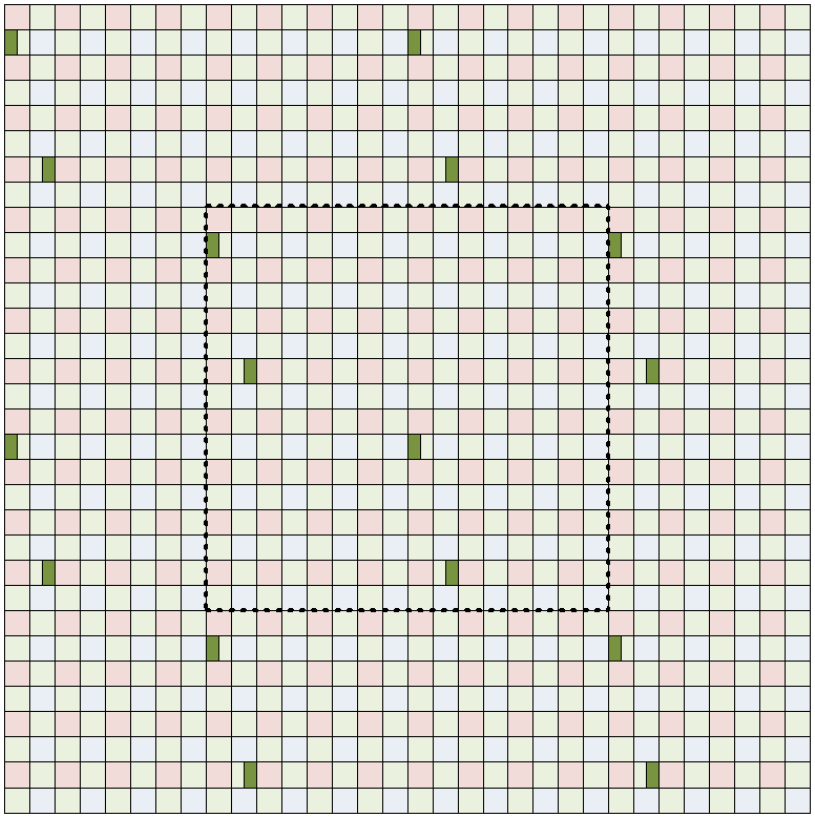
Because this feature supports both expanding and compression, it needs to function on modes from RAW10 all the way out to RAW20. Pixel companding occurs after VI_PIXFMT_WIDE merging and after MSB-repeating; the input format is normalized to RAW20.

Phase Detect Autofocus Pixel Separation and Replacement

Note: Phase Detect Autofocus (PDAF) on the SoC is limited to channels 0 through 11.

PIXFMT provides support for extracting individual pixels out of a frame. This support exists to assist PDAF algorithms that need to read specific pixels in order to do their work, but that do not wish to decode an entire DPCM stream. PIXFMT's PDAF support operates on a bitmap grid of 32 x 32 pixels, tiled over the surface of the image sensor.

Figure 7.13 Example PDAF Pattern



Some applications may wish to also dump out surrounding pixels–for instance, for a correlation on known neighbors, rather than a simple correlation on phase-detect pixels alone. This can be accomplished by setting additional bits in the bitmap corresponding to the neighboring pixels. To avoid a memory traffic amplification that would overwhelm the advantages from DPCM compression, the PDAF separator includes a crop operation, so that the algorithm can request only a small focus region of interest.

Pixels chosen by the PDAF separator are output both to the main output surface (which may include DPCM compression) and to the secondary output surface (i.e., the surface that UV or U output would be applied to). In PDAF mode, the secondary surface does not have a line stride, and outputs all captured pixels in one linear range. Optionally, the PDAF mode can override phase-detect pixels sent to the main output surface with a fixed value; doing so assists SAPOR in correcting away PDAF pixels, if they otherwise would be too close to valid-appearing neighbors. Pixels sent through the PDAF separator must be fully encompassed within the CHANSEL cropping region.

When all PDAF data are fully written to the output surface, a PDAF_DATA_DONE notification is generated. The coordinate of the last PDAF pixel is required to help flush buffered PDAF data to memory and generate PDAF_DATA_DONE. The last PDAF pixel coordinate must be valid; i.e., the corresponding PDAF pattern bit must be 1. The coordinates of the last PDAF pixel are programmed through a register.

The PDAF separator can be enabled even if PIXFMT is normally disabled. If PIXFMT is disabled, then no pixels are written to the main output surface, but data can still be written to the PDAF output surface. (This behavior is primarily vestigial from when a direct ISP link existed, but may still be useful in specialty applications.)

PDAF mode is valid only for RAW pixel types.

RAW Memory Formats

RAW data is sent to memory in four possible formats (in addition to the DPCM format). The T_R8 format is a classical 8-bit unsigned word format, packing four per 32-bit word. The T_R16 is a 16-bit unsigned word format, packing two per 32-bit word. T_R32 is a left-aligned 0.32 fixed-point format, packing one per 32-bit word. Finally, T_FP32 is an IEEE-754 single-precision floating point format, representing data on the [0,1) interval, also packing one per 32-bit word. The previous format from VI2, in which three 10-bit pixels are packed in a 32-bit word, is now unsupported; users of that application need to use DPCM instead. Both raw image modes either replicate MSBs or truncate LSBs, as needed to fit pixels into the appropriate format. (Concretely, input pixel formats have their MSBs replicated out to full width, and then are truncated down to the desired output pixel format.) Examples of these modes are given in the tables below.

Table 7.20 RAW Memory Formats

CSI Format Name
RAW6

CSI Format Name
RAW7
RAW8
RAW10
RAW12
RAW14
RAW16 (CSI native)
RAW16 (merged)
RAW20 (CSI native)
RAW20 (merged)
RAW24 (merged)

T_R8	bpp
T_R8	8
T_R16_I	16
T_R16	16
T_R24	24
T_R32	32
T_R16_F	16
T_R16_X_ISP20	16
T_R16_X_ISP24	16
T_R32_F	32
T_DPCM	var.

Figure 7.14 T_R16 Format

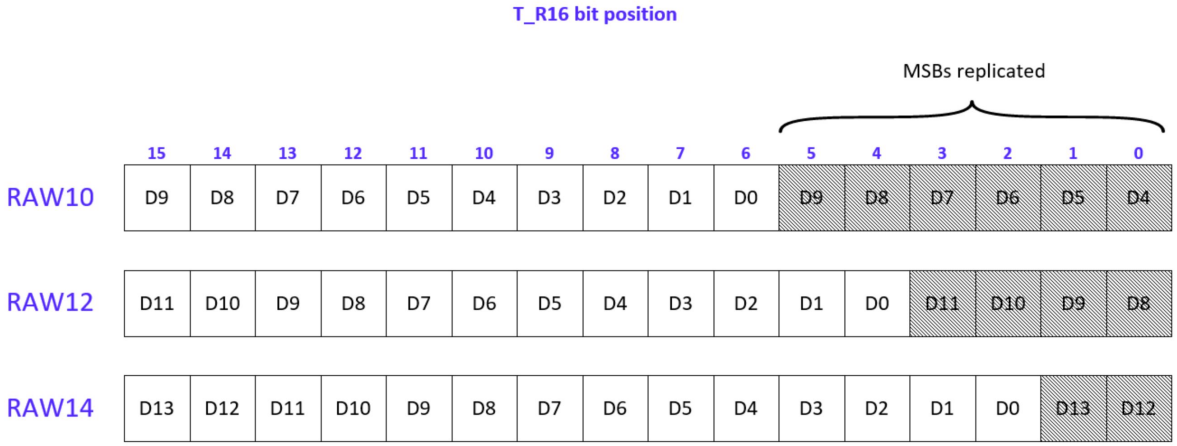


Figure 7.15 T_R16_I Format with Examples for RAW6, RAW10, RAW12, and RAW14

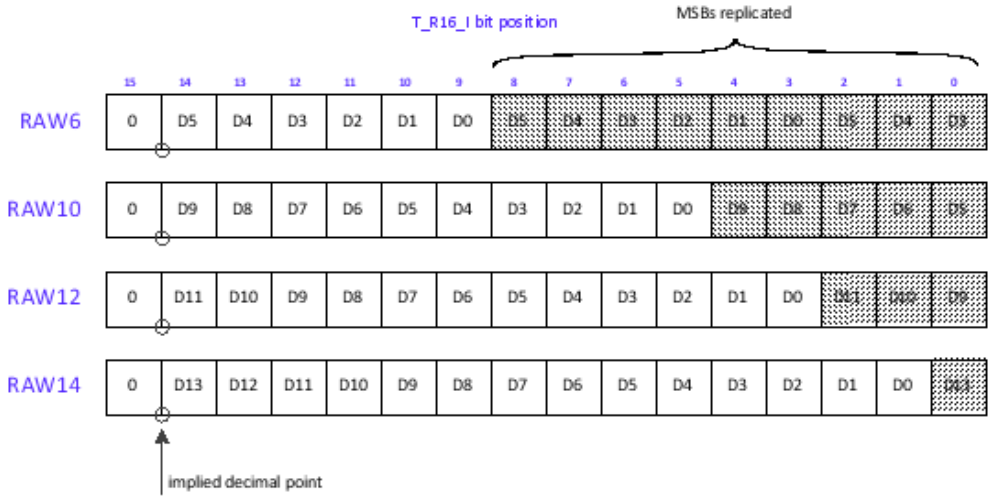
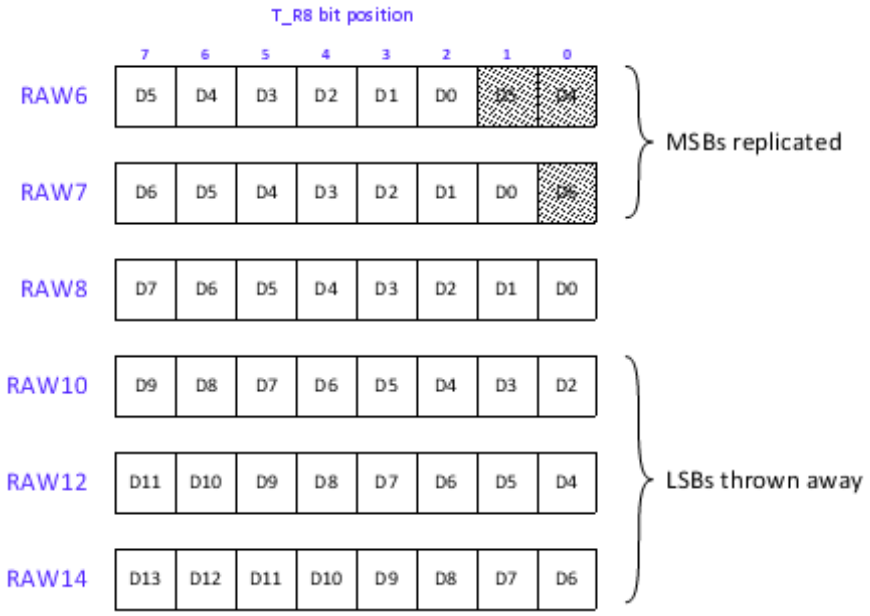


Figure 7.16 T_R8 Format with Examples for 6 through 14-bit RAW Modes



VI6 has support for ISP6's native floating-point pixel bus format. The ISP's pixel bus format is still a 16-bit format, but increases mantissa precision by two bits, and has slightly reduced dynamic range as compared to IEEE-754 half-precision floating point. ISP6's floating-point pixel bus, when operating in raw pixel mode, has a range of [-0.125, 2.0]. The mantissa is stored in bits [11:0], and the tag (combined exponent and sign) is stored in bits [15:12]; tag value interpretations are given below.

Table 7.21 ISP Floating Point Tag Representations

D[15:12]	Interpretation
4'h0	+0.mantissa * 2^{-8} (denorm)
4'h1	+1.mantissa * 2^{-8}
4'h2	+1.mantissa * 2^{-7}
4'h3	+1.mantissa * 2^{-6}
4'h4	+1.mantissa * 2^{-5}
4'h5	+1.mantissa * 2^{-4}
4'h6	+1.mantissa * 2^{-3}

D[15:12]	Interpretation
4'h7	+1.mantissa * 2^{-2}
4'h8	+1.mantissa * 2^{-1}
4'h9	+1.mantissa
4'hA	-1.mantissa * 2^{-4}
4'hB	-1.mantissa * 2^{-5}
4'hC	-1.mantissa * 2^{-6}
4'hD	-1.mantissa * 2^{-7}
4'hE	-1.mantissa * 2^{-8}
4'hF	-0.mantissa * 2^{-8} (denorm)

RAW24 Support

RAW24 is not a native data type defined in MIPI CSI-2 spec. To support uncompressed 24-bit precision data, some sensors like Sony IMX490 splits each single 24-bit pixel to two 12-bit pixels to transmit them in RAW12 style on CSI-2 protocol.

The 24-bit RAW data can be saved to memory in three options:

Memory format	BPP	Description	New/Legacy
T_R16_X_ISP24	16	ISP private floating point for 24-bit precision data	New
T_R24	24	ISP private 0.24 fixed-point format	New
T_R32	32	Legacy 0.32 fixed-point format	Legacy

RAW Padding Mode

Besides default MSB replication mode, VI6 adds other padding options on RAW8/10/12/14/20/24 to T_16/T_R32 conversion, as listed below.

Register	Field	Bits	Reset	Description
VI_CH*_PIXFMT_FORMAT_0	T_R16_R32_PAD0_EN	2	0	Control conversion rule for RAW8/10/12/14->T_R16; RAW20/24->T_R32; Should be 0 in non T_R16/T_R32 case. 0: Legacy mode: MSB replication on LSB 1: Data is right aligned, with MSB padded with 0 2: Data is left aligned, with LSB padded with 0 3: Reserved

The output RAW image can still be processed by ISP when PAD0_EN is set to 1 or 2 with T_R16 / T_R32 format, but it requires ISP being programmed in same PAD_EN mode to let MR convert the data back for further processing,

RGB Memory Formats

RGB data comes from NVCSI in one of several possible formats and VI provides a handful of options for sending the data to system memory. In VI, the input modes are decoupled from the output modes; when insufficient bits of data exist to fill out a channel, the MSBs are replicated into the LSBs, and when too much precision exists, the LSBs are truncated. Additionally, VI provides a luminance output format, which multiplies through by a matrix. The input data formats and output data formats are shown below.

Table 7.22 RGB Formats from NVCSI

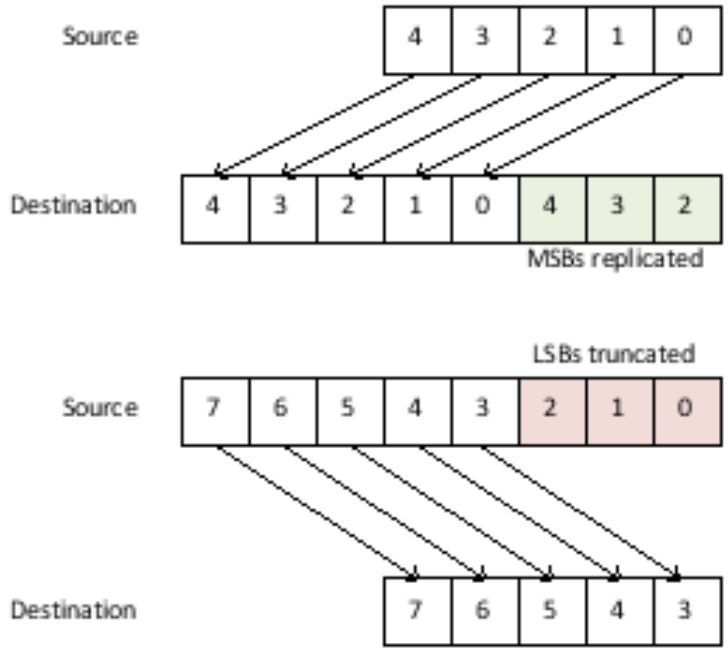
CSI Format
RGB444
RGB555
RGB565
RGB666
RGB888

Table 7.23 RGB Formats Supported by VI

Memory Format	BPP
T_A1B5G5R5	16
T_A1R5G5B5	16
T_B5G5R5A1	16
T_R5G5B5A1	16
T_R5G6B5	16
T_B5G6R5	16

Memory Format	BPP
T_A4B4G4R4	16
T_A4R4G4B4	16
T_B4G4R4A4	16
T_R4G4B4A4	16
T_A8R8G8B8	32
T_A8B8G8R8	32
T_R8G8B8A8	32
T_B8G8R8A8	32
T_A2R10G10B10	32
T_A2B10G10R10	32
T_R10G10B10A2	32
T_B10G10R10A2	32

Figure 7.17 RGB Bit Depth Conversion



YUV Memory Formats

VI also supports YCbCr / YUV formats. No support is provided for interpolating or decimating chroma data; users of that functionality need to process through ISP or through VIC. Non-planar, semi-planar, and fully planar formats are both available for 8-bit modes and 10-bit modes. Only 16-bit padded "inefficient" modes are supported in 10-bit YUV mode; more efficient memory capture can be provided with the YUV compression built into ISP. YUV capture modes are as shown in the table below.

Table 7.24 YUV Modes

CSI Format Name	Pixel Formats in Memory	bpp (per plane)	Notes
YUV422 8-bit	T_R8	8	Y component only
	T_Y8_U8_*Y8_V8 ; T_Y8_V8 Y8_U8 ; T_U8_Y8 V8_Y8; T_V8_Y8*_U8_Y8	32	YUV packed
	T_Y8___U8V8_N422	8	Semi-planar; Y+UV
	T_Y8___V8U8_N422	8	Semi-planar; Y+VU
	T_Y8__*U8*__V8_N422	8	Planar
YUV422 10-bit	T_R8	8	Y component only
	T_Y8_U8_*Y8_V8 ; T_Y8_V8 Y8_U8 ; T_U8_Y8 V8_Y8 ; T_V8_Y8*_U8_Y8	32	YUV packed
	T_Y8___U8V8_N422	8	Semi-planar; Y+UV
	T_Y8___V8U8_N422	8	Semi-planar; Y+VU
	T_Y8__*U8*__V8_N422	8	Planar
	T_Y10___U10V10_N422	16	Semi-planar; Y+UV
	T_Y10___V10U10_N422	16	Semi-planar; Y+VU
	T_Y10__*U10*__V10_N422	16	Planar
YUV420 8-bit legacy	T_R8	8	Y component only
YUV420 8-bit / 8-bit (CSPS)	T_R8	8	Y Component only.
	T_Y8___U8V8_N420	8	Semi-planar; Y+UV
	T_Y8___V8U8_N420	8	Semi-planar; Y+VU
	T_Y8__*U8*__V8_N420	8	Planar
YUV420 10-bit / 10-bit (CSPS)	T_R8	8	Y Component only.
	T_Y8___U8V8_N420	8	Semi-planar; Y+UV

CSI Format Name	Pixel Formats in Memory	bpp (per plane)	Notes
	T_Y8___V8U8_N420	8	Semi-planar; Y+VU
	T_Y8__*U8*__V8_N420	8	Planar
	T_Y10___U10V10_N420	16	Semi-planar; Y+UV
	T_Y10___V10U10_N420	16	Semi-planar; Y+VU
	T_Y10__*U10*__V10_N420	16	Planar
YUV444 8-bit	n/a	n/a	Does not exist in CSI2
YUV444 10-bit	n/a	n/a	Does not exist in CSI2

Format Conversion Constraints

YUV mode conversions that are permitted are provided preceding. RAW modes that can be converted are shown in matrix form below. Permissible RGB mode conversions are shown as a matrix in the table below.

Table 7.25 Matrix of RAW Format Conversions

	T_R8	T_R16_I	T_R16	T_R32	T_R16_F	T_R16_X_ISP20	T_R32_F	T_DPCM	T_R24	T_R16_X_ISP24
RAW6	Yes	No	No	No	No	No	No	n/a	No	No
RAW7	Yes	No	No	No	No	No	No	n/a	No	No
RAW8	Yes	No	Yes	No	Yes	Yes	Yes	n/a	No	No
RAW12	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
RAW14	Yes	Yes	Yes	Yes	Yes	Yes	Yes	n/a	No	No
RAW16	No	No	Yes	Yes	Yes	Yes	Yes	Yes	No	No
RAW20	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
RAW24	No	No	No	Yes	No	No	No	No	Yes	Yes

Table 7.26 Matrix of RGB Format Conversions

	T_...444	T_...555	T_...565	T_...888	T_...101010
RGB444	Yes	Yes	Yes	Yes	No
RGB555	No	Yes	Yes	Yes	No
RGB565	No	Yes	Yes	Yes	No
RGB666	No	Yes	Yes	Yes	No
RGB888	Yes	Yes	Yes	Yes	Yes

7.2.2.2.5 Register Programming

Register programming for PIXFMT is as shown in the table below.

Table 7.27 PIXFMT Register Programming

Register	Field	Bits	Reset	Description
VI_CH*n PIXFMT_FORMAT_(paged)*	FORMAT	8	undef	Pixel memory format for VI channel: T_R8 T_A4B4G4R4 T_A4R4G4B4 T_B4G4R4A4 T_R4G4B4A4 T_A1B5G5R5 T_A1R5G5B5 T_B5G5R5A1 T_R5G5B5A1 T_R5G6B5 T_B5G6R5 T_A8B8G8R8 T_A8R8G8B8 T_B8G8R8A8 T_R8G8B8A8 T_A2B10G10R10 T_A2R10G10B10 T_B10G10R10A2 T_R10G10B10A2 T_Y8_U8__Y8_V8 T_Y8_V8__Y8_U8 T_V8_Y8__U8_Y8 T_U8_Y8__V8_Y8 T_Y8__U8V8_N420 T_Y8__V8U8_N420 T_Y8__U8V8_N422 T_Y8__V8U8_N422 T_Y8_*U8*_V8_N422 T_Y8_*U8*_V8_N420
	FORMAT	8	undef	T_Y10__V10U10_N420 T_Y10__U10V10_N420 T_Y10_*U10*_V10_N420 T_Y10__V10U10_N422 T_Y10__U10V10_N422 T_Y10_*U10*_V10_N422 T_R16_ISP T_R16_F T_R16 T_R16_I T_R32 T_R32_F T_DPCM_RAW10 T_DPCM_RAW12 T_DPCM_RAW16 T_DPCM_RAW20
	FORMAT_PDAF	8	undef	Pixel memory format for PDAF pixels: T_R16_ISP T_R16_F T_R16 T_R32 T_R32_F

Register	Field	Bits	Reset	Description
VI_CH*n PIXFMT_WIDE _(paged)*	ENABLE	1	0	Whether to merge adjacent RAW8/RAW10 pixels.
	ENDIAN	1	0	Which order to merge adjacent pixels in. 0 for big endian, 1 for little endian.
VI_PIXFMT_COMPAND_KNEE_CFG0[0...9] NOTE: Register is global.	BASE	26:7 (U20.0)	0	Input position for this knee point.
VI_PIXFMT_COMPAND_KNEE_CFG1[0...9] NOTE: Register is global.	SCALE	16:0 (U10.7)	1.0	Scale preceding this knee point.
VI_PIXFMT_COMPAND_KNEE_CFG2[0...9] NOTE: Register is global.	OFFSET	26:0 (U20.7)	0.0	Output offset for this knee point.
VI_CH*n PIXFMT_ENABLE _(paged)*	ENABLE	1	1	Whether PIXFMT writes non-pixels for this channel.
	COMPAND	1	0	Whether PIXFMT enables the companding unit for this channel.
	PDAF	1	0	Whether PDAF separation is enabled for this channel.
VI_CH*n PIXFMT_PDAF_CROP_X _(paged)*	LEFT	16	0	Within a line, pixel position at which PDAF separation begins.
	RIGHT	16	0	Within a line, pixel position at which PDAF separation ends.
VI_CH*n PIXFMT_PDAF_CROP_Y _(paged)*	TOP	16	0	Line at which PDAF separation begins.
	BOTTOM	16	0	Line at which PDAF separation ends.
VI_CHn_PIXFMT_PDAF_LAST_PIXEL	X	16	0	X coordinate of the last PDAF pixel.
	Y	16	0	Y coordinate of the last PDAF pixel.
VI_PIXFMT_PDAF_PATTERN[0...31] NOTE: Register is global.	ENABLE	32	0	Pixel bitmap, by line. (PATTERN[y0][x0] is set if the pixel (x % 32) == x0, (y % 32) == y0 should be output to the PDAF surface.)
VI_CH*n PIXFMT_PDAF_REPLACE _(paged)*	ENABLE	1	0	Whether to replace PDAF pixels sent to the primary surface with an alternative value.
	VALUE	20	0	Value to replace PDAF pixels with (in pixel bus format).
VI_PIXFMT_PDAF_PATTERN_REPLACE [0...31] NOTE: Register is global.	REPLACE_ENABLE	32	0	Pixel bitmap, by line. (PATTERN_REPLACE[y0][x0] is set if the pixel (x % 32) == x0, (y % 32) == y0 should be replaced.)

Register	Field	Bits	Reset	Description
VI_CH*n* *PIXFMT_PDAF_REPLACE_CROP_X_(paged)*	LEFT	16	0	Within a line, pixel position at which PDAF replacement begins.
	RIGHT	16	0	Within a line, pixel position at which PDAF replacement ends.
VI_CH*n* *PIXFMT_PDAF_REPLACE_CROP_Y_(paged)*	TOP	16	0	Line at which PDAF replacement begins.
	BOTTOM	16	0	Line at which PDAF replacement ends.

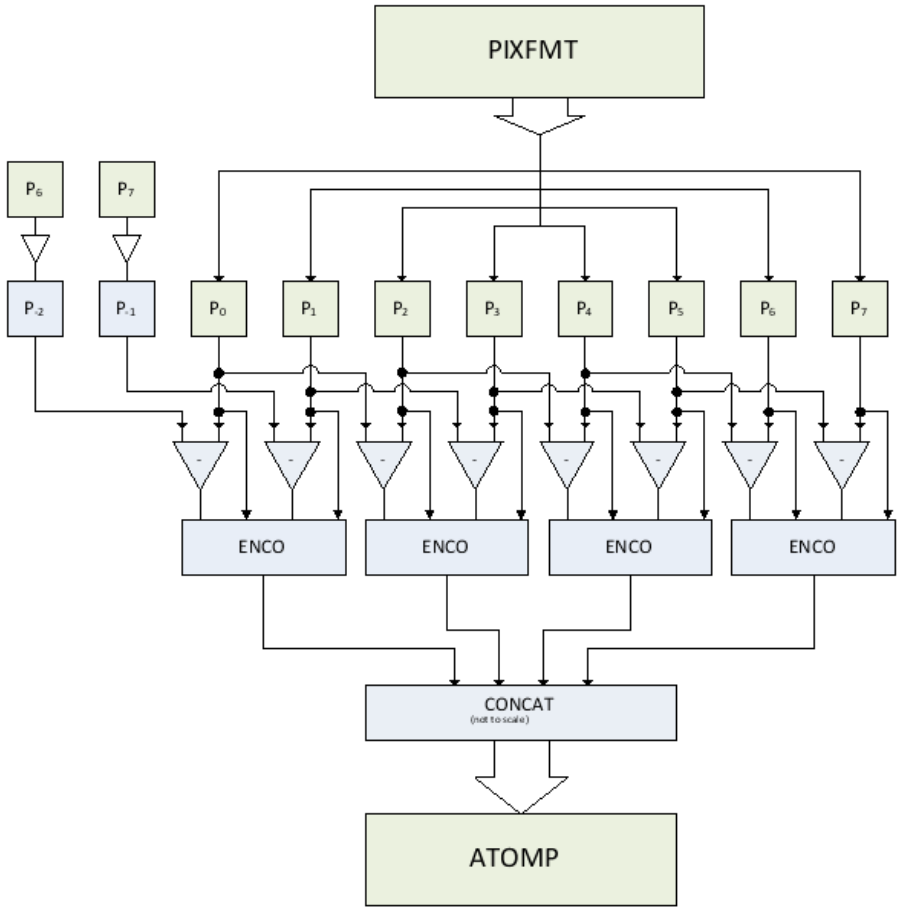
Differential Pulse Code Modulation

The Differential Pulse Code Modulation (DPCM) module implements lossless variable-bit-rate compression. The basic scheme is a familiar differential PCM encoding wrapped in an optimal code; two pixel differences are processed at a time to reduce overhead, allowing encoding complexity reductions while maintaining a reasonable bit rate savings. The DPCM line codec supports RAW10, RAW12, RAW16, and RAW20 data.

The entire DPCM datapath is shown below. The DPCM module is responsible for emitting bytes, rather than bits to ATOMP; when sufficient bits are not available for a whole number of bytes, the remainder is stored in a buffer.

A VI channel with DPCM enabled, for the most part, can be treated like a VI channel without DPCM. However, compression is not fully transparent. Notably, the DPCM scheme can result in *expansion* for sufficiently pessimal input; so, the line stride in the atom packer must be set to account for this. With the current encoding formulation, the worst case expansion is an additional three bits per two RAW10 pixels or 15%. The line stride must then be 15% larger than it would be for fully packed RAW data.

Figure 7.18 Overview of DPCM Block



Strip Mode Restart Points

Variable-rate compression systems that require applications to seek to particular regions within a DPCM line – for instance, an MR decoding a strip mode frame – cannot directly do so, in a crude implementation, without having previously decoded all prior pixels. To make this more efficient, VI provides "restart points" which behave like virtual line breaks. VI implements this by synthesizing an end of line after a variable number of input pixel packets, causing each restart to occur at the channel's line stride boundary.

VI is not capable of synthesizing restarts within a pixel packet, since that would result in emitting two atoms for one input packet, and the pipeline must move in lock-step. As a result, restart widths must be a multiple of eight pixels.

Deferred Atom-Count Write

Another aspect of the DPCM system is that a reader cannot know in advance how many atoms consist of a DPCM chunk, and hence cannot know how many atoms it needs to prefetch from the

memory subsystem. VI solves this by preallocating space for the count of atoms written, but deferring the actual write until the end of the chunk has been written, as described in Algorithm 3.

In the strip-mode case, the reader may not need to read a full chunk, but instead may only need to read a configurable overfetch region; to support this, VI also writes out an atom count corresponding to the number of atoms that a reader would need to fetch to read both the left and right overfetch region. This is computed from the VI_DPCM_STRIP register programming.

Algorithm 3: DPCM Data Placement

We refer to each atom as atom n, where the atom that is located at the base address for the chunk is atom 0, the atom that immediately follows that in memory is atom 1, and so on.

The first data from each chunk are written into atom 1 (not atom 0), and data proceeds upwards into subsequent atoms until the final input packet for the chunk has been reached. Once the final input packet for the chunk has been concatenated into an atom, the next step depends on the number of bytes remaining in the atom. If there are at least four bytes remaining in the atom, the final four bytes take on the value {atom count counter, overfetch atom counter}, and the atom is written into atom 0. If not, the atom is written into its normal place in memory, and atom 0 is written as being empty with only the final four bytes set as previously described.

The counters are formatted such that the MR unit can efficiently determine how many atoms it needs to prefetch. The counters contain the number of atoms that are not atom 0; if data for that counter is stored in atom 0, the most significant bit is additionally set. For instance, if data are stored in atoms 1, 2, 3, 4, 5, and 0, then the atom counter would be 0x8005; if only in 1, 2, 3, and 4, then the atom counter would be 0x0004.

MODE Register Programming

Since the SoC has modes that overlap in functionality, for instance backwards-compatible RAW12 mode and new run-length-encoding RAW12 mode, the incoming data-type is insufficient to select a mode alone. Unless noted, selecting a mode that is incompatible with its native data-type is undefined behavior that shall not be validated.

Table 7.28 DPCM Register Description

Register	Field	Bits	Reset	Description
VI_CH*n DPCM_STRIP _(paged)*	OVERFETCH	16	undef	Number of packets in overfetch region. Usually set to the sum of maximum left and maximum right overfetch. (0 to disable.)
VI_CH*n DPCM_CHUNK_FIRST _(paged)*	PACKETS	16	undef	Number of packets in first-generated chunk (no OVERFETCH region in first chunk).
VI_CH*n DPCM_CHUNK_BODY _(paged)*	PACKETS	16	undef	Number of packets in "body" chunks (including OVERFETCH region, if enabled).
	COUNT	16	undef	Number of "body" chunks to emit.

Register	Field	Bits	Reset	Description
VI_CH*n* *DPCM_CHUNK_PENULTIMATE _(paged)*	PACKETS	16	undef	Number of packets in chunk immediately after "body" chunks (including OVERFETCH region, if enabled).
VI_CH*n* *DPCM_CHUNK_LAST _(paged)*	PACKETS	16	undef	Number of packets in final-generated chunk (including OVERFETCH region, if enabled).
VI_CH*n* *DPCM_STATISTICS _(paged)*	LAST_ COMPRESSED	32	undef	Number of compressed bits output in previous frame.
	TOTAL_ COMPRESSED	32	undef	Number of bits output total. (Register is writable.)
VI_CH*n*_DPCM_MODE	MODE	3	3'h0	Encoding format used by DPCM. 0: Parker-style RAW10 format. 1: Parker-style RAW12 format. 2: Run-length-encoded RAW10 format. (If not implemented specifically, the same as RAW12 format, but with LSBs set to zero.) 3: Run-length-encoded RAW12 format. 4: RAW16 format for logarithmic data. (P0.5) 5: RAW20 format.
VI_CH*n* _DPCM_CLAMP_HIGH	CLAMP_VALUE	20	20'hFFFFF	Maximum value to truncate input data to.
VI_CH*n* _DPCM_CLAMP_LOW	CLAMP_VALUE	20	20'h0	Minimum value to truncate input data to.

ATOMP

The atom packer serves to align pixel packets and build memory requests to be emitted to the MC. With the responsibility of building memory requests comes the responsibility of holding memory state. ATOMP controls the current memory output address for a channel, controls the stride of lines (or restarts), and determines which atoms require write acknowledgments, generates notifications on returned write acknowledgments.

ATOMP's basic mechanism of operation is the management of atom buffers, in which it accumulates data before submitting each to the output FIFO. ATOMP works on a byte-level; from the stage up the pipeline, ATOMP receives a byte enable, and a data packet. Additionally, ATOMP manages the logic to perform the atom inversion and count appending specified by the DPCM definition.

Input Format from PIXFMT

PIXFMT's output format is described preceding, but the mechanics of how ATOMP handles it is worth noting. A sample implementation of a packet from PIXFMT contains the following signals:

- a channel number

- an input stream ID
- 32 bytes (256 bits) of data, with a validity assertion signal per byte
- a mode selection signal for non-planar mode, semi-planar mode, full planar mode, or embedded data
- and a frame status signal, with {beginning,end} of {frame,line} signals. (Note that the EOF signals correspond to CHANSEL-generated EOF signals, not to NVCSI FE signals.)

Embedded data is sent to a different base address than the frame that it came in with, though for the purposes of surface numbers to OFIF it is included in surface 0.

The frame status signals control an internal frame number increment inside ATOMP (for sending line-complete signals) and controls line-flush and write-ack generation logic.

Heterogeneous Channels

The capabilities associated with each channel are as shown below. For software that wishes to discover the capabilities of each channel, they can be probed at run time by reading the VI_ATOMP_CHANNEL_SUPPORT registers.

Table 7.29 ATOMP Heterogeneous Channel Configuration

Channels	Bayer / YUV packed / Y-only	Planar	Semi-planar	PDAF
0-11	Yes	Yes	Yes	Yes
12-35	Yes	No	No	No

OFIF

The output FIFO manager is the last level of VI's path to memory before the memory interface. It commits packets out to memory and makes the decision to drop channels when MCCIF's FIFOs are full. It also keeps track of outstanding requests and emits notifications to the system when important events have happened.

OFIF is capable of emitting five different notifications: transfer completion events (frame done, embedded data done, frame partial transfer complete), and error events (frame truncated and frame tossed). These are implemented as ATOMP notifications.

OFIF maintains a queue of outstanding transactions that need notification, sending completion events to NOTIFY only when MCCIF reports a write acknowledgment. The queue contains frame IDs and update IDs to be sent along to NOTIFY when the appropriate transaction hits memory.

The remaining functionality of OFIF is as a gatekeeper for the memory interface. OFIF can be configured to drop outbound atoms from specific channels when the FIFO's occupancy has reached critical conditions. OFIF also holds housekeeping statistics about FIFO occupancy that can be read at any time.

Multithreaded FIFO

To make the most of activate cycles in MC, OFIF batches up outbound requests and attempts to emit multiple atoms from a single surface at once if it can. This behavior is implemented with a multithreaded FIFO, one thread for each surface in each stream, and a set of thresholds that control which channel is output from the FIFO.

The VI6 output FIFO groups by stream number, and then by surface. This behavior is similar to the backend processors in VI6 ATOMP. Instead of four-atom boundaries, it operates on boundaries controlled by a programmable address alignment mask (register BATCH_MASK). The boundary can be set from one atom (64 bytes) to 1024 atoms (64 kilobytes). Atoms of a thread are accumulated in the FIFO until at least one of the following conditions is met, upon which atoms are released:

- A change in address portion masked by register BATCH_MASK hot bits. For example, when BATCH_MASK is set to 0xfffffc00 (1kB alignment), if a first atom is at 0x3c0, and a second atom is at 0x400, then the second atom causes the first atom to be flushed.
- The number of pending atoms reaches the flush threshold. This is derived from register BATCH_MASK.
- An atom carrying the last pixel of a line arrives at the FIFO.
- A frame end (FE) atom arrives at the FIFO.

OFIF performs round-robin arbitration between available surfaces; the arbitration grant is held for any given thread until another thread meets the flush conditions described preceding. This can be implemented with an ARBGEN weighted round-robin arbiter.

Table 7.30 OFIF Flush Threshold

BATCH_MASK	Flush threshold (#atoms)	Flush threshold (#bytes)
xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxx1 to xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_x100_0000	1	64
xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_1000_0000	2	128
xxxx_xxxx_xxxx_xxxx_xxxx_xxx1_0000_0000	4	256
xxxx_xxxx_xxxx_xxxx_xxxx_xx10_0000_0000	8	512
xxxx_xxxx_xxxx_xxxx_xxxx_x100_0000_0000	16	1024
xxxx_xxxx_xxxx_xxxx_xxxx_1000_0000_0000	32	2048
xxxx_xxxx_xxxx_xxxx_xxx1_0000_0000_0000	64	4096
xxxx_xxxx_xxxx_xxxx_xx10_0000_0000_0000	128	8192

BATCH_MASK	Flush threshold (#atoms)	Flush threshold (#bytes)
xxxx_xxxx_xxxx_xxxx_x100_0000_0000_0000	256	16384
xxxx_xxxx_xxxx_xxxx_1000_0000_0000_0000	512	32768
xxxx_xxxx_xxxx_xxx1_0000_0000_0000_0000 to 1000_0000_0000_0000_0000_0000_0000_0000	1024	65536

The FIFO is sized to store 1335 atoms: 759 of them are reserved for bandwidth disruption, and the other 576 atoms are for batching and DRAM thermal management. Batching and DRAM thermal management are estimated with the following assumptions:

- One stream is outputting YUV full–planar, and five streams are outputting RAW (two planes with PDAF enabled), for a total of 13 threads.
- The batch threshold is set to 1 kB.
- During a DRAM outage event, the worst case input is RAW/RGB 8 PPC and the output is a 32 bpp surface (e.g., T_R32* and T_A8R8G8B8).
- At worst, 36 FEs can arrive or be forced. Each of them becomes an FE atom.

OFIF implements a two-level quality-of-service mechanism by using a pair of watermarks. When the amount of space available (in atoms) in the DVFS FIFO goes below a channel's HIGH_WATERMARK, OFIF engages its first quality-of-service limit: packets for frames that have already begun are allowed to proceed, but all packets for any new frames for the channel are discarded. (When a new frame is rejected, a FRAME_TOSSED notification is emitted.) If the amount of space available in the DVFS FIFO continues to fall below a channel's LOW_WATERMARK, OFIF engages its second quality-of-service limit, and truncates the frame in progress on that channel, discarding all further packets for that frame. When this occurs, a FRAME_TRUNCATED notification is emitted. In this fashion, less critical frames can be proactively discarded to ensure that FIFO space exists for channels that are more important.

Note that the reset values for LOW_WATERMARK and HIGH_WATERMARK are not appropriate for applications that expect to use the entire DVFS FIFO. In applications that can result in DVFS pauses, both watermarks need to be set to 0 to allow the full DVFS FIFO depth to be used. Applications that need to maintain quality of service may require other settings.

NOTIFY

The notification engine serves as a mechanism to transmit out-of-band events back to a host. As various events appear from various stages of VI, the notification engine serializes them and stores them for further retrieval by a CPU of some form.

NOTIFY deals with possible events that VI contends with in a many-camera many-operation system. Notify is designed to be used in tandem with the Falcon processor (or another real-time

processor in some applications), allowing the benefits of a real-time application alongside the programmability of a CPU.

Event Sources

The core element of NOTIFY is the event packet. An event packet's source is described as a tuple of a tag (i.e., the event type), a channel number, and a frame number. (Event packets also sometimes have a payload.)

The following event types exist as notification sources:

- (CSIMUX) Frame start / frame end events. These occur at the beginning and end of the frame, as signaled over CSI2. Since these are the closest to the CSI interface, they may be used to determine A/V sync. An FS event has tag 0, and an FE event has tag 1.
- (CSIMUX) Frame fault events. These occur when an input error occurs that can be positively associated with a frame. These may be concurrent with frame start or frame end events. The meaning is encoded in the payload. A frame fault event has tag 2.
- (CSIMUX) Stream fault events. These occur when an input error occurs that cannot be associated with a frame. The meaning is encoded in the payload. A stream fault event has tag 3.
- (CHANSEL) Start of frame / end of frame / start of embed / end of embed events. These occur at the beginning and end of channel-delimited frames. These events have respective tags 4, 5, 6, and 7.
- (CHANSEL) Line reached events. These occur when the line timer in CHANSEL expires. This event has tag 8.
- (CHANSEL) Single-channel fault (with potential embedded frame end). These occur when a fault attached to a single channel occurs. These are described in the CHANSEL section preceding. This event has tag 9 when no frame end is embedded, and 10 when a frame end is embedded.
- (CHANSEL) Many-channel faults. These are described in the CHANSEL section preceding. These events have tag 11 for no-match, tag 12 for a match collision, tag 13 for short frame, and tag 14 for a load collision.
- (ATOMP) Surface packer overflow. This event occurs when a single surface packer has overflowed. If this event occurs, contact the VI architecture team. This event has tag 15.
- (ATOMP) Start of frame / end of frame. These events occur once the NVCSI FS/FE that were matched in CHANSEL have reached ATOMP, and hence delimit when it is safe to reprogram a channel. (If these are merged into simultaneous notifications, then a special variant of an FE event is emitted.) These events have respective tag values of 16 and 17.
- (ATOMP / OFIF) Frame done / embedded data / PDAF data done. These events occur when a full frame, a full set of embedded data, or the last of phase-detect autofocus data, has been flushed out to memory, and a write ACK has been received from MC. These events have respective tag values of 18, 19, and 23.
- (ATOMP / OFIF) Frame partial completion. This event occurs when a partial set of lines has been written to memory, and a write ACK has been received. Note that these notifications are applicable only to the main surface. This event has a tag value of 20.

- (ATOMP / OFIF) Frame truncated / frame tossed. These events occur when a frame is interrupted either mid-frame or without writing any data out at all because of memory controller congestion. These events have respective tag values of 21 and 22.
- (VGPIO) VGPIO interrupt. This event occurs when a VGPIO transitions in a programmed fashion. This event has tag values of 29 and 30.
- (FML) FM-Lite completion. This event occurs with a focus metric value that has been computed by FM-Lite at the end of a frame. This event has a tag value of 31.

Processor Transport

As notifications come from the various event sources, they are time-stamped, sorted, and then enqueued in an internal buffer. The buffer is made accessible to processors through registers.

The NOTIFY output buffer communicates with Falcon with one interrupt.

VI6 NOTIFY does a coarse-grained classification of each event into one of two groups: no-output and normal-priority; it does so by looking only at the tag for each event, as programmed through the classification configuration registers. No-output events, as the name suggests, are always discarded: the equivalent behavior is as if they were never emitted to begin with. Normal-priority events are routed to the output buffer directly; and update a counter of important events that have not yet been processed by host processor.

The timestamp mechanism uses the global system TSC as a source. To save space in the per-source staging FIFOs, the full system timestamp is not encoded in each entry; only the 18 least significant bits are encoded in each FIFO entry, resulting in a maximum residency of 8ms in the input FIFO. To sort them at pop time, the LSB-encoded timestamp is subtracted from the LSBs of the current system timestamp, and the largest value is the oldest. (The full system timestamp of the event is similarly reconstructed at pop time.)

It is crucial that a notification is read out in time. If a notification stays in the FIFO too long (> 8ms), then encoded 18-bit timestamp overflows and VI hardware fails to send out notification in correct order and thus fail system operation. Timestamp overflow is reported to VI master error and as an uncorrectable error to HSM.

Intermediate per-source FIFOs are sized based on the maximum reasonable number of consecutive notifications that they can produce. Since a single intermediate FIFO overflowing is considered catastrophic, it is important that potentially high-volume notification sources can withstand bursts from other notification sources that may also be producing errors.

The procedure to read from the readout FIFO is to first read from VI_NOTIFY_FIFO_TAG, which pops an entry from the FIFO into _TAG, _TIMESTAMP, _DATA, and _EXT_DATA and returns the popped entry; then, read the remaining data from _TIMESTAMP, _DATA, and _EXT_DATA.

When every time a notification tagged as high-priority is pushed to the read-out FIFO, VI increments a counter. The counter is available to be read in VI_NOTIFY_HIGHPRIO; writing a value to _HIGHPRIO subtracts that amount from the counter (clamping at zero). The NOTIFY interrupt request line is asserted as long as there are entries in the read-out FIFO; as soon as the last entry is

read (i.e., it is popped out with a read to VI_NOTIFY_FIFO_TAG), the NOTIFY interrupt line is deasserted.

It is an error for any intermediate source FIFO to overflow; this error is signaled by setting the NOTIFY_FIFO_OVERFLOW bit in the VI master error register, which triggers a VI master error interrupt as a result.

Falcon

As mentioned preceding, VI6 has a Falcon microprocessor to process notification events. Special hardware peripherals interface the Falcon to VI6's register file and to the rest of the system.

The Falcon is responsible for managing channel state in the pipeline and for communicating frame completion state to the rest of the system. The Falcon presents to the system programmer an interface that processes a list of frames.

The general model of the Falcon, as described preceding, is a list of frames to be processed on a channel. A frame generally consists of a basic set of registers (for instance match registers, data-type registers, and destination registers), and a set of events that constitute frame success or failure. When the final disposition of a frame has been decided upon, i.e., whether it succeeded or failed, a frame contains a list of actions for each case. Actions can be incrementing a Host1x syncpt through the sync point shim, incrementing a GPU semaphore, raising a system interrupt, or other internal events.

Virtualization and Security Requirements

The SoC supports an environment in which multiple operating systems can coexist. As a result, VI is virtualized; multiple guest operating systems (i.e., virtual machines) may individually request channels on VI, and each guest is isolated from the others.

VGPIO

VI provides for six programmable GPIOs to control various aspects of the camera system and all of them may be configured as either I/O or PWM.

Table 7.31 VGP1-VGP6 Functions

GPIO	Function A	Function B	Function C
VGP1	I/O	PWM	XVS[0-2] / XHS[0-2]
VGP2	I/O	PWM	XVS[0-2] / XHS[0-2]
VGP3	I/O	PWM	XVS[0-2] / XHS[0-2]
VGP4	I/O	PWM	XVS[0-2] / XHS[0-2]
VGP5	I/O	PWM	XVS[0-2] / XHS[0-2]

GPIO	Function A	Function B	Function C
VGP6	I/O	PWM	XVS[0-2] / XHS[0-2]

The Pulse Width Modulation (PWM) unit generates a series of 128 identical pulses. The unit may use any of the (VGP1-6) pins as an output, provided that the unit is enabled and the pin is configured to carry PWM data. A pulse's length (in clock cycles), duty cycle and starting polarity are programmable. A single pulse can take between 2 and 32 clock cycles, where a high or low level is no more than 16 cycles long.

XVS[0-2] and XHS[0-2] are video synchronization signals generated by SYNCGEN.

The PWM and SYNCGEN units use VI_CONST_CLK. True to its name, VI_CONST_CLK is constant, but VI_CLK can be dynamically scaled; this allows for consistent output frequencies, even as the rest of the system is reconfigured.

A series is always 128 times the length of a single pulse. Once the unit is enabled, a series can be either generated once, a set number of times, or repeatedly until the unit is disabled.

Any interrupts configured for VGP pins result in an interrupt being generated for the Falcon. XVS[0-2] / XHS[0-2] are muxed with VGPIO internally in VI (controlled by PIN_OUTPUT_SELECT_VGP*).

SLVS-EC SYNCGEN

As it was no longer required, the SLVSEC camera interface is not present.

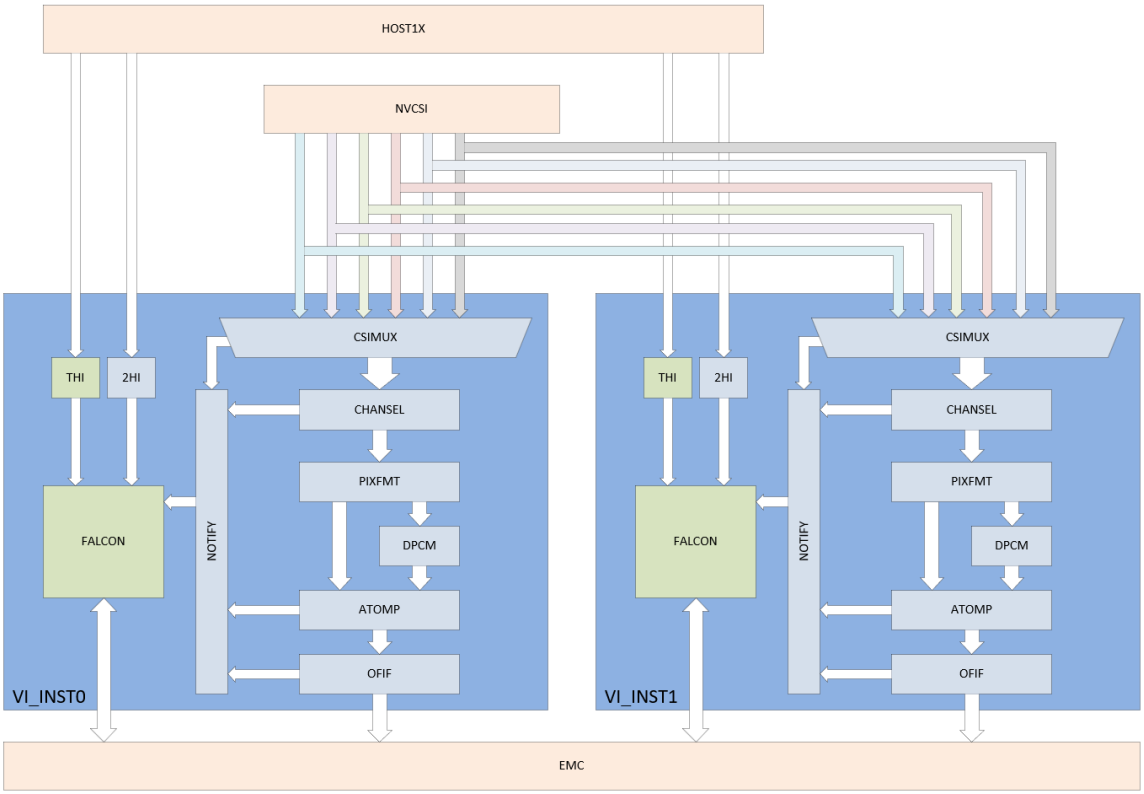
However, the related SYCNGEN module is still kept in VI6, because it provides an alternative way to generate some accurate periodic pulses.

VI6 has three SYNCGEN instances, and as such, drives out three pairs of synchronization signals (namely, XVS0/XHS0, XVS1/XHS1, and XVS2/XHS2). It can then drive up to three different timing patterns to image sensors.

INCK	Input clock to image sensor
XVS	Vertical sync signal transmitted to sensor, active low
XHS	Horizontal sync signal transmitted to sensor, active low
HCLK	Internal generated horizontal scan clock

Second VI Instance

A second VI instance is is available in some variants, allowing up to 72 VI channels, and providing redundancy for some use cases.



As shown in preceding diagram, each of the VI instance has its own host1x i/f (for THI/2HI) and AXI i/f (for Falcon/pipeline). The output six streams from NVCSI are routed to both VIs' input, Software should take care of the CSIMUX setting to allow two VI instances processing different NVCSI streams. The NVCSI streams can be terminated at the CSIMUX input for each VI instance by programming VI_CSIMUX_INPUT_STREAM_MUX_0 if necessary.

The two VI instances share same VI clock / FLCG from CAR. But the reset signal is separated hence each one can be put into reset state individually.

7.2.2.3 Video Input Registers

7.2.2.3.1 General VI Registers

VI_EC_FEATURE_0

Offset: 0xc00
Byte Offset: 0x3000
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00440003 (0b0000,0000,0100,0100,xxxx,xxxx,xx00,0011)

Bit	Reset	Description
31:16	0x44	**NUM_ERR**: Number of errors connected to this collator. This is passed as a build time option to the plugin
5:0	0x3	**NUM_ERR_SLICES**: Number of error slices supported by this error collator, does not include the GlobalSpace and is derived by ceil (NUM_ERR/32). SW shall first read this register to determine the number of slices and read the required number of Error_Status registers .

VI_EC_SWRESET_0

Offset: 0xc01
Byte Offset: 0x3004
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SWRST**: 1'b1: Issue a SW reset to the Error Collator. This will reset all the registers (Except SCR), counters and logic of the Error Collator. SW can use this bit to flush errors logged into the error collator for ex, after Boot, SC7/8 exit. 1'b0: Do nothing, reset value. This bit is auto-cleared.

VI_EC_MISSIONERR_TYPE_0

Offset: 0xc02
Byte Offset: 0x3008
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0101)

Bit	Reset	Description
5:0	0x5	**CODE**: This register indicates the fault code of the error line based on the value of MISSIONERR_INDEX Register. This can be used by a fault handling agent to triage an error without requiring device-specific code.The possible values of this field are: 6'd0 : None 6'd1 : Parity Error on internal data path 6'd2 : ECC SEC Error on internal data path 6'd3 : ECC DED Error on internal data path 6'd4 : Comparator Error 6'd5 : Register Parity Error 6'd6 : Parity Error from on-chip SRAM/FIFO 6'd7 : ECC SEC Error from on-chip SRAM/FIFO 6'd8 : ECC DED Error from on-chip SRAM/FIFO 6'd9 : Clock Monitor Error 6'd10 : Voltage Error 6'd11 : Temperature Error 6'd12 : CRC Error on internal data path 6'd16 : SW Corrected Error 6'd17 : SW Uncorrected Error 6'd18 : SW Generic Error 6'd20 : ECC SED Error on internal data path 6'd21 : ECC SED Error from on-chip SRAM/FIFO 6'd32 : Other HW Corrected Error 6'd33 : Other HW Uncorrected Error

VI_EC_CURRENT_COUNTER_VALUE_0

Offset: 0xc03
Byte Offset: 0x300c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
8:0	0x0	**VALUE**: Provides the current value of the counter corresponding to the error in MissionErr_Index Register. Default provides the value of error 0 counter. Bit[8] is the ovrerflow bit post which the counter saturates and does not counter further.

VI_EC_MISSIONERR_USERVALUE_0

Offset: 0xc04
Byte Offset: 0x3010
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0xdead0000 (0b1101,1110,1010,1101,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0xdead0000	**VALUE**: Reflects the value on the error_<i>_user when error_<i>_pulse = MissionError. Only the user signals corresponding to the first error_<i>_pulse only will b elatched.Subsequent values will be dropped. The signals will get latched only when ERRSLICE<n>_MISSIONERR_STATUS is CLEAR. IPs must tie off the unused bits of this signals to 0, if unused.

VI_EC_MISSIONERR_INDEX_0

Offset: 0xc05
Byte Offset: 0x3014
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000,0000)

Bit	Reset	Description
6:0	0x0	**IDX**: BINARY Encoded. For error number 32, register should be programmed with value 0x20. Write to this register with Error number will update: - MISSIONERR_TYPE Register with the Error-Code for the Error. - CURRENT_COUNTER_VALUE Register with the error's SEC/DED Counter. - MISSIONERR_USERVALUE with value of the first error_<i>_user signal. SW can use this to trage the error. number shall update the MISSIONERR_TYPE register with the error code and the Current_Counter_Value register with the value of the errors SEC/DED counter. SW can use this register to triage the error.

VI_EC_CORRECTABLE_THRESHOLD_0

Offset: 0xc06
Byte Offset: 0x3018
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**COUNT**: Threshold value for all SEC Fault Reporting Units connected to this error collator. SEC Errors are logged once the threshold is reached and the overflow bit is set. 7'b0 : Log SEC error after receiving 1 Error. 7'b1 : Log SEC error after receiving 2 Errors. ... 7'bFF : Log SEC error after receiving 256 Errors.

VI_EC_MISSIONERR_INJECT_UNLOCK_0

Offset: 0xc07
Byte Offset: 0x301c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Writes to ERRSLICE_XXX_MISSIONERR_INJECT registers are disabled until this register is written with a value of 0xE1. This is to prevent an inadvertent safety error injection in the safety plugin due to: 1. A fault on ERRSLICE_XXX_MISSIONERR_INJECT register itself. 2. Erroneous SW. The register shall be written with a value of 0x0 to reestablish the lock after user has completed the error injection testing. 0xE1 : Unlock the MISSIONERR_INJECT Register 0x0 : Lock the MISSIONERR_INJECT Register **0** = LOCK **225** = UNLOCK

VI_EC_ERRSLICE0_MISSIONERR_ENABLE_0

Offset: 0xc0c
Byte Offset: 0x3030
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31	0x1	**ERR31**: 1'b1 -> Enable Mission Error Reporting for Parity Error on internal data path from FAULT_UDF_TAG_VI_CHANSEL_DATA_PARITY_ERR 1'b0 -> Disable Mission Error Reporting for Parity Error on internal data path from FAULT_UDF_TAG_VI_CHANSEL_DATA_PARITY_ERR **0** = DISABLE **1** = ENABLE
30	0x1	**ERR30**: 1'b1 -> Enable Mission Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM5_ECC_DED 1'b0 -> Disable Mission Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM5_ECC_DED **0** = DISABLE **1** = ENABLE
29	0x1	**ERR29**: 1'b1 -> Enable Mission Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM5_ECC_SEC 1'b0 -> Disable Mission Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM5_ECC_SEC **0** = DISABLE **1** = ENABLE
28	0x1	**ERR28**: 1'b1 -> Enable Mission Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM4_ECC_DED 1'b0 -> Disable Mission Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM4_ECC_DED **0** = DISABLE **1** = ENABLE
27	0x1	**ERR27**: 1'b1 -> Enable Mission Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM4_ECC_SEC 1'b0 -> Disable Mission Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM4_ECC_SEC **0** = DISABLE **1** = ENABLE
26	0x1	**ERR26**: 1'b1 -> Enable Mission Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM3_ECC_DED 1'b0 -> Disable Mission Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM3_ECC_DED **0** = DISABLE **1** = ENABLE
25	0x1	**ERR25**: 1'b1 -> Enable Mission Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM3_ECC_SEC 1'b0 -> Disable Mission Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM3_ECC_SEC **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
24	0x1	**ERR24**: 1'b1 -> Enable Mission Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM2_ECC_DED 1'b0 -> Disable Mission Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM2_ECC_DED **0** = DISABLE **1** = ENABLE
23	0x1	**ERR23**: 1'b1 -> Enable Mission Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM2_ECC_SEC 1'b0 -> Disable Mission Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM2_ECC_SEC **0** = DISABLE **1** = ENABLE
22	0x1	**ERR22**: 1'b1 -> Enable Mission Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM1_ECC_DED 1'b0 -> Disable Mission Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM1_ECC_DED **0** = DISABLE **1** = ENABLE
21	0x1	**ERR21**: 1'b1 -> Enable Mission Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM1_ECC_SEC 1'b0 -> Disable Mission Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM1_ECC_SEC **0** = DISABLE **1** = ENABLE
20	0x1	**ERR20**: 1'b1 -> Enable Mission Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM0_ECC_DED 1'b0 -> Disable Mission Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM0_ECC_DED **0** = DISABLE **1** = ENABLE
19	0x1	**ERR19**: 1'b1 -> Enable Mission Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM0_ECC_SEC 1'b0 -> Disable Mission Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM0_ECC_SEC **0** = DISABLE **1** = ENABLE
18	0x1	**ERR18**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from NV_vi.u_csimux.u_reg 1'b0 -> Disable Mission Error Reporting for Register Parity Error from NV_vi.u_csimux.u_reg **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
17	0x1	**ERR17**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from NV_vi.u_chansel.u_shadow_reg 1'b0 -> Disable Mission Error Reporting for Register Parity Error from NV_vi.u_chansel.u_shadow_reg **0** = DISABLE **1** = ENABLE
16	0x1	**ERR16**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from NV_vi.u_chansel.u_direct_reg 1'b0 -> Disable Mission Error Reporting for Register Parity Error from NV_vi.u_chansel.u_direct_reg **0** = DISABLE **1** = ENABLE
15	0x1	**ERR15**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from NV_vi.u_atomp.u_reg_stg0 1'b0 -> Disable Mission Error Reporting for Register Parity Error from NV_vi.u_atomp.u_reg_stg0 **0** = DISABLE **1** = ENABLE
14	0x1	**ERR14**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from NV_vi.u_pixfmt.u_cfgreg 1'b0 -> Disable Mission Error Reporting for Register Parity Error from NV_vi.u_pixfmt.u_cfgreg **0** = DISABLE **1** = ENABLE
13	0x1	**ERR13**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from NV_vi.u_pixfmt.u_reg 1'b0 -> Disable Mission Error Reporting for Register Parity Error from NV_vi.u_pixfmt.u_reg **0** = DISABLE **1** = ENABLE
12	0x1	**ERR12**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from NV_vi.u_csbmaster.u_chanreg 1'b0 -> Disable Mission Error Reporting for Register Parity Error from NV_vi.u_csbmaster.u_chanreg **0** = DISABLE **1** = ENABLE
11	0x1	**ERR11**: 1'b1 -> Enable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_NOTIFY_TIMESTAMP_OVRFLOW_ERR 1'b0 -> Disable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_NOTIFY_TIMESTAMP_OVRFLOW_ERR **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
10	0x1	**ERR10**: 1'b1 -> Enable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_HOST_PKTINJECT_STALL_ERR 1'b0 -> Disable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_HOST_PKTINJECT_STALL_ERR **0** = DISABLE **1** = ENABLE
9	0x1	**ERR9**: 1'b1 -> Enable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_CSIMUX_FIFO_OVFL_ERR 1'b0 -> Disable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_CSIMUX_FIFO_OVFL_ERR **0** = DISABLE **1** = ENABLE
8	0x1	**ERR8**: 1'b1 -> Enable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_PACKER_OVFL_ERR 1'b0 -> Disable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_PACKER_OVFL_ERR **0** = DISABLE **1** = ENABLE
7	0x1	**ERR7**: 1'b1 -> Enable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_NOTIFY_FIFO_OVFL_ERR 1'b0 -> Disable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_NOTIFY_FIFO_OVFL_ERR **0** = DISABLE **1** = ENABLE
6	0x1	**ERR6**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from NV_VIFALCON_falcon.cfgregs.ptgt 1'b0 -> Disable Mission Error Reporting for Register Parity Error from NV_VIFALCON_falcon.cfgregs.ptgt **0** = DISABLE **1** = ENABLE
5	0x1	**ERR5**: 1'b1 -> Enable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_FALCON2VI_OTHER_HW_UNCORRECTABLE_ERR 1'b0 -> Disable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_FALCON2VI_OTHER_HW_UNCORRECTABLE_ERR **0** = DISABLE **1** = ENABLE
4	0x1	**ERR4**: 1'b1 -> Enable Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_FALCON2VI_OTHER_HW_CORRECTABLE_ERR 1'b0 -> Disable Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_FALCON2VI_OTHER_HW_CORRECTABLE_ERR **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
3	0x1	**ERR3**: 1'b1 -> Enable Mission Error Reporting for Parity Error from on-chip SRAM/ FIFO from NV_vi.u_NV_VIFALCON_falcon.dmem 1'b0 -> Disable Mission Error Reporting for Parity Error from on-chip SRAM/ FIFO from NV_vi.u_NV_VIFALCON_falcon.dmem **0** = DISABLE **1** = ENABLE
2	0x1	**ERR2**: 1'b1 -> Enable Mission Error Reporting for ECC DED Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.imem 1'b0 -> Disable Mission Error Reporting for ECC DED Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.imem **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Mission Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.imem 1'b0 -> Disable Mission Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.imem **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from NV_VI_err_collator 1'b0 -> Disable Mission Error Reporting for Register Parity Error from NV_VI_err_collator **0** = DISABLE **1** = ENABLE

VI_EC_ERRSLICE0_MISSIONERR_FORCE_0

Offset: 0xc0d
Byte Offset: 0x3034
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**ERR31**: 1'b1 -> Force Assertion of Mission Error Reporting for Parity Error on internal data path from FAULT_UDF_TAG_VI_CHANSEL_DATA_PARITY_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
30	0x0	**ERR30**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM5_ECC_DED 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
29	0x0	**ERR29**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM5_ECC_SEC 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
28	0x0	**ERR28**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM4_ECC_DED 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
27	0x0	**ERR27**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM4_ECC_SEC 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
26	0x0	**ERR26**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM3_ECC_DED 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
25	0x0	**ERR25**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM3_ECC_SEC 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
24	0x0	**ERR24**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM2_ECC_DED 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
23	0x0	**ERR23**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM2_ECC_SEC 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
22	0x0	**ERR22**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM1_ECC_DED 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
21	0x0	**ERR21**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM1_ECC_SEC 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
20	0x0	**ERR20**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM0_ECC_DED 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
19	0x0	**ERR19**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM0_ECC_SEC 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
18	0x0	**ERR18**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from NV_vi.u_csimux.u_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
17	0x0	**ERR17**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from NV_vi.u_chansel.u_shadow_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
16	0x0	**ERR16**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from NV_vi.u_chansel.u_direct_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
15	0x0	**ERR15**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from NV_vi.u_atomp.u_reg_stg0 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
14	0x0	**ERR14**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from NV_vi.u_pixfmt.u_cfgreg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
13	0x0	**ERR13**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from NV_vi.u_pixfmt.u_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
12	0x0	**ERR12**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from NV_vi.u_csbmaster.u_chanreg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
11	0x0	**ERR11**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_NOTIFY_TIMESTAMP_OVRFLOW_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
10	0x0	**ERR10**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_HOST_PKTINJECT_STALL_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
9	0x0	**ERR9**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_CSIMUX_FIFO_OVFL_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
8	0x0	**ERR8**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_PACKER_OVFL_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
7	0x0	**ERR7**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_NOTIFY_FIFO_OVFL_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
6	0x0	**ERR6**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from NV_VIFALCON_falcon.cfgregs.ptgt 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
5	0x0	**ERR5**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_FALCON2VI_OTHER_HW_UNCORRECTABLE_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
4	0x0	**ERR4**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Corrected Error from FAULT_UDF_TAG_FALCON2VI_OTHER_HW_CORRECTABLE_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
3	0x0	**ERR3**: 1'b1 -> Force Assertion of Mission Error Reporting for Parity Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.dmem 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC DED Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.imem 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.imem 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from NV_VI_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

VI_EC_ERRSLICE0_MISSIONERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of MissionError_Enable register to avoid silent dropping of errors.

Offset: 0xc0e
Byte Offset: 0x3038
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**ERR31**: 1'b1 -> Error_31_pulse[1:0] for Parity Error on internal data path from FAULT_UDF_TAG_VI_CHANSEL_DATA_PARITY_ERR was equal to 2'b10. 1'b0 -> Error_31_pulse[1:0] for Parity Error on internal data path from FAULT_UDF_TAG_VI_CHANSEL_DATA_PARITY_ERR was equal to 2'b01.
30	0x0	**ERR30**: 1'b1 -> Error_30_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM5_ECC_DED was equal to 2'b10. 1'b0 -> Error_30_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM5_ECC_DED was equal to 2'b01.
29	0x0	**ERR29**: 1'b1 -> Error_29_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM5_ECC_SEC was equal to 2'b10. 1'b0 -> Error_29_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM5_ECC_SEC was equal to 2'b01.
28	0x0	**ERR28**: 1'b1 -> Error_28_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM4_ECC_DED was equal to 2'b10. 1'b0 -> Error_28_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM4_ECC_DED was equal to 2'b01.

Bit	Reset	Description
27	0x0	**ERR27**: 1'b1 -> Error_27_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM4_ECC_SEC was equal to 2'b10. 1'b0 -> Error_27_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM4_ECC_SEC was equal to 2'b01.
26	0x0	**ERR26**: 1'b1 -> Error_26_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM3_ECC_DED was equal to 2'b10. 1'b0 -> Error_26_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM3_ECC_DED was equal to 2'b01.
25	0x0	**ERR25**: 1'b1 -> Error_25_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM3_ECC_SEC was equal to 2'b10. 1'b0 -> Error_25_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM3_ECC_SEC was equal to 2'b01.
24	0x0	**ERR24**: 1'b1 -> Error_24_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM2_ECC_DED was equal to 2'b10. 1'b0 -> Error_24_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM2_ECC_DED was equal to 2'b01.
23	0x0	**ERR23**: 1'b1 -> Error_23_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM2_ECC_SEC was equal to 2'b10. 1'b0 -> Error_23_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM2_ECC_SEC was equal to 2'b01.
22	0x0	**ERR22**: 1'b1 -> Error_22_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM1_ECC_DED was equal to 2'b10. 1'b0 -> Error_22_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM1_ECC_DED was equal to 2'b01.
21	0x0	**ERR21**: 1'b1 -> Error_21_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM1_ECC_SEC was equal to 2'b10. 1'b0 -> Error_21_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM1_ECC_SEC was equal to 2'b01.
20	0x0	**ERR20**: 1'b1 -> Error_20_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM0_ECC_DED was equal to 2'b10. 1'b0 -> Error_20_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM0_ECC_DED was equal to 2'b01.
19	0x0	**ERR19**: 1'b1 -> Error_19_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM0_ECC_SEC was equal to 2'b10. 1'b0 -> Error_19_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM0_ECC_SEC was equal to 2'b01.
18	0x0	**ERR18**: 1'b1 -> Error_18_pulse[1:0] for Register Parity Error from NV_vi.u_csimux.u_reg was equal to 2'b10. 1'b0 -> Error_18_pulse[1:0] for Register Parity Error from NV_vi.u_csimux.u_reg was equal to 2'b01.

Bit	Reset	Description
17	0x0	**ERR17**: 1'b1 -> Error_17_pulse[1:0] for Register Parity Error from NV_vi.u_chansel.u_shadow_reg was equal to 2'b10. 1'b0 -> Error_17_pulse[1:0] for Register Parity Error from NV_vi.u_chansel.u_shadow_reg was equal to 2'b01.
16	0x0	**ERR16**: 1'b1 -> Error_16_pulse[1:0] for Register Parity Error from NV_vi.u_chansel.u_direct_reg was equal to 2'b10. 1'b0 -> Error_16_pulse[1:0] for Register Parity Error from NV_vi.u_chansel.u_direct_reg was equal to 2'b01.
15	0x0	**ERR15**: 1'b1 -> Error_15_pulse[1:0] for Register Parity Error from NV_vi.u_atomp.u_reg_stg0 was equal to 2'b10. 1'b0 -> Error_15_pulse[1:0] for Register Parity Error from NV_vi.u_atomp.u_reg_stg0 was equal to 2'b01.
14	0x0	**ERR14**: 1'b1 -> Error_14_pulse[1:0] for Register Parity Error from NV_vi.u_pixfmt.u_cfgreg was equal to 2'b10. 1'b0 -> Error_14_pulse[1:0] for Register Parity Error from NV_vi.u_pixfmt.u_cfgreg was equal to 2'b01.
13	0x0	**ERR13**: 1'b1 -> Error_13_pulse[1:0] for Register Parity Error from NV_vi.u_pixfmt.u_reg was equal to 2'b10. 1'b0 -> Error_13_pulse[1:0] for Register Parity Error from NV_vi.u_pixfmt.u_reg was equal to 2'b01.
12	0x0	**ERR12**: 1'b1 -> Error_12_pulse[1:0] for Register Parity Error from NV_vi.u_csbmaster.u_chanreg was equal to 2'b10. 1'b0 -> Error_12_pulse[1:0] for Register Parity Error from NV_vi.u_csbmaster.u_chanreg was equal to 2'b01.
11	0x0	**ERR11**: 1'b1 -> Error_11_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_NOTIFY_TIMESTAMP_OVRFLOW_ERR was equal to 2'b10. 1'b0 -> Error_11_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_NOTIFY_TIMESTAMP_OVRFLOW_ERR was equal to 2'b01.
10	0x0	**ERR10**: 1'b1 -> Error_10_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_HOST_PKTINJECT_STALL_ERR was equal to 2'b10. 1'b0 -> Error_10_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_HOST_PKTINJECT_STALL_ERR was equal to 2'b01.
9	0x0	**ERR9**: 1'b1 -> Error_9_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_CSIMUX_FIFO_OVFL_ERR was equal to 2'b10. 1'b0 -> Error_9_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_CSIMUX_FIFO_OVFL_ERR was equal to 2'b01.
8	0x0	**ERR8**: 1'b1 -> Error_8_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_PACKER_OVFL_ERR was equal to 2'b10. 1'b0 -> Error_8_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_PACKER_OVFL_ERR was equal to 2'b01.

Bit	Reset	Description
7	0x0	**ERR7**: 1'b1 -> Error_7_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_NOTIFY_FIFO_OVFL_ERR was equal to 2'b10. 1'b0 -> Error_7_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_NOTIFY_FIFO_OVFL_ERR was equal to 2'b01.
6	0x0	**ERR6**: 1'b1 -> Error_6_pulse[1:0] for Register Parity Error from NV_VIFALCON_falcon.cfgregs.ptgt was equal to 2'b10. 1'b0 -> Error_6_pulse[1:0] for Register Parity Error from NV_VIFALCON_falcon.cfgregs.ptgt was equal to 2'b01.
5	0x0	**ERR5**: 1'b1 -> Error_5_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_FALCON2VI_OTHER_HW_UNCORRECTABLE_ERR was equal to 2'b10. 1'b0 -> Error_5_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_FALCON2VI_OTHER_HW_UNCORRECTABLE_ERR was equal to 2'b01.
4	0x0	**ERR4**: 1'b1 -> Error_4_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_FALCON2VI_OTHER_HW_CORRECTABLE_ERR was equal to 2'b10. 1'b0 -> Error_4_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_FALCON2VI_OTHER_HW_CORRECTABLE_ERR was equal to 2'b01.
3	0x0	**ERR3**: 1'b1 -> Error_3_pulse[1:0] for Parity Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.dmem was equal to 2'b10. 1'b0 -> Error_3_pulse[1:0] for Parity Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.dmem was equal to 2'b01.
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for ECC DED Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.imem was equal to 2'b10. 1'b0 -> Error_2_pulse[1:0] for ECC DED Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.imem was equal to 2'b01.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for ECC SEC Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.imem was equal to 2'b10. 1'b0 -> Error_1_pulse[1:0] for ECC SEC Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.imem was equal to 2'b01.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from NV_VI_err_collator was equal to 2'b10. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from NV_VI_err_collator was equal to 2'b01.

VI_EC_ERRSLICE0_MISSIONERR_INJECT_0

Offset: 0xc0f
Byte Offset: 0x303c
Read/Write: R/W

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,00xx,0000)

Bit	Reset	Description
31	0x0	**ERR31**: 1'b1 -> Assert the inject_error_31 output for Parity Error on internal data path to FAULT_UDF_TAG_VI_CHANSEL_DATA_PARITY_ERR to allow for error injection. 1'b0 -> De-Asseert inject_error_31 output. **0** = DISABLE **1** = ENABLE
30	0x0	**ERR30**: 1'b1 -> Assert the inject_error_30 output for ECC DED Error on internal data path to FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM5_ECC_DED to allow for error injection. 1'b0 -> De-Asseert inject_error_30 output. **0** = DISABLE **1** = ENABLE
29	0x0	**ERR29**: 1'b1 -> Assert the inject_error_29 output for ECC SEC Error on internal data path to FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM5_ECC_SEC to allow for error injection. 1'b0 -> De-Asseert inject_error_29 output. **0** = DISABLE **1** = ENABLE
28	0x0	**ERR28**: 1'b1 -> Assert the inject_error_28 output for ECC DED Error on internal data path to FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM4_ECC_DED to allow for error injection. 1'b0 -> De-Asseert inject_error_28 output. **0** = DISABLE **1** = ENABLE
27	0x0	**ERR27**: 1'b1 -> Assert the inject_error_27 output for ECC SEC Error on internal data path to FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM4_ECC_SEC to allow for error injection. 1'b0 -> De-Asseert inject_error_27 output. **0** = DISABLE **1** = ENABLE
26	0x0	**ERR26**: 1'b1 -> Assert the inject_error_26 output for ECC DED Error on internal data path to FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM3_ECC_DED to allow for error injection. 1'b0 -> De-Asseert inject_error_26 output. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
25	0x0	**ERR25**: 1'b1 -> Assert the inject_error_25 output for ECC SEC Error on internal data path to FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM3_ECC_SEC to allow for error injection. 1'b0 -> De-Asseert inject_error_25 output. **0** = DISABLE **1** = ENABLE
24	0x0	**ERR24**: 1'b1 -> Assert the inject_error_24 output for ECC DED Error on internal data path to FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM2_ECC_DED to allow for error injection. 1'b0 -> De-Asseert inject_error_24 output. **0** = DISABLE **1** = ENABLE
23	0x0	**ERR23**: 1'b1 -> Assert the inject_error_23 output for ECC SEC Error on internal data path to FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM2_ECC_SEC to allow for error injection. 1'b0 -> De-Asseert inject_error_23 output. **0** = DISABLE **1** = ENABLE
22	0x0	**ERR22**: 1'b1 -> Assert the inject_error_22 output for ECC DED Error on internal data path to FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM1_ECC_DED to allow for error injection. 1'b0 -> De-Asseert inject_error_22 output. **0** = DISABLE **1** = ENABLE
21	0x0	**ERR21**: 1'b1 -> Assert the inject_error_21 output for ECC SEC Error on internal data path to FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM1_ECC_SEC to allow for error injection. 1'b0 -> De-Asseert inject_error_21 output. **0** = DISABLE **1** = ENABLE
20	0x0	**ERR20**: 1'b1 -> Assert the inject_error_20 output for ECC DED Error on internal data path to FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM0_ECC_DED to allow for error injection. 1'b0 -> De-Asseert inject_error_20 output. **0** = DISABLE **1** = ENABLE
19	0x0	**ERR19**: 1'b1 -> Assert the inject_error_19 output for ECC SEC Error on internal data path to FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM0_ECC_SEC to allow for error injection. 1'b0 -> De-Asseert inject_error_19 output. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
18	0x0	**ERR18**: 1'b1 -> Assert the inject_error_18 output for Register Parity Error to NV_vi.u_csimux.u_reg to allow for error injection. 1'b0 -> De-Asseert inject_error_18 output. **0** = DISABLE **1** = ENABLE
17	0x0	**ERR17**: 1'b1 -> Assert the inject_error_17 output for Register Parity Error to NV_vi.u_chansel.u_shadow_reg to allow for error injection. 1'b0 -> De-Asseert inject_error_17 output. **0** = DISABLE **1** = ENABLE
16	0x0	**ERR16**: 1'b1 -> Assert the inject_error_16 output for Register Parity Error to NV_vi.u_chansel.u_direct_reg to allow for error injection. 1'b0 -> De-Asseert inject_error_16 output. **0** = DISABLE **1** = ENABLE
15	0x0	**ERR15**: 1'b1 -> Assert the inject_error_15 output for Register Parity Error to NV_vi.u_atomp.u_reg_stg0 to allow for error injection. 1'b0 -> De-Asseert inject_error_15 output. **0** = DISABLE **1** = ENABLE
14	0x0	**ERR14**: 1'b1 -> Assert the inject_error_14 output for Register Parity Error to NV_vi.u_pixfmt.u_cfgreg to allow for error injection. 1'b0 -> De-Asseert inject_error_14 output. **0** = DISABLE **1** = ENABLE
13	0x0	**ERR13**: 1'b1 -> Assert the inject_error_13 output for Register Parity Error to NV_vi.u_pixfmt.u_reg to allow for error injection. 1'b0 -> De-Asseert inject_error_13 output. **0** = DISABLE **1** = ENABLE
12	0x0	**ERR12**: 1'b1 -> Assert the inject_error_12 output for Register Parity Error to NV_vi.u_csbmaster.u_chanreg to allow for error injection. 1'b0 -> De-Asseert inject_error_12 output. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
11	0x0	**ERR11**: 1'b1 -> Assert the inject_error_11 output for Other Uncorrected Error to FAULT_UDF_TAG_VI_FATAL_NOTIFY_TIMESTAMP_OVRFLOW_ERR to allow for error injection. 1'b0 -> De-Asseert inject_error_11 output. **0** = DISABLE **1** = ENABLE
10	0x0	**ERR10**: 1'b1 -> Assert the inject_error_10 output for Other Uncorrected Error to FAULT_UDF_TAG_VI_FATAL_HOST_PKTINJECT_STALL_ERR to allow for error injection. 1'b0 -> De-Asseert inject_error_10 output. **0** = DISABLE **1** = ENABLE
9	0x0	**ERR9**: 1'b1 -> Assert the inject_error_9 output for Other Uncorrected Error to FAULT_UDF_TAG_VI_FATAL_CSIMUX_FIFO_OVFL_ERR to allow for error injection. 1'b0 -> De-Asseert inject_error_9 output. **0** = DISABLE **1** = ENABLE
8	0x0	**ERR8**: 1'b1 -> Assert the inject_error_8 output for Other Uncorrected Error to FAULT_UDF_TAG_VI_FATAL_ATOMP_PACKER_OVFL_ERR to allow for error injection. 1'b0 -> De-Asseert inject_error_8 output. **0** = DISABLE **1** = ENABLE
7	0x0	**ERR7**: 1'b1 -> Assert the inject_error_7 output for Other Uncorrected Error to FAULT_UDF_TAG_VI_FATAL_NOTIFY_FIFO_OVFL_ERR to allow for error injection. 1'b0 -> De-Asseert inject_error_7 output. **0** = DISABLE **1** = ENABLE
6	0x0	**ERR6**: 1'b1 -> Assert the inject_error_6 output for Register Parity Error to NV_VIFALCON_falcon.cfgregs.ptgt to allow for error injection. 1'b0 -> De-Asseert inject_error_6 output. **0** = DISABLE **1** = ENABLE
3	0x0	**ERR3**: 1'b1 -> Assert the inject_error_3 output for Parity Error from on-chip SRAM/ FIFO to NV_vi.u_NV_VIFALCON_falcon.dmem to allow for error injection. 1'b0 -> De-Asseert inject_error_3 output. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Assert the inject_error_2 output for ECC DED Error from on-chip SRAM/FIFO to NV_vi.u_NV_VIFALCON_falcon.imem to allow for error injection. 1'b0 -> De-Asseert inject_error_2 output. **0** = DISABLE **1** = ENABLE
1	0x0	**ERR1**: 1'b1 -> Assert the inject_error_1 output for ECC SEC Error from on-chip SRAM/FIFO to NV_vi.u_NV_VIFALCON_falcon.imem to allow for error injection. 1'b0 -> De-Asseert inject_error_1 output. **0** = DISABLE **1** = ENABLE
0	0x0	**ERR0**: 1'b1 -> Assert the inject_error_0 output for Register Parity Error to NV_VI_err_collator to allow for error injection. 1'b0 -> De-Asseert inject_error_0 output. **0** = DISABLE **1** = ENABLE

VI_EC_ERRSLICE0_LATENTERR_ENABLE_0

Offset: 0xc10
Byte Offset: 0x3040
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31	0x1	**ERR31**: 1'b1 -> Enable Latent Error Reporting for Parity Error on internal data path from FAULT_UDF_TAG_VI_CHANSEL_DATA_PARITY_ERR 1'b0 -> Disable Latent Error Reporting for Parity Error on internal data path from FAULT_UDF_TAG_VI_CHANSEL_DATA_PARITY_ERR **0** = DISABLE **1** = ENABLE
30	0x1	**ERR30**: 1'b1 -> Enable Latent Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM5_ECC_DED 1'b0 -> Disable Latent Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM5_ECC_DED **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
29	0x1	**ERR29**: 1'b1 -> Enable Latent Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM5_ECC_SEC 1'b0 -> Disable Latent Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM5_ECC_SEC **0** = DISABLE **1** = ENABLE
28	0x1	**ERR28**: 1'b1 -> Enable Latent Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM4_ECC_DED 1'b0 -> Disable Latent Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM4_ECC_DED **0** = DISABLE **1** = ENABLE
27	0x1	**ERR27**: 1'b1 -> Enable Latent Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM4_ECC_SEC 1'b0 -> Disable Latent Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM4_ECC_SEC **0** = DISABLE **1** = ENABLE
26	0x1	**ERR26**: 1'b1 -> Enable Latent Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM3_ECC_DED 1'b0 -> Disable Latent Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM3_ECC_DED **0** = DISABLE **1** = ENABLE
25	0x1	**ERR25**: 1'b1 -> Enable Latent Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM3_ECC_SEC 1'b0 -> Disable Latent Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM3_ECC_SEC **0** = DISABLE **1** = ENABLE
24	0x1	**ERR24**: 1'b1 -> Enable Latent Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM2_ECC_DED 1'b0 -> Disable Latent Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM2_ECC_DED **0** = DISABLE **1** = ENABLE
23	0x1	**ERR23**: 1'b1 -> Enable Latent Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM2_ECC_SEC 1'b0 -> Disable Latent Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM2_ECC_SEC **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
22	0x1	**ERR22**: 1'b1 -> Enable Latent Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM1_ECC_DED 1'b0 -> Disable Latent Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM1_ECC_DED **0** = DISABLE **1** = ENABLE
21	0x1	**ERR21**: 1'b1 -> Enable Latent Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM1_ECC_SEC 1'b0 -> Disable Latent Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM1_ECC_SEC **0** = DISABLE **1** = ENABLE
20	0x1	**ERR20**: 1'b1 -> Enable Latent Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM0_ECC_DED 1'b0 -> Disable Latent Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM0_ECC_DED **0** = DISABLE **1** = ENABLE
19	0x1	**ERR19**: 1'b1 -> Enable Latent Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM0_ECC_SEC 1'b0 -> Disable Latent Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM0_ECC_SEC **0** = DISABLE **1** = ENABLE
18	0x1	**ERR18**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from NV_vi.u_csimux.u_reg 1'b0 -> Disable Latent Error Reporting for Register Parity Error from NV_vi.u_csimux.u_reg **0** = DISABLE **1** = ENABLE
17	0x1	**ERR17**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from NV_vi.u_chansel.u_shadow_reg 1'b0 -> Disable Latent Error Reporting for Register Parity Error from NV_vi.u_chansel.u_shadow_reg **0** = DISABLE **1** = ENABLE
16	0x1	**ERR16**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from NV_vi.u_chansel.u_direct_reg 1'b0 -> Disable Latent Error Reporting for Register Parity Error from NV_vi.u_chansel.u_direct_reg **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
15	0x1	**ERR15**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from NV_vi.u_atomp.u_reg_stg0 1'b0 -> Disable Latent Error Reporting for Register Parity Error from NV_vi.u_atomp.u_reg_stg0 **0** = DISABLE **1** = ENABLE
14	0x1	**ERR14**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from NV_vi.u_pixfmt.u_cfgreg 1'b0 -> Disable Latent Error Reporting for Register Parity Error from NV_vi.u_pixfmt.u_cfgreg **0** = DISABLE **1** = ENABLE
13	0x1	**ERR13**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from NV_vi.u_pixfmt.u_reg 1'b0 -> Disable Latent Error Reporting for Register Parity Error from NV_vi.u_pixfmt.u_reg **0** = DISABLE **1** = ENABLE
12	0x1	**ERR12**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from NV_vi.u_csbmaster.u_chanreg 1'b0 -> Disable Latent Error Reporting for Register Parity Error from NV_vi.u_csbmaster.u_chanreg **0** = DISABLE **1** = ENABLE
11	0x1	**ERR11**: 1'b1 -> Enable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_NOTIFY_TIMESTAMP_OVRFLOW_ERR 1'b0 -> Disable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_NOTIFY_TIMESTAMP_OVRFLOW_ERR **0** = DISABLE **1** = ENABLE
10	0x1	**ERR10**: 1'b1 -> Enable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_HOST_PKTINJECT_STALL_ERR 1'b0 -> Disable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_HOST_PKTINJECT_STALL_ERR **0** = DISABLE **1** = ENABLE
9	0x1	**ERR9**: 1'b1 -> Enable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_CSIMUX_FIFO_OVFL_ERR 1'b0 -> Disable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_CSIMUX_FIFO_OVFL_ERR **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
8	0x1	**ERR8**: 1'b1 -> Enable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_PACKER_OVFL_ERR 1'b0 -> Disable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_PACKER_OVFL_ERR **0** = DISABLE **1** = ENABLE
7	0x1	**ERR7**: 1'b1 -> Enable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_NOTIFY_FIFO_OVFL_ERR 1'b0 -> Disable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_NOTIFY_FIFO_OVFL_ERR **0** = DISABLE **1** = ENABLE
6	0x1	**ERR6**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from NV_VIFALCON_falcon.cfgregs.ptgt 1'b0 -> Disable Latent Error Reporting for Register Parity Error from NV_VIFALCON_falcon.cfgregs.ptgt **0** = DISABLE **1** = ENABLE
5	0x1	**ERR5**: 1'b1 -> Enable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_FALCON2VI_OTHER_HW_UNCORRECTABLE_ERR 1'b0 -> Disable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_FALCON2VI_OTHER_HW_UNCORRECTABLE_ERR **0** = DISABLE **1** = ENABLE
4	0x1	**ERR4**: 1'b1 -> Enable Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_FALCON2VI_OTHER_HW_CORRECTABLE_ERR 1'b0 -> Disable Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_FALCON2VI_OTHER_HW_CORRECTABLE_ERR **0** = DISABLE **1** = ENABLE
3	0x1	**ERR3**: 1'b1 -> Enable Latent Error Reporting for Parity Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.dmem 1'b0 -> Disable Latent Error Reporting for Parity Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.dmem **0** = DISABLE **1** = ENABLE
2	0x1	**ERR2**: 1'b1 -> Enable Latent Error Reporting for ECC DED Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.imem 1'b0 -> Disable Latent Error Reporting for ECC DED Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.imem **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
1	0x1	**ERR1**: 1'b1 -> Enable Latent Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.imem 1'b0 -> Disable Latent Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.imem **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from NV_VI_err_collator 1'b0 -> Disable Latent Error Reporting for Register Parity Error from NV_VI_err_collator **0** = DISABLE **1** = ENABLE

VI_EC_ERRSLICE0_LATENTERR_FORCE_0

Offset: 0xc11
Byte Offset: 0x3044
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**ERR31**: 1'b1 -> Force Assertion of Latent Error Reporting for Parity Error on internal data path from FAULT_UDF_TAG_VI_CHANSEL_DATA_PARITY_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
30	0x0	**ERR30**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM5_ECC_DED 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
29	0x0	**ERR29**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM5_ECC_SEC 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
28	0x0	**ERR28**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM4_ECC_DED 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
27	0x0	**ERR27**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM4_ECC_SEC 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
26	0x0	**ERR26**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM3_ECC_DED 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
25	0x0	**ERR25**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM3_ECC_SEC 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
24	0x0	**ERR24**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM2_ECC_DED 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
23	0x0	**ERR23**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM2_ECC_SEC 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
22	0x0	**ERR22**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM1_ECC_DED 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
21	0x0	**ERR21**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM1_ECC_SEC 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
20	0x0	**ERR20**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM0_ECC_DED 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
19	0x0	**ERR19**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM0_ECC_SEC 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
18	0x0	**ERR18**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from NV_vi.u_csimux.u_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
17	0x0	**ERR17**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from NV_vi.u_chansel.u_shadow_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
16	0x0	**ERR16**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from NV_vi.u_chansel.u_direct_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
15	0x0	**ERR15**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from NV_vi.u_atomp.u_reg_stg0 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
14	0x0	**ERR14**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from NV_vi.u_pixfmt.u_cfgreg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
13	0x0	**ERR13**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from NV_vi.u_pixfmt.u_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
12	0x0	**ERR12**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from NV_vi.u_csbmaster.u_chanreg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
11	0x0	**ERR11**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_NOTIFY_TIMESTAMP_OVRFLOW_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
10	0x0	**ERR10**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_HOST_PKTINJECT_STALL_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
9	0x0	**ERR9**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_CSIMUX_FIFO_OVFL_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
8	0x0	**ERR8**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_PACKER_OVFL_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
7	0x0	**ERR7**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_NOTIFY_FIFO_OVFL_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
6	0x0	**ERR6**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from NV_VIFALCON_falcon.cfgregs.ptgt 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
5	0x0	**ERR5**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_FALCON2VI_OTHER_HW_UNCORRECTABLE_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
4	0x0	**ERR4**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Corrected Error from FAULT_UDF_TAG_FALCON2VI_OTHER_HW_CORRECTABLE_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
3	0x0	**ERR3**: 1'b1 -> Force Assertion of Latent Error Reporting for Parity Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.dmem 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC DED Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.imem 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.imem 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from NV_VI_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

VI_EC_ERRSLICE0_LATENTERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of LatentError_Enable register, to avoid silent dropping of errors.

Offset: 0xc12
Byte Offset: 0x3048
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**ERR31**: 1'b1 -> Error_31_pulse[1:0] for Parity Error on internal data path from FAULT_UDF_TAG_VI_CHANSEL_DATA_PARITY_ERR was equal to 2'b00 or 2'b11. 1'b0 -> Error_31_pulse[1:0] for Parity Error on internal data path from FAULT_UDF_TAG_VI_CHANSEL_DATA_PARITY_ERR was equal to 2'b01 or 2'b10, but no latent error.
30	0x0	**ERR30**: 1'b1 -> Error_30_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM5_ECC_DED was equal to 2'b00 or 2'b11. 1'b0 -> Error_30_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM5_ECC_DED was equal to 2'b01 or 2'b10, but no latent error.
29	0x0	**ERR29**: 1'b1 -> Error_29_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM5_ECC_SEC was equal to 2'b00 or 2'b11. 1'b0 -> Error_29_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM5_ECC_SEC was equal to 2'b01 or 2'b10, but no latent error.
28	0x0	**ERR28**: 1'b1 -> Error_28_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM4_ECC_DED was equal to 2'b00 or 2'b11. 1'b0 -> Error_28_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM4_ECC_DED was equal to 2'b01 or 2'b10, but no latent error.
27	0x0	**ERR27**: 1'b1 -> Error_27_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM4_ECC_SEC was equal to 2'b00 or 2'b11. 1'b0 -> Error_27_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM4_ECC_SEC was equal to 2'b01 or 2'b10, but no latent error.
26	0x0	**ERR26**: 1'b1 -> Error_26_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM3_ECC_DED was equal to 2'b00 or 2'b11. 1'b0 -> Error_26_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM3_ECC_DED was equal to 2'b01 or 2'b10, but no latent error.

Bit	Reset	Description
25	0x0	**ERR25**: 1'b1 -> Error_25_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM3_ECC_SEC was equal to 2'b00 or 2'b11. 1'b0 -> Error_25_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM3_ECC_SEC was equal to 2'b01 or 2'b10, but no latent error.
24	0x0	**ERR24**: 1'b1 -> Error_24_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM2_ECC_DED was equal to 2'b00 or 2'b11. 1'b0 -> Error_24_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM2_ECC_DED was equal to 2'b01 or 2'b10, but no latent error.
23	0x0	**ERR23**: 1'b1 -> Error_23_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM2_ECC_SEC was equal to 2'b00 or 2'b11. 1'b0 -> Error_23_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM2_ECC_SEC was equal to 2'b01 or 2'b10, but no latent error.
22	0x0	**ERR22**: 1'b1 -> Error_22_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM1_ECC_DED was equal to 2'b00 or 2'b11. 1'b0 -> Error_22_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM1_ECC_DED was equal to 2'b01 or 2'b10, but no latent error.
21	0x0	**ERR21**: 1'b1 -> Error_21_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM1_ECC_SEC was equal to 2'b00 or 2'b11. 1'b0 -> Error_21_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM1_ECC_SEC was equal to 2'b01 or 2'b10, but no latent error.
20	0x0	**ERR20**: 1'b1 -> Error_20_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM0_ECC_DED was equal to 2'b00 or 2'b11. 1'b0 -> Error_20_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM0_ECC_DED was equal to 2'b01 or 2'b10, but no latent error.
19	0x0	**ERR19**: 1'b1 -> Error_19_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM0_ECC_SEC was equal to 2'b00 or 2'b11. 1'b0 -> Error_19_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM0_ECC_SEC was equal to 2'b01 or 2'b10, but no latent error.
18	0x0	**ERR18**: 1'b1 -> Error_18_pulse[1:0] for Register Parity Error from NV_vi.u_csimux.u_reg was equal to 2'b00 or 2'b11. 1'b0 -> Error_18_pulse[1:0] for Register Parity Error from NV_vi.u_csimux.u_reg was equal to 2'b01 or 2'b10, but no latent error.

Bit	Reset	Description
17	0x0	**ERR17**: 1'b1 -> Error_17_pulse[1:0] for Register Parity Error from NV_vi.u_chansel.u_shadow_reg was equal to 2'b00 or 2'b11. 1'b0 -> Error_17_pulse[1:0] for Register Parity Error from NV_vi.u_chansel.u_shadow_reg was equal to 2'b01 or 2'b10, but no latent error.
16	0x0	**ERR16**: 1'b1 -> Error_16_pulse[1:0] for Register Parity Error from NV_vi.u_chansel.u_direct_reg was equal to 2'b00 or 2'b11. 1'b0 -> Error_16_pulse[1:0] for Register Parity Error from NV_vi.u_chansel.u_direct_reg was equal to 2'b01 or 2'b10, but no latent error.
15	0x0	**ERR15**: 1'b1 -> Error_15_pulse[1:0] for Register Parity Error from NV_vi.u_atomp.u_reg_stg0 was equal to 2'b00 or 2'b11. 1'b0 -> Error_15_pulse[1:0] for Register Parity Error from NV_vi.u_atomp.u_reg_stg0 was equal to 2'b01 or 2'b10, but no latent error.
14	0x0	**ERR14**: 1'b1 -> Error_14_pulse[1:0] for Register Parity Error from NV_vi.u_pixfmt.u_cfgreg was equal to 2'b00 or 2'b11. 1'b0 -> Error_14_pulse[1:0] for Register Parity Error from NV_vi.u_pixfmt.u_cfgreg was equal to 2'b01 or 2'b10, but no latent error.
13	0x0	**ERR13**: 1'b1 -> Error_13_pulse[1:0] for Register Parity Error from NV_vi.u_pixfmt.u_reg was equal to 2'b00 or 2'b11. 1'b0 -> Error_13_pulse[1:0] for Register Parity Error from NV_vi.u_pixfmt.u_reg was equal to 2'b01 or 2'b10, but no latent error.
12	0x0	**ERR12**: 1'b1 -> Error_12_pulse[1:0] for Register Parity Error from NV_vi.u_csbmaster.u_chanreg was equal to 2'b00 or 2'b11. 1'b0 -> Error_12_pulse[1:0] for Register Parity Error from NV_vi.u_csbmaster.u_chanreg was equal to 2'b01 or 2'b10, but no latent error.
11	0x0	**ERR11**: 1'b1 -> Error_11_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_NOTIFY_TIMESTAMP_OVRFLOW_ERR was equal to 2'b00 or 2'b11. 1'b0 -> Error_11_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_NOTIFY_TIMESTAMP_OVRFLOW_ERR was equal to 2'b01 or 2'b10, but no latent error.
10	0x0	**ERR10**: 1'b1 -> Error_10_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_HOST_PKTINJECT_STALL_ERR was equal to 2'b00 or 2'b11. 1'b0 -> Error_10_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_HOST_PKTINJECT_STALL_ERR was equal to 2'b01 or 2'b10, but no latent error.
9	0x0	**ERR9**: 1'b1 -> Error_9_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_CSIMUX_FIFO_OVFL_ERR was equal to 2'b00 or 2'b11. 1'b0 -> Error_9_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_CSIMUX_FIFO_OVFL_ERR was equal to 2'b01 or 2'b10, but no latent error.

Bit	Reset	Description
8	0x0	**ERR8**: 1'b1 -> Error_8_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_PACKER_OVFL_ERR was equal to 2'b00 or 2'b11. 1'b0 -> Error_8_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_PACKER_OVFL_ERR was equal to 2'b01 or 2'b10, but no latent error.
7	0x0	**ERR7**: 1'b1 -> Error_7_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_NOTIFY_FIFO_OVFL_ERR was equal to 2'b00 or 2'b11. 1'b0 -> Error_7_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_NOTIFY_FIFO_OVFL_ERR was equal to 2'b01 or 2'b10, but no latent error.
6	0x0	**ERR6**: 1'b1 -> Error_6_pulse[1:0] for Register Parity Error from NV_VIFALCON_falcon.cfgregs.ptgt was equal to 2'b00 or 2'b11. 1'b0 -> Error_6_pulse[1:0] for Register Parity Error from NV_VIFALCON_falcon.cfgregs.ptgt was equal to 2'b01 or 2'b10, but no latent error.
5	0x0	**ERR5**: 1'b1 -> Error_5_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_FALCON2VI_OTHER_HW_UNCORRECTABLE_ERR was equal to 2'b00 or 2'b11. 1'b0 -> Error_5_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_FALCON2VI_OTHER_HW_UNCORRECTABLE_ERR was equal to 2'b01 or 2'b10, but no latent error.
4	0x0	**ERR4**: 1'b1 -> Error_4_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_FALCON2VI_OTHER_HW_CORRECTABLE_ERR was equal to 2'b00 or 2'b11. 1'b0 -> Error_4_pulse[1:0] for Other Corrected Error from FAULT_UDF_TAG_FALCON2VI_OTHER_HW_CORRECTABLE_ERR was equal to 2'b01 or 2'b10, but no latent error.
3	0x0	**ERR3**: 1'b1 -> Error_3_pulse[1:0] for Parity Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.dmem was equal to 2'b00 or 2'b11. 1'b0 -> Error_3_pulse[1:0] for Parity Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.dmem was equal to 2'b01 or 2'b10, but no latent error.
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for ECC DED Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.imem was equal to 2'b00 or 2'b11. 1'b0 -> Error_2_pulse[1:0] for ECC DED Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.imem was equal to 2'b01 or 2'b10, but no latent error.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for ECC SEC Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.imem was equal to 2'b00 or 2'b11. 1'b0 -> Error_1_pulse[1:0] for ECC SEC Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.imem was equal to 2'b01 or 2'b10, but no latent error.

Bit	Reset	Description
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from NV_VI_err_collator was equal to 2'b00 or 2'b11. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from NV_VI_err_collator was equal to 2'b01 or 2'b10, but no latent error.

VI_EC_ERRSLICE0_COUNTER_RELOAD_0

Offset: 0xc14
Byte Offset: 0x3050
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**ERR31**: 1'b1 -> Reload Error Counter for Parity Error on internal data path from FAULT_UDF_TAG_VI_CHANSEL_DATA_PARITY_ERR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
30	0x0	**ERR30**: 1'b1 -> Reload Error Counter for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM5_ECC_DED 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
29	0x0	**ERR29**: 1'b1 -> Reload Error Counter for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM5_ECC_SEC 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
28	0x0	**ERR28**: 1'b1 -> Reload Error Counter for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM4_ECC_DED 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
27	0x0	**ERR27**: 1'b1 -> Reload Error Counter for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM4_ECC_SEC 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

Bit	Reset	Description
26	0x0	**ERR26**: 1'b1 -> Reload Error Counter for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM3_ECC_DED 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
25	0x0	**ERR25**: 1'b1 -> Reload Error Counter for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM3_ECC_SEC 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
24	0x0	**ERR24**: 1'b1 -> Reload Error Counter for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM2_ECC_DED 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
23	0x0	**ERR23**: 1'b1 -> Reload Error Counter for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM2_ECC_SEC 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
22	0x0	**ERR22**: 1'b1 -> Reload Error Counter for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM1_ECC_DED 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
21	0x0	**ERR21**: 1'b1 -> Reload Error Counter for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM1_ECC_SEC 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
20	0x0	**ERR20**: 1'b1 -> Reload Error Counter for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM0_ECC_DED 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
19	0x0	**ERR19**: 1'b1 -> Reload Error Counter for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM0_ECC_SEC 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

Bit	Reset	Description
18	0x0	**ERR18**: 1'b1 -> Reload Error Counter for Register Parity Error from NV_vi.u_csimux.u_reg 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
17	0x0	**ERR17**: 1'b1 -> Reload Error Counter for Register Parity Error from NV_vi.u_chansel.u_shadow_reg 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
16	0x0	**ERR16**: 1'b1 -> Reload Error Counter for Register Parity Error from NV_vi.u_chansel.u_direct_reg 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
15	0x0	**ERR15**: 1'b1 -> Reload Error Counter for Register Parity Error from NV_vi.u_atomp.u_reg_stg0 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
14	0x0	**ERR14**: 1'b1 -> Reload Error Counter for Register Parity Error from NV_vi.u_pixfmt.u_cfgreg 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
13	0x0	**ERR13**: 1'b1 -> Reload Error Counter for Register Parity Error from NV_vi.u_pixfmt.u_reg 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
12	0x0	**ERR12**: 1'b1 -> Reload Error Counter for Register Parity Error from NV_vi.u_csbmaster.u_chanreg 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
11	0x0	**ERR11**: 1'b1 -> Reload Error Counter for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_NOTIFY_TIMESTAMP_OVRFLOW_ERR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

Bit	Reset	Description
10	0x0	**ERR10**: 1'b1 -> Reload Error Counter for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_HOST_PKTINJECT_STALL_ERR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
9	0x0	**ERR9**: 1'b1 -> Reload Error Counter for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_CSIMUX_FIFO_OVFL_ERR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
8	0x0	**ERR8**: 1'b1 -> Reload Error Counter for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_PACKER_OVFL_ERR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
7	0x0	**ERR7**: 1'b1 -> Reload Error Counter for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_NOTIFY_FIFO_OVFL_ERR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
6	0x0	**ERR6**: 1'b1 -> Reload Error Counter for Register Parity Error from NV_VIFALCON_falcon.cfgregs.ptgt 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
5	0x0	**ERR5**: 1'b1 -> Reload Error Counter for Other Uncorrected Error from FAULT_UDF_TAG_FALCON2VI_OTHER_HW_UNCORRECTABLE_ERR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
4	0x0	**ERR4**: 1'b1 -> Reload Error Counter for Other Corrected Error from FAULT_UDF_TAG_FALCON2VI_OTHER_HW_CORRECTABLE_ERR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
3	0x0	**ERR3**: 1'b1 -> Reload Error Counter for Parity Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.dmem 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Reload Error Counter for ECC DED Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.imem 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
1	0x0	**ERR1**: 1'b1 -> Reload Error Counter for ECC SEC Error from on-chip SRAM/FIFO from NV_vi.u_NV_VIFALCON_falcon.imem 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
0	0x0	**ERR0**: 1'b1 -> Reload Error Counter for Register Parity Error from NV_VI_err_collator 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

VI_EC_ERRSLICE0_MISSIONERR_ECC_CORRECTION_DIS_0

Offset: 0xc15
Byte Offset: 0x3054
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxx0x,0x0x,0x0x,0xxx,xxxx,xxxx,xxxx,xx0x)

Bit	Reset	Description
29	0x0	**ERR29**: 1'b1 -> Disables ECC correction in ECC Terminate plugins for FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM5_ECC_SEC and shall report single bit errors over the ded_err_pulse to only allow detection and no correction. 1'b0 -> ECC correction is enabled in ECC terminate plugins for FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM5_ECC_SEC and shall correct single bit errors and report them over sec_err_pulse **0** = DISABLE **1** = ENABLE
27	0x0	**ERR27**: 1'b1 -> Disables ECC correction in ECC Terminate plugins for FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM4_ECC_SEC and shall report single bit errors over the ded_err_pulse to only allow detection and no correction. 1'b0 -> ECC correction is enabled in ECC terminate plugins for FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM4_ECC_SEC and shall correct single bit errors and report them over sec_err_pulse **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
25	0x0	**ERR25**: 1'b1 -> Disables ECC correction in ECC Terminate plugins for FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM3_ECC_SEC and shall report single bit errors over the ded_err_pulse to only allow detection and no correction. 1'b0 -> ECC correction is enabled in ECC terminate plugins for FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM3_ECC_SEC and shall correct single bit errors and report them over sec_err_pulse **0** = DISABLE **1** = ENABLE
23	0x0	**ERR23**: 1'b1 -> Disables ECC correction in ECC Terminate plugins for FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM2_ECC_SEC and shall report single bit errors over the ded_err_pulse to only allow detection and no correction. 1'b0 -> ECC correction is enabled in ECC terminate plugins for FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM2_ECC_SEC and shall correct single bit errors and report them over sec_err_pulse **0** = DISABLE **1** = ENABLE
21	0x0	**ERR21**: 1'b1 -> Disables ECC correction in ECC Terminate plugins for FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM1_ECC_SEC and shall report single bit errors over the ded_err_pulse to only allow detection and no correction. 1'b0 -> ECC correction is enabled in ECC terminate plugins for FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM1_ECC_SEC and shall correct single bit errors and report them over sec_err_pulse **0** = DISABLE **1** = ENABLE
19	0x0	**ERR19**: 1'b1 -> Disables ECC correction in ECC Terminate plugins for FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM0_ECC_SEC and shall report single bit errors over the ded_err_pulse to only allow detection and no correction. 1'b0 -> ECC correction is enabled in ECC terminate plugins for FAULT_UDF_TAG_VI_CSIMUX_CTRL_STREAM0_ECC_SEC and shall correct single bit errors and report them over sec_err_pulse **0** = DISABLE **1** = ENABLE
1	0x0	**ERR1**: 1'b1 -> Disables ECC correction in ECC Terminate plugins for NV_vi.u_NV_VIFALCON_falcon.imem and shall report single bit errors over the ded_err_pulse to only allow detection and no correction. 1'b0 -> ECC correction is enabled in ECC terminate plugins for NV_vi.u_NV_VIFALCON_falcon.imem and shall correct single bit errors and report them over sec_err_pulse **0** = DISABLE **1** = ENABLE

VI_EC_ERRSLICE1_MISSIONERR_ENABLE_0

Offset: 0xc18
Byte Offset: 0x3060
Read/Write: R/W

Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31	0x1	**ERR63**: 1'b1 -> Enable Mission Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR4 1'b0 -> Disable Mission Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR4 **0** = DISABLE **1** = ENABLE
30	0x1	**ERR62**: 1'b1 -> Enable Mission Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR3 1'b0 -> Disable Mission Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR3 **0** = DISABLE **1** = ENABLE
29	0x1	**ERR61**: 1'b1 -> Enable Mission Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR3 1'b0 -> Disable Mission Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR3 **0** = DISABLE **1** = ENABLE
28	0x1	**ERR60**: 1'b1 -> Enable Mission Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR2 1'b0 -> Disable Mission Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR2 **0** = DISABLE **1** = ENABLE
27	0x1	**ERR59**: 1'b1 -> Enable Mission Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR2 1'b0 -> Disable Mission Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR2 **0** = DISABLE **1** = ENABLE
26	0x1	**ERR58**: 1'b1 -> Enable Mission Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR1 1'b0 -> Disable Mission Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR1 **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
25	0x1	**ERR57**: 1'b1 -> Enable Mission Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR1 1'b0 -> Disable Mission Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR1 **0** = DISABLE **1** = ENABLE
24	0x1	**ERR56**: 1'b1 -> Enable Mission Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR0 1'b0 -> Disable Mission Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR0 **0** = DISABLE **1** = ENABLE
23	0x1	**ERR55**: 1'b1 -> Enable Mission Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR0 1'b0 -> Disable Mission Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR0 **0** = DISABLE **1** = ENABLE
22	0x1	**ERR54**: 1'b1 -> Enable Mission Error Reporting for SW Generic Error from NV_vi.u_safety.u_NV_VI_err_collator.sw_err 1'b0 -> Disable Mission Error Reporting for SW Generic Error from NV_vi.u_safety.u_NV_VI_err_collator.sw_err **0** = DISABLE **1** = ENABLE
21	0x1	**ERR53**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from NV_vi.u_csbmaster.u_cfgreg 1'b0 -> Disable Mission Error Reporting for Register Parity Error from NV_vi.u_csbmaster.u_cfgreg **0** = DISABLE **1** = ENABLE
20	0x1	**ERR52**: 1'b1 -> Enable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_CHANNEL_TRUNCATED_ERR 1'b0 -> Disable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_CHANNEL_TRUNCATED_ERR **0** = DISABLE **1** = ENABLE
19	0x1	**ERR51**: 1'b1 -> Enable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_CHANNEL_TOSSED_ERR 1'b0 -> Disable Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_CHANNEL_TOSSED_ERR **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
18	0x1	**ERR50**: 1'b1 -> Enable Mission Error Reporting for Parity Error from on-chip SRAM/ FIFO from FAULT_UDF_TAG_VI_ATOMP_DVFSFIFO_ADR_RAM_PARITY_ERR 1'b0 -> Disable Mission Error Reporting for Parity Error from on-chip SRAM/ FIFO from FAULT_UDF_TAG_VI_ATOMP_DVFSFIFO_ADR_RAM_PARITY_ERR **0** = DISABLE **1** = ENABLE
17	0x1	**ERR49**: 1'b1 -> Enable Mission Error Reporting for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HRD_FIFO_ECC_DED 1'b0 -> Disable Mission Error Reporting for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HRD_FIFO_ECC_DED **0** = DISABLE **1** = ENABLE
16	0x1	**ERR48**: 1'b1 -> Enable Mission Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HRD_FIFO_ECC_SEC 1'b0 -> Disable Mission Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HRD_FIFO_ECC_SEC **0** = DISABLE **1** = ENABLE
15	0x1	**ERR47**: 1'b1 -> Enable Mission Error Reporting for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HWR_FIFO_ECC_DED 1'b0 -> Disable Mission Error Reporting for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HWR_FIFO_ECC_DED **0** = DISABLE **1** = ENABLE
14	0x1	**ERR46**: 1'b1 -> Enable Mission Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HWR_FIFO_ECC_SEC 1'b0 -> Disable Mission Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HWR_FIFO_ECC_SEC **0** = DISABLE **1** = ENABLE
13	0x1	**ERR45**: 1'b1 -> Enable Mission Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_NOTIFY_FIFO_ECC_DED 1'b0 -> Disable Mission Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_NOTIFY_FIFO_ECC_DED **0** = DISABLE **1** = ENABLE
12	0x1	**ERR44**: 1'b1 -> Enable Mission Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_NOTIFY_FIFO_ECC_SEC 1'b0 -> Disable Mission Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_NOTIFY_FIFO_ECC_SEC **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
11	0x1	**ERR43**: 1'b1 -> Enable Mission Error Reporting for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_WR_ACK_FIFO_ECC_DED 1'b0 -> Disable Mission Error Reporting for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_WR_ACK_FIFO_ECC_DED **0** = DISABLE **1** = ENABLE
10	0x1	**ERR42**: 1'b1 -> Enable Mission Error Reporting for ECC SEC Error from on-chip SRAM/FIFOfrom FAULT_UDF_TAG_VI_ATOMP_WR_ACK_FIFO_ECC_SEC 1'b0 -> Disable Mission Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_WR_ACK_FIFO_ECC_SEC **0** = DISABLE **1** = ENABLE
9	0x1	**ERR41**: 1'b1 -> Enable Mission Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_OFIF_FIFO_CTRL_ECC_DED 1'b0 -> Disable Mission Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_OFIF_FIFO_CTRL_ECC_DED **0** = DISABLE **1** = ENABLE
8	0x1	**ERR40**: 1'b1 -> Enable Mission Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_OFIF_FIFO_CTRL_ECC_SEC 1'b0 -> Disable Mission Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_OFIF_FIFO_CTRL_ECC_SEC **0** = DISABLE **1** = ENABLE
7	0x1	**ERR39**: 1'b1 -> Enable Mission Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM5_CRC_ERR 1'b0 -> Disable Mission Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM5_CRC_ERR **0** = DISABLE **1** = ENABLE
6	0x1	**ERR38**: 1'b1 -> Enable Mission Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM4_CRC_ERR 1'b0 -> Disable Mission Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM4_CRC_ERR **0** = DISABLE **1** = ENABLE
5	0x1	**ERR37**: 1'b1 -> Enable Mission Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM3_CRC_ERR 1'b0 -> Disable Mission Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM3_CRC_ERR **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
4	0x1	**ERR36**: 1'b1 -> Enable Mission Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM2_CRC_ERR 1'b0 -> Disable Mission Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM2_CRC_ERR **0** = DISABLE **1** = ENABLE
3	0x1	**ERR35**: 1'b1 -> Enable Mission Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM1_CRC_ERR 1'b0 -> Disable Mission Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM1_CRC_ERR **0** = DISABLE **1** = ENABLE
2	0x1	**ERR34**: 1'b1 -> Enable Mission Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM0_CRC_ERR 1'b0 -> Disable Mission Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM0_CRC_ERR **0** = DISABLE **1** = ENABLE
1	0x1	**ERR33**: 1'b1 -> Enable Mission Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_PIXFMT_CTRL_ECC_DED 1'b0 -> Disable Mission Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_PIXFMT_CTRL_ECC_DED **0** = DISABLE **1** = ENABLE
0	0x1	**ERR32**: 1'b1 -> Enable Mission Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_PIXFMT_CTRL_ECC_SEC 1'b0 -> Disable Mission Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_PIXFMT_CTRL_ECC_SEC **0** = DISABLE **1** = ENABLE

VI_EC_ERRSLICE1_MISSIONERR_FORCE_0

Offset: 0xc19
Byte Offset: 0x3064
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**ERR63**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR4 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
30	0x0	**ERR62**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR3 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
29	0x0	**ERR61**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR3 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
28	0x0	**ERR60**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR2 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
27	0x0	**ERR59**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR2 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
26	0x0	**ERR58**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR1 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
25	0x0	**ERR57**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR1 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
24	0x0	**ERR56**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR0 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
23	0x0	**ERR55**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR0 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
22	0x0	**ERR54**: 1'b1 -> Force Assertion of Mission Error Reporting for SW Generic Error from NV_vi.u_safety.u_NV_VI_err_collator.sw_err 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
21	0x0	**ERR53**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from NV_vi.u_csbmaster.u_cfgreg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
20	0x0	**ERR52**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_CHANNEL_TRUNCATED_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
19	0x0	**ERR51**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_CHANNEL_TOSSED_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
18	0x0	**ERR50**: 1'b1 -> Force Assertion of Mission Error Reporting for Parity Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_DVFSFIFO_ADR_RAM_PARITY_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
17	0x0	**ERR49**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HRD_FIFO_ECC_DED 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
16	0x0	**ERR48**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HRD_FIFO_ECC_SEC 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
15	0x0	**ERR47**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HWR_FIFO_ECC_DED 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
14	0x0	**ERR46**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HWR_FIFO_ECC_SEC 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
13	0x0	**ERR45**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_NOTIFY_FIFO_ECC_DED 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
12	0x0	**ERR44**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_NOTIFY_FIFO_ECC_SEC 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
11	0x0	**ERR43**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_WR_ACK_FIFO_ECC_DED 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
10	0x0	**ERR42**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_WR_ACK_FIFO_ECC_SEC 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
9	0x0	**ERR41**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_OFIF_FIFO_CTRL_ECC_DED 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
8	0x0	**ERR40**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_OFIF_FIFO_CTRL_ECC_SEC 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
7	0x0	**ERR39**: 1'b1 -> Force Assertion of Mission Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM5_CRC_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
6	0x0	**ERR38**: 1'b1 -> Force Assertion of Mission Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM4_CRC_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
5	0x0	**ERR37**: 1'b1 -> Force Assertion of Mission Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM3_CRC_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
4	0x0	**ERR36**: 1'b1 -> Force Assertion of Mission Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM2_CRC_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
3	0x0	**ERR35**: 1'b1 -> Force Assertion of Mission Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM1_CRC_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
2	0x0	**ERR34**: 1'b1 -> Force Assertion of Mission Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM0_CRC_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
1	0x0	**ERR33**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_PIXFMT_CTRL_ECC_DED 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR32**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_PIXFMT_CTRL_ECC_SEC 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

VI_EC_ERRSLICE1_MISSIONERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of MissionError_Enable register, to avoid silent dropping of errors.

Offset: 0xc1a
Byte Offset: 0x3068
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**ERR63**: 1'b1 -> Error_63_pulse[1:0] for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR4 was equal to 2'b10. 1'b0 -> Error_63_pulse[1:0] for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR4 was equal to 2'b01.
30	0x0	**ERR62**: 1'b1 -> Error_62_pulse[1:0] for Other Corrected Error from USER_DEFINED_CORRECTED_ERR3 was equal to 2'b10. 1'b0 -> Error_62_pulse[1:0] for Other Corrected Error from USER_DEFINED_CORRECTED_ERR3 was equal to 2'b01.
29	0x0	**ERR61**: 1'b1 -> Error_61_pulse[1:0] for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR3 was equal to 2'b10. 1'b0 -> Error_61_pulse[1:0] for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR3 was equal to 2'b01.

Bit	Reset	Description
28	0x0	**ERR60**: 1'b1 -> Error_60_pulse[1:0] for Other Corrected Error from USER_DEFINED_CORRECTED_ERR2 was equal to 2'b10. 1'b0 -> Error_60_pulse[1:0] for Other Corrected Error from USER_DEFINED_CORRECTED_ERR2 was equal to 2'b01.
27	0x0	**ERR59**: 1'b1 -> Error_59_pulse[1:0] for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR2 was equal to 2'b10. 1'b0 -> Error_59_pulse[1:0] for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR2 was equal to 2'b01.
26	0x0	**ERR58**: 1'b1 -> Error_58_pulse[1:0] for Other Corrected Error from USER_DEFINED_CORRECTED_ERR1 was equal to 2'b10. 1'b0 -> Error_58_pulse[1:0] for Other Corrected Error from USER_DEFINED_CORRECTED_ERR1 was equal to 2'b01.
25	0x0	**ERR57**: 1'b1 -> Error_57_pulse[1:0] for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR1 was equal to 2'b10. 1'b0 -> Error_57_pulse[1:0] for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR1 was equal to 2'b01.
24	0x0	**ERR56**: 1'b1 -> Error_56_pulse[1:0] for Other Corrected Error from USER_DEFINED_CORRECTED_ERR0 was equal to 2'b10. 1'b0 -> Error_56_pulse[1:0] for Other Corrected Error from USER_DEFINED_CORRECTED_ERR0 was equal to 2'b01.
23	0x0	**ERR55**: 1'b1 -> Error_55_pulse[1:0] for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR0 was equal to 2'b10. 1'b0 -> Error_55_pulse[1:0] for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR0 was equal to 2'b01.
22	0x0	**ERR54**: 1'b1 -> Error_54_pulse[1:0] for SW Generic Error from NV_vi.u_safety.u_NV_VI_err_collator.sw_err was equal to 2'b10. 1'b0 -> Error_54_pulse[1:0] for SW Generic Error from NV_vi.u_safety.u_NV_VI_err_collator.sw_err was equal to 2'b01.
21	0x0	**ERR53**: 1'b1 -> Error_53_pulse[1:0] for Register Parity Error from NV_vi.u_csbmaster.u_cfgreg was equal to 2'b10. 1'b0 -> Error_53_pulse[1:0] for Register Parity Error from NV_vi.u_csbmaster.u_cfgreg was equal to 2'b01.
20	0x0	**ERR52**: 1'b1 -> Error_52_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_CHANNEL_TRUNCATED_ERR was equal to 2'b10. 1'b0 -> Error_52_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_CHANNEL_TRUNCATED_ERR was equal to 2'b01.
19	0x0	**ERR51**: 1'b1 -> Error_51_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_CHANNEL_TOSSED_ERR was equal to 2'b10. 1'b0 -> Error_51_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_CHANNEL_TOSSED_ERR was equal to 2'b01.

Bit	Reset	Description
18	0x0	**ERR50**: 1'b1 -> Error_50_pulse[1:0] for Parity Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_DVFSFIFO_ADR_RAM_PARITY_ERR was equal to 2'b10. 1'b0 -> Error_50_pulse[1:0] for Parity Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_DVFSFIFO_ADR_RAM_PARITY_ERR was equal to 2'b01.
17	0x0	**ERR49**: 1'b1 -> Error_49_pulse[1:0] for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HRD_FIFO_ECC_DED was equal to 2'b10. 1'b0 -> Error_49_pulse[1:0] for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HRD_FIFO_ECC_DED was equal to 2'b01.
16	0x0	**ERR48**: 1'b1 -> Error_48_pulse[1:0] for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HRD_FIFO_ECC_SEC was equal to 2'b10. 1'b0 -> Error_48_pulse[1:0] for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HRD_FIFO_ECC_SEC was equal to 2'b01.
15	0x0	**ERR47**: 1'b1 -> Error_47_pulse[1:0] for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HWR_FIFO_ECC_DED was equal to 2'b10. 1'b0 -> Error_47_pulse[1:0] for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HWR_FIFO_ECC_DED was equal to 2'b01.
14	0x0	**ERR46**: 1'b1 -> Error_46_pulse[1:0] for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HWR_FIFO_ECC_SEC was equal to 2'b10. 1'b0 -> Error_46_pulse[1:0] for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HWR_FIFO_ECC_SEC was equal to 2'b01.
13	0x0	**ERR45**: 1'b1 -> Error_45_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_NOTIFY_FIFO_ECC_DED was equal to 2'b10. 1'b0 -> Error_45_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_NOTIFY_FIFO_ECC_DED was equal to 2'b01.
12	0x0	**ERR44**: 1'b1 -> Error_44_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_NOTIFY_FIFO_ECC_SEC was equal to 2'b10. 1'b0 -> Error_44_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_NOTIFY_FIFO_ECC_SEC was equal to 2'b01.
11	0x0	**ERR43**: 1'b1 -> Error_43_pulse[1:0] for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_WR_ACK_FIFO_ECC_DED was equal to 2'b10. 1'b0 -> Error_43_pulse[1:0] for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_WR_ACK_FIFO_ECC_DED was equal to 2'b01.
10	0x0	**ERR42**: 1'b1 -> Error_42_pulse[1:0] for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_WR_ACK_FIFO_ECC_SEC was equal to 2'b10. 1'b0 -> Error_42_pulse[1:0] for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_WR_ACK_FIFO_ECC_SEC was equal to 2'b01.

Bit	Reset	Description
9	0x0	**ERR41**: 1'b1 -> Error_41_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_OFIF_FIFO_CTRL_ECC_DED was equal to 2'b10. 1'b0 -> Error_41_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_OFIF_FIFO_CTRL_ECC_DED was equal to 2'b01.
8	0x0	**ERR40**: 1'b1 -> Error_40_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_OFIF_FIFO_CTRL_ECC_SEC was equal to 2'b10. 1'b0 -> Error_40_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_OFIF_FIFO_CTRL_ECC_SEC was equal to 2'b01.
7	0x0	**ERR39**: 1'b1 -> Error_39_pulse[1:0] for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM5_CRC_ERR was equal to 2'b10. 1'b0 -> Error_39_pulse[1:0] for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM5_CRC_ERR was equal to 2'b01.
6	0x0	**ERR38**: 1'b1 -> Error_38_pulse[1:0] for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM4_CRC_ERR was equal to 2'b10. 1'b0 -> Error_38_pulse[1:0] for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM4_CRC_ERR was equal to 2'b01.
5	0x0	**ERR37**: 1'b1 -> Error_37_pulse[1:0] for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM3_CRC_ERR was equal to 2'b10. 1'b0 -> Error_37_pulse[1:0] for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM3_CRC_ERR was equal to 2'b01.
4	0x0	**ERR36**: 1'b1 -> Error_36_pulse[1:0] for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM2_CRC_ERR was equal to 2'b10. 1'b0 -> Error_36_pulse[1:0] for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM2_CRC_ERR was equal to 2'b01.
3	0x0	**ERR35**: 1'b1 -> Error_35_pulse[1:0] for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM1_CRC_ERR was equal to 2'b10. 1'b0 -> Error_35_pulse[1:0] for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM1_CRC_ERR was equal to 2'b01.
2	0x0	**ERR34**: 1'b1 -> Error_34_pulse[1:0] for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM0_CRC_ERR was equal to 2'b10. 1'b0 -> Error_34_pulse[1:0] for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM0_CRC_ERR was equal to 2'b01.
1	0x0	**ERR33**: 1'b1 -> Error_33_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_PIXFMT_CTRL_ECC_DED was equal to 2'b10. 1'b0 -> Error_33_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_PIXFMT_CTRL_ECC_DED was equal to 2'b01.
0	0x0	**ERR32**: 1'b1 -> Error_32_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_PIXFMT_CTRL_ECC_SEC was equal to 2'b10. 1'b0 -> Error_32_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_PIXFMT_CTRL_ECC_SEC was equal to 2'b01.

VI_EC_ERRSLICE1_MISSIONERR_INJECT_0

Offset: 0xc1b
Byte Offset: 0x306c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0x00,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**ERR63**: 1'b1 -> Assert the inject_error_63 output for Other Uncorrected Error to USER_DEFINED_UNCORRECTED_ERR4 to allow for error injection. 1'b0 -> De-Asseert inject_error_63 output. **0** = DISABLE **1** = ENABLE
30	0x0	**ERR62**: 1'b1 -> Assert the inject_error_62 output for Other Corrected Error to USER_DEFINED_CORRECTED_ERR3 to allow for error injection. 1'b0 -> De-Asseert inject_error_62 output. **0** = DISABLE **1** = ENABLE
29	0x0	**ERR61**: 1'b1 -> Assert the inject_error_61 output for Other Uncorrected Error to USER_DEFINED_UNCORRECTED_ERR3 to allow for error injection. 1'b0 -> De-Asseert inject_error_61 output. **0** = DISABLE **1** = ENABLE
28	0x0	**ERR60**: 1'b1 -> Assert the inject_error_60 output for Other Corrected Error to USER_DEFINED_CORRECTED_ERR2 to allow for error injection. 1'b0 -> De-Asseert inject_error_60 output. **0** = DISABLE **1** = ENABLE
27	0x0	**ERR59**: 1'b1 -> Assert the inject_error_59 output for Other Uncorrected Error to USER_DEFINED_UNCORRECTED_ERR2 to allow for error injection. 1'b0 -> De-Asseert inject_error_59 output. **0** = DISABLE **1** = ENABLE
26	0x0	**ERR58**: 1'b1 -> Assert the inject_error_58 output for Other Corrected Error to USER_DEFINED_CORRECTED_ERR1 to allow for error injection. 1'b0 -> De-Asseert inject_error_58 output. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
25	0x0	**ERR57**: 1'b1 -> Assert the inject_error_57 output for Other Uncorrected Error to USER_DEFINED_UNCORRECTED_ERR1 to allow for error injection. 1'b0 -> De-Asseert inject_error_57 output. **0** = DISABLE **1** = ENABLE
24	0x0	**ERR56**: 1'b1 -> Assert the inject_error_56 output for Other Corrected Error to USER_DEFINED_CORRECTED_ERR0 to allow for error injection. 1'b0 -> De-Asseert inject_error_56 output. **0** = DISABLE **1** = ENABLE
23	0x0	**ERR55**: 1'b1 -> Assert the inject_error_55 output for Other Uncorrected Error to USER_DEFINED_UNCORRECTED_ERR0 to allow for error injection. 1'b0 -> De-Asseert inject_error_55 output. **0** = DISABLE **1** = ENABLE
21	0x0	**ERR53**: 1'b1 -> Assert the inject_error_53 output for Register Parity Error to NV_vi.u_csbmaster.u_cfgreg to allow for error injection. 1'b0 -> De-Asseert inject_error_53 output. **0** = DISABLE **1** = ENABLE
20	0x0	**ERR52**: 1'b1 -> Assert the inject_error_52 output for Other Uncorrected Error to FAULT_UDF_TAG_VI_FATAL_ATOMP_CHANNEL_TRUNCATED_ERR to allow for error injection. 1'b0 -> De-Asseert inject_error_52 output. **0** = DISABLE **1** = ENABLE
19	0x0	**ERR51**: 1'b1 -> Assert the inject_error_51 output for Other Uncorrected Error to FAULT_UDF_TAG_VI_FATAL_ATOMP_CHANNEL_TOSSED_ERR to allow for error injection. 1'b0 -> De-Asseert inject_error_51 output. **0** = DISABLE **1** = ENABLE
18	0x0	**ERR50**: 1'b1 -> Assert the inject_error_50 output for Parity Error from on-chip SRAM/FIFO to FAULT_UDF_TAG_VI_ATOMP_DVFSFIFO_ADR_RAM_PARITY_ERR to allow for error injection. 1'b0 -> De-Asseert inject_error_50 output. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
17	0x0	**ERR49**: 1'b1 -> Assert the inject_error_49 output for ECC DED Error from on-chip SRAM/FIFO to FAULT_UDF_TAG_VI_2HI_HRD_FIFO_ECC_DED to allow for error injection. 1'b0 -> De-Asseert inject_error_49 output. **0** = DISABLE **1** = ENABLE
16	0x0	**ERR48**: 1'b1 -> Assert the inject_error_48 output for ECC SEC Error from on-chip SRAM/FIFO to FAULT_UDF_TAG_VI_2HI_HRD_FIFO_ECC_SEC to allow for error injection. 1'b0 -> De-Asseert inject_error_48 output. **0** = DISABLE **1** = ENABLE
15	0x0	**ERR47**: 1'b1 -> Assert the inject_error_47 output for ECC DED Error from on-chip SRAM/FIFO to FAULT_UDF_TAG_VI_2HI_HWR_FIFO_ECC_DED to allow for error injection. 1'b0 -> De-Asseert inject_error_47 output. **0** = DISABLE **1** = ENABLE
14	0x0	**ERR46**: 1'b1 -> Assert the inject_error_46 output for ECC SEC Error from on-chip SRAM/FIFO to FAULT_UDF_TAG_VI_2HI_HWR_FIFO_ECC_SEC to allow for error injection. 1'b0 -> De-Asseert inject_error_46 output. **0** = DISABLE **1** = ENABLE
13	0x0	**ERR45**: 1'b1 -> Assert the inject_error_45 output for ECC DED Error on internal data path to FAULT_UDF_TAG_VI_NOTIFY_FIFO_ECC_DED to allow for error injection. 1'b0 -> De-Asseert inject_error_45 output. **0** = DISABLE **1** = ENABLE
12	0x0	**ERR44**: 1'b1 -> Assert the inject_error_44 output for ECC SEC Error on internal data path to FAULT_UDF_TAG_VI_NOTIFY_FIFO_ECC_SEC to allow for error injection. 1'b0 -> De-Asseert inject_error_44 output. **0** = DISABLE **1** = ENABLE
11	0x0	**ERR43**: 1'b1 -> Assert the inject_error_43 output for ECC DED Error from on-chip SRAM/FIFO to FAULT_UDF_TAG_VI_ATOMP_WR_ACK_FIFO_ECC_DED to allow for error injection. 1'b0 -> De-Asseert inject_error_43 output. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
10	0x0	**ERR42**: 1'b1 -> Assert the inject_error_42 output for ECC SEC Error from on-chip SRAM/FIFO to FAULT_UDF_TAG_VI_ATOMP_WR_ACK_FIFO_ECC_SEC to allow for error injection. 1'b0 -> De-Asseert inject_error_42 output. **0** = DISABLE **1** = ENABLE
9	0x0	**ERR41**: 1'b1 -> Assert the inject_error_41 output for ECC DED Error on internal data path to FAULT_UDF_TAG_VI_OFIF_FIFO_CTRL_ECC_DED to allow for error injection. 1'b0 -> De-Asseert inject_error_41 output. **0** = DISABLE **1** = ENABLE
8	0x0	**ERR40**: 1'b1 -> Assert the inject_error_40 output for ECC SEC Error on internal data path to FAULT_UDF_TAG_VI_OFIF_FIFO_CTRL_ECC_SEC to allow for error injection. 1'b0 -> De-Asseert inject_error_40 output. **0** = DISABLE **1** = ENABLE
7	0x0	**ERR39**: 1'b1 -> Assert the inject_error_39 output for CRC Error on internal data path to FAULT_UDF_TAG_VI_ATOMP_STREAM5_CRC_ERR to allow for error injection. 1'b0 -> De-Asseert inject_error_39 output. **0** = DISABLE **1** = ENABLE
6	0x0	**ERR38**: 1'b1 -> Assert the inject_error_38 output for CRC Error on internal data path to FAULT_UDF_TAG_VI_ATOMP_STREAM4_CRC_ERR to allow for error injection. 1'b0 -> De-Asseert inject_error_38 output. **0** = DISABLE **1** = ENABLE
5	0x0	**ERR37**: 1'b1 -> Assert the inject_error_37 output for CRC Error on internal data path to FAULT_UDF_TAG_VI_ATOMP_STREAM3_CRC_ERR to allow for error injection. 1'b0 -> De-Asseert inject_error_37 output. **0** = DISABLE **1** = ENABLE
4	0x0	**ERR36**: 1'b1 -> Assert the inject_error_36 output for CRC Error on internal data path to FAULT_UDF_TAG_VI_ATOMP_STREAM2_CRC_ERR to allow for error injection. 1'b0 -> De-Asseert inject_error_36 output. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
3	0x0	**ERR35**: 1'b1 -> Assert the inject_error_35 output for CRC Error on internal data path to FAULT_UDF_TAG_VI_ATOMP_STREAM1_CRC_ERR to allow for error injection. 1'b0 -> De-Asseert inject_error_35 output. **0** = DISABLE **1** = ENABLE
2	0x0	**ERR34**: 1'b1 -> Assert the inject_error_34 output for CRC Error on internal data path to FAULT_UDF_TAG_VI_ATOMP_STREAM0_CRC_ERR to allow for error injection. 1'b0 -> De-Asseert inject_error_34 output. **0** = DISABLE **1** = ENABLE
1	0x0	**ERR33**: 1'b1 -> Assert the inject_error_33 output for ECC DED Error on internal data path to FAULT_UDF_TAG_VI_PIXFMT_CTRL_ECC_DED to allow for error injection. 1'b0 -> De-Asseert inject_error_33 output. **0** = DISABLE **1** = ENABLE
0	0x0	**ERR32**: 1'b1 -> Assert the inject_error_32 output for ECC SEC Error on internal data path to FAULT_UDF_TAG_VI_PIXFMT_CTRL_ECC_SEC to allow for error injection. 1'b0 -> De-Asseert inject_error_32 output. **0** = DISABLE **1** = ENABLE

VI_EC_ERRSLICE1_LATENTERR_ENABLE_0

Offset: 0xc1c
Byte Offset: 0x3070
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31	0x1	**ERR63**: 1'b1 -> Enable Latent Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR4 1'b0 -> Disable Latent Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR4 **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
30	0x1	**ERR62**: 1'b1 -> Enable Latent Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR3 1'b0 -> Disable Latent Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR3 **0** = DISABLE **1** = ENABLE
29	0x1	**ERR61**: 1'b1 -> Enable Latent Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR3 1'b0 -> Disable Latent Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR3 **0** = DISABLE **1** = ENABLE
28	0x1	**ERR60**: 1'b1 -> Enable Latent Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR2 1'b0 -> Disable Latent Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR2 **0** = DISABLE **1** = ENABLE
27	0x1	**ERR59**: 1'b1 -> Enable Latent Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR2 1'b0 -> Disable Latent Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR2 **0** = DISABLE **1** = ENABLE
26	0x1	**ERR58**: 1'b1 -> Enable Latent Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR1 1'b0 -> Disable Latent Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR1 **0** = DISABLE **1** = ENABLE
25	0x1	**ERR57**: 1'b1 -> Enable Latent Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR1 1'b0 -> Disable Latent Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR1 **0** = DISABLE **1** = ENABLE
24	0x1	**ERR56**: 1'b1 -> Enable Latent Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR0 1'b0 -> Disable Latent Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR0 **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
23	0x1	**ERR55**: 1'b1 -> Enable Latent Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR0 1'b0 -> Disable Latent Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR0 **0** = DISABLE **1** = ENABLE
22	0x1	**ERR54**: 1'b1 -> Enable Latent Error Reporting for SW Generic Error from NV_vi.u_safety.u_NV_VI_err_collator.sw_err 1'b0 -> Disable Latent Error Reporting for SW Generic Error from NV_vi.u_safety.u_NV_VI_err_collator.sw_err **0** = DISABLE **1** = ENABLE
21	0x1	**ERR53**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from NV_vi.u_csbmaster.u_cfgreg 1'b0 -> Disable Latent Error Reporting for Register Parity Error from NV_vi.u_csbmaster.u_cfgreg **0** = DISABLE **1** = ENABLE
20	0x1	**ERR52**: 1'b1 -> Enable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_CHANNEL_TRUNCATED_ERR 1'b0 -> Disable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_CHANNEL_TRUNCATED_ERR **0** = DISABLE **1** = ENABLE
19	0x1	**ERR51**: 1'b1 -> Enable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_CHANNEL_TOSSED_ERR 1'b0 -> Disable Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_CHANNEL_TOSSED_ERR **0** = DISABLE **1** = ENABLE
18	0x1	**ERR50**: 1'b1 -> Enable Latent Error Reporting for Parity Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_DVFSFIFO_ADR_RAM_PARITY_ERR 1'b0 -> Disable Latent Error Reporting for Parity Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_DVFSFIFO_ADR_RAM_PARITY_ERR **0** = DISABLE **1** = ENABLE
17	0x1	**ERR49**: 1'b1 -> Enable Latent Error Reporting for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HRD_FIFO_ECC_DED 1'b0 -> Disable Latent Error Reporting for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HRD_FIFO_ECC_DED **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
16	0x1	**ERR48**: 1'b1 -> Enable Latent Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HRD_FIFO_ECC_SEC 1'b0 -> Disable Latent Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HRD_FIFO_ECC_SEC **0** = DISABLE **1** = ENABLE
15	0x1	**ERR47**: 1'b1 -> Enable Latent Error Reporting for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HWR_FIFO_ECC_DED 1'b0 -> Disable Latent Error Reporting for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HWR_FIFO_ECC_DED **0** = DISABLE **1** = ENABLE
14	0x1	**ERR46**: 1'b1 -> Enable Latent Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HWR_FIFO_ECC_SEC 1'b0 -> Disable Latent Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HWR_FIFO_ECC_SEC **0** = DISABLE **1** = ENABLE
13	0x1	**ERR45**: 1'b1 -> Enable Latent Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_NOTIFY_FIFO_ECC_DED 1'b0 -> Disable Latent Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_NOTIFY_FIFO_ECC_DED **0** = DISABLE **1** = ENABLE
12	0x1	**ERR44**: 1'b1 -> Enable Latent Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_NOTIFY_FIFO_ECC_SEC 1'b0 -> Disable Latent Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_NOTIFY_FIFO_ECC_SEC **0** = DISABLE **1** = ENABLE
11	0x1	**ERR43**: 1'b1 -> Enable Latent Error Reporting for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_WR_ACK_FIFO_ECC_DED 1'b0 -> Disable Latent Error Reporting for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_WR_ACK_FIFO_ECC_DED **0** = DISABLE **1** = ENABLE
10	0x1	**ERR42**: 1'b1 -> Enable Latent Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_WR_ACK_FIFO_ECC_SEC 1'b0 -> Disable Latent Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_WR_ACK_FIFO_ECC_SEC **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
9	0x1	**ERR41**: 1'b1 -> Enable Latent Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_OFIF_FIFO_CTRL_ECC_DED 1'b0 -> Disable Latent Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_OFIF_FIFO_CTRL_ECC_DED **0** = DISABLE **1** = ENABLE
8	0x1	**ERR40**: 1'b1 -> Enable Latent Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_OFIF_FIFO_CTRL_ECC_SEC 1'b0 -> Disable Latent Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_OFIF_FIFO_CTRL_ECC_SEC **0** = DISABLE **1** = ENABLE
7	0x1	**ERR39**: 1'b1 -> Enable Latent Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM5_CRC_ERR 1'b0 -> Disable Latent Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM5_CRC_ERR **0** = DISABLE **1** = ENABLE
6	0x1	**ERR38**: 1'b1 -> Enable Latent Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM4_CRC_ERR 1'b0 -> Disable Latent Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM4_CRC_ERR **0** = DISABLE **1** = ENABLE
5	0x1	**ERR37**: 1'b1 -> Enable Latent Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM3_CRC_ERR 1'b0 -> Disable Latent Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM3_CRC_ERR **0** = DISABLE **1** = ENABLE
4	0x1	**ERR36**: 1'b1 -> Enable Latent Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM2_CRC_ERR 1'b0 -> Disable Latent Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM2_CRC_ERR **0** = DISABLE **1** = ENABLE
3	0x1	**ERR35**: 1'b1 -> Enable Latent Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM1_CRC_ERR 1'b0 -> Disable Latent Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM1_CRC_ERR **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
2	0x1	**ERR34**: 1'b1 -> Enable Latent Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM0_CRC_ERR 1'b0 -> Disable Latent Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM0_CRC_ERR **0** = DISABLE **1** = ENABLE
1	0x1	**ERR33**: 1'b1 -> Enable Latent Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_PIXFMT_CTRL_ECC_DED 1'b0 -> Disable Latent Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_PIXFMT_CTRL_ECC_DED **0** = DISABLE **1** = ENABLE
0	0x1	**ERR32**: 1'b1 -> Enable Latent Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_PIXFMT_CTRL_ECC_SEC 1'b0 -> Disable Latent Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_PIXFMT_CTRL_ECC_SEC **0** = DISABLE **1** = ENABLE

VI_EC_ERRSLICE1_LATENTERR_FORCE_0

Offset: 0xc1d
Byte Offset: 0x3074
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**ERR63**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR4 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
30	0x0	**ERR62**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR3 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
29	0x0	**ERR61**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR3 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
28	0x0	**ERR60**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR2 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
27	0x0	**ERR59**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR2 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
26	0x0	**ERR58**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR1 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
25	0x0	**ERR57**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR1 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
24	0x0	**ERR56**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR0 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
23	0x0	**ERR55**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR0 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
22	0x0	**ERR54**: 1'b1 -> Force Assertion of Latent Error Reporting for SW Generic Error from NV_vi.u_safety.u_NV_VI_err_collator.sw_err 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
21	0x0	**ERR53**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from NV_vi.u_csbmaster.u_cfgreg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
20	0x0	**ERR52**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_CHANNEL_TRUNCATED_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
19	0x0	**ERR51**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_CHANNEL_TOSSED_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
18	0x0	**ERR50**: 1'b1 -> Force Assertion of Latent Error Reporting for Parity Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_DVFSFIFO_ADR_RAM_PARITY_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
17	0x0	**ERR49**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HRD_FIFO_ECC_DED 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
16	0x0	**ERR48**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HRD_FIFO_ECC_SEC 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
15	0x0	**ERR47**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HWR_FIFO_ECC_DED 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
14	0x0	**ERR46**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HWR_FIFO_ECC_SEC 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
13	0x0	**ERR45**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_NOTIFY_FIFO_ECC_DED 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
12	0x0	**ERR44**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_NOTIFY_FIFO_ECC_SEC 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
11	0x0	**ERR43**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_WR_ACK_FIFO_ECC_DED 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
10	0x0	**ERR42**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_WR_ACK_FIFO_ECC_SEC 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
9	0x0	**ERR41**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_OFIF_FIFO_CTRL_ECC_DED 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
8	0x0	**ERR40**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_OFIF_FIFO_CTRL_ECC_SEC 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
7	0x0	**ERR39**: 1'b1 -> Force Assertion of Latent Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM5_CRC_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
6	0x0	**ERR38**: 1'b1 -> Force Assertion of Latent Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM4_CRC_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
5	0x0	**ERR37**: 1'b1 -> Force Assertion of Latent Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM3_CRC_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
4	0x0	**ERR36**: 1'b1 -> Force Assertion of Latent Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM2_CRC_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
3	0x0	**ERR35**: 1'b1 -> Force Assertion of Latent Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM1_CRC_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
2	0x0	**ERR34**: 1'b1 -> Force Assertion of Latent Error Reporting for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM0_CRC_ERR 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR33**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_PIXFMT_CTRL_ECC_DED 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR32**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_PIXFMT_CTRL_ECC_SEC 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

VI_EC_ERRSLICE1_LATENTERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of LatentError_Enable register, to avoid silent dropping of errors.

Offset: 0xc1e
Byte Offset: 0x3078
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**ERR63**: 1'b1 -> Error_63_pulse[1:0] for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR4 was equal to 2'b00 or 2'b11. 1'b0 -> Error_63_pulse[1:0] for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR4 was equal to 2'b01 or 2'b10, but no latent error.
30	0x0	**ERR62**: 1'b1 -> Error_62_pulse[1:0] for Other Corrected Error from USER_DEFINED_CORRECTED_ERR3 was equal to 2'b00 or 2'b11. 1'b0 -> Error_62_pulse[1:0] for Other Corrected Error from USER_DEFINED_CORRECTED_ERR3 was equal to 2'b01 or 2'b10, but no latent error.
29	0x0	**ERR61**: 1'b1 -> Error_61_pulse[1:0] for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR3 was equal to 2'b00 or 2'b11. 1'b0 -> Error_61_pulse[1:0] for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR3 was equal to 2'b01 or 2'b10, but no latent error.
28	0x0	**ERR60**: 1'b1 -> Error_60_pulse[1:0] for Other Corrected Error from USER_DEFINED_CORRECTED_ERR2 was equal to 2'b00 or 2'b11. 1'b0 -> Error_60_pulse[1:0] for Other Corrected Error from USER_DEFINED_CORRECTED_ERR2 was equal to 2'b01 or 2'b10, but no latent error.
27	0x0	**ERR59**: 1'b1 -> Error_59_pulse[1:0] for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR2 was equal to 2'b00 or 2'b11. 1'b0 -> Error_59_pulse[1:0] for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR2 was equal to 2'b01 or 2'b10, but no latent error.
26	0x0	**ERR58**: 1'b1 -> Error_58_pulse[1:0] for Other Corrected Error from USER_DEFINED_CORRECTED_ERR1 was equal to 2'b00 or 2'b11. 1'b0 -> Error_58_pulse[1:0] for Other Corrected Error from USER_DEFINED_CORRECTED_ERR1 was equal to 2'b01 or 2'b10, but no latent error.

Bit	Reset	Description
25	0x0	**ERR57**: 1'b1 -> Error_57_pulse[1:0] for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR1 was equal to 2'b00 or 2'b11. 1'b0 -> Error_57_pulse[1:0] for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR1 was equal to 2'b01 or 2'b10, but no latent error.
24	0x0	**ERR56**: 1'b1 -> Error_56_pulse[1:0] for Other Corrected Error from USER_DEFINED_CORRECTED_ERR0 was equal to 2'b00 or 2'b11. 1'b0 -> Error_56_pulse[1:0] for Other Corrected Error from USER_DEFINED_CORRECTED_ERR0 was equal to 2'b01 or 2'b10, but no latent error.
23	0x0	**ERR55**: 1'b1 -> Error_55_pulse[1:0] for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR0 was equal to 2'b00 or 2'b11. 1'b0 -> Error_55_pulse[1:0] for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR0 was equal to 2'b01 or 2'b10, but no latent error.
22	0x0	**ERR54**: 1'b1 -> Error_54_pulse[1:0] for SW Generic Error from NV_vi.u_safety.u_NV_VI_err_collator.sw_err was equal to 2'b00 or 2'b11. 1'b0 -> Error_54_pulse[1:0] for SW Generic Error from NV_vi.u_safety.u_NV_VI_err_collator.sw_err was equal to 2'b01 or 2'b10, but no latent error.
21	0x0	**ERR53**: 1'b1 -> Error_53_pulse[1:0] for Register Parity Error from NV_vi.u_csbmaster.u_cfgreg was equal to 2'b00 or 2'b11. 1'b0 -> Error_53_pulse[1:0] for Register Parity Error from NV_vi.u_csbmaster.u_cfgreg was equal to 2'b01 or 2'b10, but no latent error.
20	0x0	**ERR52**: 1'b1 -> Error_52_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_CHANNEL_TRUNCATED_ERR was equal to 2'b00 or 2'b11. 1'b0 -> Error_52_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_CHANNEL_TRUNCATED_ERR was equal to 2'b01 or 2'b10, but no latent error.
19	0x0	**ERR51**: 1'b1 -> Error_51_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_CHANNEL_TOSSED_ERR was equal to 2'b00 or 2'b11. 1'b0 -> Error_51_pulse[1:0] for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_CHANNEL_TOSSED_ERR was equal to 2'b01 or 2'b10, but no latent error.
18	0x0	**ERR50**: 1'b1 -> Error_50_pulse[1:0] for Parity Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_DVFSFIFO_ADR_RAM_PARITY_ERR was equal to 2'b00 or 2'b11. 1'b0 -> Error_50_pulse[1:0] for Parity Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_DVFSFIFO_ADR_RAM_PARITY_ERR was equal to 2'b01 or 2'b10, but no latent error.

Bit	Reset	Description
17	0x0	**ERR49**: 1'b1 -> Error_49_pulse[1:0] for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HRD_FIFO_ECC_DED was equal to 2'b00 or 2'b11. 1'b0 -> Error_49_pulse[1:0] for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HRD_FIFO_ECC_DED was equal to 2'b01 or 2'b10, but no latent error.
16	0x0	**ERR48**: 1'b1 -> Error_48_pulse[1:0] for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HRD_FIFO_ECC_SEC was equal to 2'b00 or 2'b11. 1'b0 -> Error_48_pulse[1:0] for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HRD_FIFO_ECC_SEC was equal to 2'b01 or 2'b10, but no latent error.
15	0x0	**ERR47**: 1'b1 -> Error_47_pulse[1:0] for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HWR_FIFO_ECC_DED was equal to 2'b00 or 2'b11. 1'b0 -> Error_47_pulse[1:0] for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HWR_FIFO_ECC_DED was equal to 2'b01 or 2'b10, but no latent error.
14	0x0	**ERR46**: 1'b1 -> Error_46_pulse[1:0] for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HWR_FIFO_ECC_SEC was equal to 2'b00 or 2'b11. 1'b0 -> Error_46_pulse[1:0] for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HWR_FIFO_ECC_SEC was equal to 2'b01 or 2'b10, but no latent error.
13	0x0	**ERR45**: 1'b1 -> Error_45_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_NOTIFY_FIFO_ECC_DED was equal to 2'b00 or 2'b11. 1'b0 -> Error_45_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_NOTIFY_FIFO_ECC_DED was equal to 2'b01 or 2'b10, but no latent error.
12	0x0	**ERR44**: 1'b1 -> Error_44_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_NOTIFY_FIFO_ECC_SEC was equal to 2'b00 or 2'b11. 1'b0 -> Error_44_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_NOTIFY_FIFO_ECC_SEC was equal to 2'b01 or 2'b10, but no latent error.
11	0x0	**ERR43**: 1'b1 -> Error_43_pulse[1:0] for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_WR_ACK_FIFO_ECC_DED was equal to 2'b00 or 2'b11. 1'b0 -> Error_43_pulse[1:0] for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_WR_ACK_FIFO_ECC_DED was equal to 2'b01 or 2'b10, but no latent error.
10	0x0	**ERR42**: 1'b1 -> Error_42_pulse[1:0] for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_WR_ACK_FIFO_ECC_SEC was equal to 2'b00 or 2'b11. 1'b0 -> Error_42_pulse[1:0] for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_WR_ACK_FIFO_ECC_SEC was equal to 2'b01 or 2'b10, but no latent error.

Bit	Reset	Description
9	0x0	**ERR41**: 1'b1 -> Error_41_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_OFIF_FIFO_CTRL_ECC_DED was equal to 2'b00 or 2'b11. 1'b0 -> Error_41_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_OFIF_FIFO_CTRL_ECC_DED was equal to 2'b01 or 2'b10, but no latent error.
8	0x0	**ERR40**: 1'b1 -> Error_40_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_OFIF_FIFO_CTRL_ECC_SEC was equal to 2'b00 or 2'b11. 1'b0 -> Error_40_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_OFIF_FIFO_CTRL_ECC_SEC was equal to 2'b01 or 2'b10, but no latent error.
7	0x0	**ERR39**: 1'b1 -> Error_39_pulse[1:0] for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM5_CRC_ERR was equal to 2'b00 or 2'b11. 1'b0 -> Error_39_pulse[1:0] for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM5_CRC_ERR was equal to 2'b01 or 2'b10, but no latent error.
6	0x0	**ERR38**: 1'b1 -> Error_38_pulse[1:0] for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM4_CRC_ERR was equal to 2'b00 or 2'b11. 1'b0 -> Error_38_pulse[1:0] for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM4_CRC_ERR was equal to 2'b01 or 2'b10, but no latent error.
5	0x0	**ERR37**: 1'b1 -> Error_37_pulse[1:0] for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM3_CRC_ERR was equal to 2'b00 or 2'b11. 1'b0 -> Error_37_pulse[1:0] for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM3_CRC_ERR was equal to 2'b01 or 2'b10, but no latent error.
4	0x0	**ERR36**: 1'b1 -> Error_36_pulse[1:0] for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM2_CRC_ERR was equal to 2'b00 or 2'b11. 1'b0 -> Error_36_pulse[1:0] for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM2_CRC_ERR was equal to 2'b01 or 2'b10, but no latent error.
3	0x0	**ERR35**: 1'b1 -> Error_35_pulse[1:0] for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM1_CRC_ERR was equal to 2'b00 or 2'b11. 1'b0 -> Error_35_pulse[1:0] for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM1_CRC_ERR was equal to 2'b01 or 2'b10, but no latent error.
2	0x0	**ERR34**: 1'b1 -> Error_34_pulse[1:0] for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM0_CRC_ERR was equal to 2'b00 or 2'b11. 1'b0 -> Error_34_pulse[1:0] for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM0_CRC_ERR was equal to 2'b01 or 2'b10, but no latent error.
1	0x0	**ERR33**: 1'b1 -> Error_33_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_PIXFMT_CTRL_ECC_DED was equal to 2'b00 or 2'b11. 1'b0 -> Error_33_pulse[1:0] for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_PIXFMT_CTRL_ECC_DED was equal to 2'b01 or 2'b10, but no latent error.

Bit	Reset	Description
0	0x0	**ERR32**: 1'b1 -> Error_32_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_PIXFMT_CTRL_ECC_SEC was equal to 2'b00 or 2'b11. 1'b0 -> Error_32_pulse[1:0] for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_PIXFMT_CTRL_ECC_SEC was equal to 2'b01 or 2'b10, but no latent error.

VI_EC_ERRSLICE1_COUNTER_RELOAD_0

Offset: 0xc20
Byte Offset: 0x3080
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**ERR63**: 1'b1 -> Reload Error Counter for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR4 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
30	0x0	**ERR62**: 1'b1 -> Reload Error Counter for Other Corrected Error from USER_DEFINED_CORRECTED_ERR3 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
29	0x0	**ERR61**: 1'b1 -> Reload Error Counter for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR3 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
28	0x0	**ERR60**: 1'b1 -> Reload Error Counter for Other Corrected Error from USER_DEFINED_CORRECTED_ERR2 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
27	0x0	**ERR59**: 1'b1 -> Reload Error Counter for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR2 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

26	0x0	**ERR58**: 1'b1 -> Reload Error Counter for Other Corrected Error from USER_DEFINED_CORRECTED_ERR1 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
25	0x0	**ERR57**: 1'b1 -> Reload Error Counter for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR1 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
24	0x0	**ERR56**: 1'b1 -> Reload Error Counter for Other Corrected Error from USER_DEFINED_CORRECTED_ERR0 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
23	0x0	**ERR55**: 1'b1 -> Reload Error Counter for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR0 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
22	0x0	**ERR54**: 1'b1 -> Reload Error Counter for SW Generic Error from NV_vi.u_safety.u_NV_VI_err_collator.sw_err 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
21	0x0	**ERR53**: 1'b1 -> Reload Error Counter for Register Parity Error from NV_vi.u_csbmaster.u_cfgreg 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
20	0x0	**ERR52**: 1'b1 -> Reload Error Counter for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_CHANNEL_TRUNCATED_ERR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
19	0x0	**ERR51**: 1'b1 -> Reload Error Counter for Other Uncorrected Error from FAULT_UDF_TAG_VI_FATAL_ATOMP_CHANNEL_TOSSED_ERR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

18	0x0	**ERR50**: 1'b1 -> Reload Error Counter for Parity Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_DVFSFIFO_ADR_RAM_PARITY_ERR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
17	0x0	**ERR49**: 1'b1 -> Reload Error Counter for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HRD_FIFO_ECC_DED 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
16	0x0	**ERR48**: 1'b1 -> Reload Error Counter for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HRD_FIFO_ECC_SEC 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
15	0x0	**ERR47**: 1'b1 -> Reload Error Counter for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HWR_FIFO_ECC_DED 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
14	0x0	**ERR46**: 1'b1 -> Reload Error Counter for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_2HI_HWR_FIFO_ECC_SEC 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
13	0x0	**ERR45**: 1'b1 -> Reload Error Counter for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_NOTIFY_FIFO_ECC_DED 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
12	0x0	**ERR44**: 1'b1 -> Reload Error Counter for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_NOTIFY_FIFO_ECC_SEC 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
11	0x0	**ERR43**: 1'b1 -> Reload Error Counter for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_WR_ACK_FIFO_ECC_DED 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

10	0x0	**ERR42**: 1'b1 -> Reload Error Counter for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_ATOMP_WR_ACK_FIFO_ECC_SEC 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
9	0x0	**ERR41**: 1'b1 -> Reload Error Counter for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_OFIF_FIFO_CTRL_ECC_DED 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
8	0x0	**ERR40**: 1'b1 -> Reload Error Counter for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_OFIF_FIFO_CTRL_ECC_SEC 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
7	0x0	**ERR39**: 1'b1 -> Reload Error Counter for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM5_CRC_ERR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
6	0x0	**ERR38**: 1'b1 -> Reload Error Counter for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM4_CRC_ERR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
5	0x0	**ERR37**: 1'b1 -> Reload Error Counter for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM3_CRC_ERR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
4	0x0	**ERR36**: 1'b1 -> Reload Error Counter for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM2_CRC_ERR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
3	0x0	**ERR35**: 1'b1 -> Reload Error Counter for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM1_CRC_ERR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

2	0x0	**ERR34**: 1'b1 -> Reload Error Counter for CRC Error on internal data path from FAULT_UDF_TAG_VI_ATOMP_STREAM0_CRC_ERR 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
1	0x0	**ERR33**: 1'b1 -> Reload Error Counter for ECC DED Error on internal data path from FAULT_UDF_TAG_VI_PIXFMT_CTRL_ECC_DED 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
0	0x0	**ERR32**: 1'b1 -> Reload Error Counter for ECC SEC Error on internal data path from FAULT_UDF_TAG_VI_PIXFMT_CTRL_ECC_SEC 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

VI_EC_ERRSLICE1_MISSIONERR_ECC_CORRECTION_DIS_0

Offset: 0xc21
Byte Offset: 0x3084
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,x0x0,x0x0,xxxx,xxx0)

Bit	Reset	Description
16	0x0	**ERR48**: 1'b1 -> Disables ECC correction in ECC Terminate plugins for FAULT_UDF_TAG_VI_2HI_HRD_FIFO_ECC_SEC and shall report single bit errors over the ded_err_pulse to only allow detection and no correction. 1'b0 -> ECC correction is enabled in ECC terminate plugins for FAULT_UDF_TAG_VI_2HI_HRD_FIFO_ECC_SEC and shall correct single bit errors and report them over sec_err_pulse **0** = DISABLE **1** = ENABLE
14	0x0	**ERR46**: 1'b1 -> Disables ECC correction in ECC Terminate plugins for FAULT_UDF_TAG_VI_2HI_HWR_FIFO_ECC_SEC and shall report single bit errors over the ded_err_pulse to only allow detection and no correction. 1'b0 -> ECC correction is enabled in ECC terminate plugins for FAULT_UDF_TAG_VI_2HI_HWR_FIFO_ECC_SEC and shall correct single bit errors and report them over sec_err_pulse **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
12	0x0	**ERR44**: 1'b1 -> Disables ECC correction in ECC Terminate plugins for FAULT_UDF_TAG_VI_NOTIFY_FIFO_ECC_SEC and shall report single bit errors over the ded_err_pulse to only allow detection and no correction. 1'b0 -> ECC correction is enabled in ECC terminate plugins for FAULT_UDF_TAG_VI_NOTIFY_FIFO_ECC_SEC and shall correct single bit errors and report them over sec_err_pulse **0** = DISABLE **1** = ENABLE
10	0x0	**ERR42**: 1'b1 -> Disables ECC correction in ECC Terminate plugins for FAULT_UDF_TAG_VI_ATOMP_WR_ACK_FIFO_ECC_SEC and shall report single bit errors over the ded_err_pulse to only allow detection and no correction. 1'b0 -> ECC correction is enabled in ECC terminate plugins for FAULT_UDF_TAG_VI_ATOMP_WR_ACK_FIFO_ECC_SEC and shall correct single bit errors and report them over sec_err_pulse **0** = DISABLE **1** = ENABLE
8	0x0	**ERR40**: 1'b1 -> Disables ECC correction in ECC Terminate plugins for FAULT_UDF_TAG_VI_OFIF_FIFO_CTRL_ECC_SEC and shall report single bit errors over the ded_err_pulse to only allow detection and no correction. 1'b0 -> ECC correction is enabled in ECC terminate plugins for FAULT_UDF_TAG_VI_OFIF_FIFO_CTRL_ECC_SEC and shall correct single bit errors and report them over sec_err_pulse **0** = DISABLE **1** = ENABLE
0	0x0	**ERR32**: 1'b1 -> Disables ECC correction in ECC Terminate plugins for FAULT_UDF_TAG_VI_PIXFMT_CTRL_ECC_SEC and shall report single bit errors over the ded_err_pulse to only allow detection and no correction. 1'b0 -> ECC correction is enabled in ECC terminate plugins for FAULT_UDF_TAG_VI_PIXFMT_CTRL_ECC_SEC and shall correct single bit errors and report them over sec_err_pulse **0** = DISABLE **1** = ENABLE

VI_EC_ERRSLICE2_MISSIONERR_ENABLE_0

Offset: 0xc24
Byte Offset: 0x3090
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x0000001f (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1,1111)

Bit	Reset	Description
4	0x1	**ERR68**: 1'b1 -> Enable Mission Error Reporting for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_NOTIFY_TSC_FIFO_ECC_DED 1'b0 -> Disable Mission Error Reporting for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_NOTIFY_TSC_FIFO_ECC_DED **0** = DISABLE **1** = ENABLE
3	0x1	**ERR67**: 1'b1 -> Enable Mission Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_NOTIFY_TSC_FIFO_ECC_SEC 1'b0 -> Disable Mission Error Reporting for ECC SEC Error from on-chip SRAM/FIFOfrom FAULT_UDF_TAG_VI_NOTIFY_TSC_FIFO_ECC_SEC **0** = DISABLE **1** = ENABLE
2	0x1	**ERR66**: 1'b1 -> Enable Mission Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR5 1'b0 -> Disable Mission Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR5 **0** = DISABLE **1** = ENABLE
1	0x1	**ERR65**: 1'b1 -> Enable Mission Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR5 1'b0 -> Disable Mission Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR5 **0** = DISABLE **1** = ENABLE
0	0x1	**ERR64**: 1'b1 -> Enable Mission Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR4 1'b0 -> Disable Mission Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR4 **0** = DISABLE **1** = ENABLE

VI_EC_ERRSLICE2_MISSIONERR_FORCE_0

Offset: 0xc25
Byte Offset: 0x3094
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**ERR68**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_NOTIFY_TSC_FIFO_ECC_DED 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
3	0x0	**ERR67**: 1'b1 -> Force Assertion of Mission Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_NOTIFY_TSC_FIFO_ECC_SEC 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
2	0x0	**ERR66**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR5 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR65**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR5 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR64**: 1'b1 -> Force Assertion of Mission Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR4 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

VI_EC_ERRSLICE2_MISSIONERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of MissionError_Enable register, to avoid silent dropping of errors.

Offset: 0xc26
Byte Offset: 0x3098
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**ERR68**: 1'b1 -> Error_68_pulse[1:0] for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_NOTIFY_TSC_FIFO_ECC_DED was equal to 2'b10. 1'b0 -> Error_68_pulse[1:0] for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_NOTIFY_TSC_FIFO_ECC_DED was equal to 2'b01.
3	0x0	**ERR67**: 1'b1 -> Error_67_pulse[1:0] for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_NOTIFY_TSC_FIFO_ECC_SEC was equal to 2'b10. 1'b0 -> Error_67_pulse[1:0] for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_NOTIFY_TSC_FIFO_ECC_SEC was equal to 2'b01.
2	0x0	**ERR66**: 1'b1 -> Error_66_pulse[1:0] for Other Corrected Error from USER_DEFINED_CORRECTED_ERR5 was equal to 2'b10. 1'b0 -> Error_66_pulse[1:0] for Other Corrected Error from USER_DEFINED_CORRECTED_ERR5 was equal to 2'b01.
1	0x0	**ERR65**: 1'b1 -> Error_65_pulse[1:0] for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR5 was equal to 2'b10. 1'b0 -> Error_65_pulse[1:0] for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR5 was equal to 2'b01.
0	0x0	**ERR64**: 1'b1 -> Error_64_pulse[1:0] for Other Corrected Error from USER_DEFINED_CORRECTED_ERR4 was equal to 2'b10. 1'b0 -> Error_64_pulse[1:0] for Other Corrected Error from USER_DEFINED_CORRECTED_ERR4 was equal to 2'b01.

VI_EC_ERRSLICE2_MISSIONERR_INJECT_0

Offset: 0xc27
Byte Offset: 0x309c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000

Bit	Reset	Description
4	0x0	**ERR68**: 1'b1 -> Assert the inject_error_68 output for ECC DED Error from on-chip SRAM/FIFO to FAULT_UDF_TAG_VI_NOTIFY_TSC_FIFO_ECC_DED to allow for error injection. 1'b0 -> De-Asseert inject_error_68 output. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
3	0x0	**ERR67**: 1'b1 -> Assert the inject_error_67 output for ECC SEC Error from on-chip SRAM/FIFO to FAULT_UDF_TAG_VI_NOTIFY_TSC_FIFO_ECC_SEC to allow for error injection. 1'b0 -> De-Asseert inject_error_67 output. **0** = DISABLE **1** = ENABLE
2	0x0	**ERR66**: 1'b1 -> Assert the inject_error_66 output for Other Corrected Error to USER_DEFINED_CORRECTED_ERR5 to allow for error injection. 1'b0 -> De-Asseert inject_error_66 output. **0** = DISABLE **1** = ENABLE
1	0x0	**ERR65**: 1'b1 -> Assert the inject_error_65 output for Other Uncorrected Error to USER_DEFINED_UNCORRECTED_ERR5 to allow for error injection. 1'b0 -> De-Asseert inject_error_65 output. **0** = DISABLE **1** = ENABLE
0	0x0	**ERR64**: 1'b1 -> Assert the inject_error_64 output for Other Corrected Error to USER_DEFINED_CORRECTED_ERR4 to allow for error injection. 1'b0 -> De-Asseert inject_error_64 output. **0** = DISABLE **1** = ENABLE

VI_EC_ERRSLICE2_LATENTERR_ENABLE_0

Offset: 0xc28
Byte Offset: 0x30a0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x0000001f (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1,1111)

Bit	Reset	Description
4	0x1	**ERR68**: 1'b1 -> Enable Latent Error Reporting for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_NOTIFY_TSC_FIFO_ECC_DED 1'b0 -> Disable Latent Error Reporting for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_NOTIFY_TSC_FIFO_ECC_DED **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
3	0x1	**ERR67**: 1'b1 -> Enable Latent Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_NOTIFY_TSC_FIFO_ECC_SEC 1'b0 -> Disable Latent Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_NOTIFY_TSC_FIFO_ECC_SEC **0** = DISABLE **1** = ENABLE
2	0x1	**ERR66**: 1'b1 -> Enable Latent Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR5 1'b0 -> Disable Latent Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR5 **0** = DISABLE **1** = ENABLE
1	0x1	**ERR65**: 1'b1 -> Enable Latent Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR5 1'b0 -> Disable Latent Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR5 **0** = DISABLE **1** = ENABLE
0	0x1	**ERR64**: 1'b1 -> Enable Latent Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR4 1'b0 -> Disable Latent Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR4 **0** = DISABLE **1** = ENABLE

VI_EC_ERRSLICE2_LATENTERR_FORCE_0

Offset: 0xc29
Byte Offset: 0x30a4
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**ERR68**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_NOTIFY_TSC_FIFO_ECC_DED 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
3	0x0	**ERR67**: 1'b1 -> Force Assertion of Latent Error Reporting for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_NOTIFY_TSC_FIFO_ECC_SEC 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
2	0x0	**ERR66**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR5 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR65**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR5 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR64**: 1'b1 -> Force Assertion of Latent Error Reporting for Other Corrected Error from USER_DEFINED_CORRECTED_ERR4 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

VI_EC_ERRSLICE2_LATENTERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of LatentError_Enable register, to avoid silent dropping of errors.

Offset: 0xc2a
Byte Offset: 0x30a8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**ERR68**: 1'b1 -> Error_68_pulse[1:0] for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_NOTIFY_TSC_FIFO_ECC_DED was equal to 2'b00 or 2'b11. 1'b0 -> Error_68_pulse[1:0] for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_NOTIFY_TSC_FIFO_ECC_DED was equal to 2'b01 or 2'b10, but no latent error.

Bit	Reset	Description
3	0x0	**ERR67**: 1'b1 -> Error_67_pulse[1:0] for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_NOTIFY_TSC_FIFO_ECC_SEC was equal to 2'b00 or 2'b11. 1'b0 -> Error_67_pulse[1:0] for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_NOTIFY_TSC_FIFO_ECC_SEC was equal to 2'b01 or 2'b10, but no latent error.
2	0x0	**ERR66**: 1'b1 -> Error_66_pulse[1:0] for Other Corrected Error from USER_DEFINED_CORRECTED_ERR5 was equal to 2'b00 or 2'b11. 1'b0 -> Error_66_pulse[1:0] for Other Corrected Error from USER_DEFINED_CORRECTED_ERR5 was equal to 2'b01 or 2'b10, but no latent error.
1	0x0	**ERR65**: 1'b1 -> Error_65_pulse[1:0] for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR5 was equal to 2'b00 or 2'b11. 1'b0 -> Error_65_pulse[1:0] for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR5 was equal to 2'b01 or 2'b10, but no latent error.
0	0x0	**ERR64**: 1'b1 -> Error_64_pulse[1:0] for Other Corrected Error from USER_DEFINED_CORRECTED_ERR4 was equal to 2'b00 or 2'b11. 1'b0 -> Error_64_pulse[1:0] for Other Corrected Error from USER_DEFINED_CORRECTED_ERR4 was equal to 2'b01 or 2'b10, but no latent error.

VI_EC_ERRSLICE2_COUNTER_RELOAD_0

Offset: 0xc2c
Byte Offset: 0x30b0
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**ERR68**: 1'b1 -> Reload Error Counter for ECC DED Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_NOTIFY_TSC_FIFO_ECC_DED 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
3	0x0	**ERR67**: 1'b1 -> Reload Error Counter for ECC SEC Error from on-chip SRAM/FIFO from FAULT_UDF_TAG_VI_NOTIFY_TSC_FIFO_ECC_SEC 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

Bit	Reset	Description
2	0x0	**ERR66**: 1'b1 -> Reload Error Counter for Other Corrected Error from USER_DEFINED_CORRECTED_ERR5 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
1	0x0	**ERR65**: 1'b1 -> Reload Error Counter for Other Uncorrected Error from USER_DEFINED_UNCORRECTED_ERR5 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
0	0x0	**ERR64**: 1'b1 -> Reload Error Counter for Other Corrected Error from USER_DEFINED_CORRECTED_ERR4 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

VI_EC_ERRSLICE2_MISSIONERR_ECC_CORRECTION_DIS_0

Offset: 0xc2d
Byte Offset: 0x30b4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0xxx)

Bit	Reset	Description
3	0x0	**ERR67**: 1'b1 -> Disables ECC correction in ECC Terminate plugins for FAULT_UDF_TAG_VI_NOTIFY_TSC_FIFO_ECC_SEC and shall report single bit errors over the ded_err_pulse to only allow detection and no correction. 1'b0 -> ECC correction is enabled in ECC terminate plugins for FAULT_UDF_TAG_VI_NOTIFY_TSC_FIFO_ECC_SEC and shall correct single bit errors and report them over sec_err_pulse **0** = DISABLE **1** = ENABLE

VI_CFG_VI_INCR_SYNCPT_0

Offset: 0x1000
Byte Offset: 0x4000
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:10	0x0	**VI_COND**: Condition mapped from raise/wait **0** = IMMEDIATE **1** = OP_DONE **2** = RD_DONE **3** = REG_WR_SAFE **4** = ENGINE_IDLE **5** = COND_5 **6** = COND_6 **7** = COND_7 **8** = COND_8 **9** = COND_9 **10** = COND_10 **11** = COND_11 **12** = COND_12 **13** = COND_13 **14** = COND_14 **15** = COND_15 **16** = COND_16 **17** = COND_17 **18** = COND_18 **19** = COND_19 **20** = COND_20 **21** = COND_21 **22** = COND_22 **23** = COND_23 **24** = COND_24 **25** = COND_25 **26** = COND_26 **27** = COND_27 **28** = COND_28 **29** = COND_29 **30** = COND_30 **31** = COND_31
9:0	0x0	**VI_INDX**: Syncpt index value

VI_CFG_VI_INCR_SYNCPT_CNTRL_0

Offset: 0x1001
Byte Offset: 0x4004
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,xxxx,xxx0)

Bit	Reset	Description
8	0x0	**VI_INCR_SYNCPT_NO_STALL**: If NO_STALL is 1, then when fifos are full, INCR_SYNCPT methods will be dropped and the INCR_SYNCPT_ERROR[COND] bit will be set. If NO_STALL is 0, then when fifos are full, the client host interface will be stalled.
0	0x0	**VI_INCR_SYNCPT_SOFT_RESET**: If SOFT_RESET is set, then all internal state of the client syncpt block will be reset. To do soft reset, first set SOFT_RESET of all Host Controller clients affected, then clear all SOFT_RESETs.

VI_CFG_VI_INCR_SYNCPT_ERROR_0

Offset: 0x1002
Byte Offset: 0x4008
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**VI_COND_STATUS**: COND_STATUS[COND] is set if the FIFO for COND overflows. This bit is sticky and will remain set until cleared. Cleared by writing 1.

VI_CFG_INTSTATUS_0

Offset: 0x1008
Byte Offset: 0x4020
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**CTXSW_INT**: Context switch interrupt status (clear on write)

VI_CFG_INTERRUPT_STATUS_0

Offset: 0x100d
Byte Offset: 0x4034
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xxxx,xx00,000x,00xx)

Bit	Reset	Description
29	0x0	**VGP6_INT_STATUS**: VGP6 pin Interrupt Status This bit controls interrupt when VGP6 rising/falling edge is detected. 0= Interrupt not detected 1= Interrupt detected **0** = NOINTR **1** = INTR
28	0x0	**VGP5_INT_STATUS**: VGP5 pin Interrupt Status This bit controls interrupt when VGP5 rising/falling edge is detected. 0= Interrupt not detected 1= Interrupt detected **0** = NOINTR **1** = INTR
27	0x0	**VGP4_INT_STATUS**: VGP4 pin Interrupt Status This bit controls interrupt when VGP4 rising/falling edge is detected. 0= Interrupt not detected 1= Interrupt detected **0** = NOINTR **1** = INTR
26	0x0	**VGP3_INT_STATUS**: VGP3 pin Interrupt Status This bit controls interrupt when VGP3 rising/falling edge is detected. 0= Interrupt not detected 1= Interrupt detected **0** = NOINTR **1** = INTR
25	0x0	**VGP2_INT_STATUS**: VGP2 pin Interrupt Status This bit controls interrupt when VGP2 rising/falling edge is detected. 0= Interrupt not detected 1= Interrupt detected **0** = NOINTR **1** = INTR

Bit	Reset	Description
24	0x0	**VGP1_INT_STATUS**: VGP1 pin Interrupt Status This bit controls interrupt when VGP1 rising/falling edge is detected. 0= Interrupt not detected 1= Interrupt detected **0** = NOINTR **1** = INTR
23	0x0	**CSBMASTER_CFGREG_REG_PARITY_ERR_STATUS**: **0** = NOINTR **1** = INTR
22	0x0	**CSIMUX_REG_PARITY_ERR_STATUS**: **0** = NOINTR **1** = INTR
21	0x0	**CHANSEL_SHADOW_REG_PARITY_ERR_STATUS**: **0** = NOINTR **1** = INTR
20	0x0	**CHANSEL_DIRECT_REG_PARITY_ERR_STATUS**: **0** = NOINTR **1** = INTR
19	0x0	**ATOMP_REG_PARITY_ERR_STATUS**: **0** = NOINTR **1** = INTR
18	0x0	**PIXFMT_CFG_REG_PARITY_ERR_STATUS**: **0** = NOINTR **1** = INTR
17	0x0	**PIXFMT_REG_PARITY_ERR_STATUS**: **0** = NOINTR **1** = INTR
16	0x0	**CHANCMD_REG_PARITY_ERR_STATUS**: **0** = NOINTR **1** = INTR

Bit	Reset	Description
9	0x0	**ATOMP_TOSSED_ERR_STATUS**: ATOMP Tossed happened **0** = NOINTR **1** = INTR
8	0x0	**ATOMP_TRUNCATED_ERR_STATUS**: ATOMP Truncated happened **0** = NOINTR **1** = INTR
7	0x0	**HOST_PKTINJECT_STALL_ERR_STATUS**: PKTINEJCT FIFO overflow **0** = NOINTR **1** = INTR
6	0x0	**CSIMUX_FIFO_OVFL_ERR_STATUS**: CSIMUX async FIFO overflow (OR of all 6 streams FIFO) **0** = NOINTR **1** = INTR
5	0x0	**ATOMP_PACKER_OVFL_ERR_STATUS**: ATOMP BackEnd FIFO overflow (OR of all 18 BE fifos) **0** = NOINTR **1** = INTR
3	0x0	**NOTIFY_FIFO_OVFL_ERR_STATUS**: Set by Hardware when any of the NOTIFY input FIFO has overflowed. This basically means that Real Time CPU has not been able to keep up its pace with the notify events arrival speed. **0** = NOINTR **1** = INTR
2	0x0	**NOTIFY_TIMESTAMP_OVFL_ERR_STATUS**: any event of notify infifo is read out after 8 ms this causes wrong timestamp read out check VI_NOTIFY_INFIFO_TIMESTAMP_OVRFLOW_ERR_STATUS_0 for more details **0** = NOINTR **1** = INTR

VI_CFG_INTERRUPT_MASK_0

Offset: 0x100e
Byte Offset: 0x4038
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xxxx,xx00,000x,00xx)

Bit	Reset	Description
29	0x0	**VGP6_INT_MASK**: VGP6 pin Interrupt Mask This bit controls interrupt when VGP6 rising/falling edge is detected. 0= Disabled 1= Enabled **0** = DISABLED **1** = ENABLED
28	0x0	**VGP5_INT_MASK**: VGP5 pin Interrupt Mask This bit controls interrupt when VGP5 rising/falling edge is detected. 0= Disabled 1= Enabled **0** = DISABLED **1** = ENABLED
27	0x0	**VGP4_INT_MASK**: VGP4 pin Interrupt Mask This bit controls interrupt when VGP4 rising/falling edge is detected. 0= Disabled 1= Enabled **0** = DISABLED **1** = ENABLED
26	0x0	**VGP3_INT_MASK**: VGP3 pin Interrupt Mask This bit controls interrupt when VGP3 rising/falling edge is detected. 0= Disabled 1= Enabled **0** = DISABLED **1** = ENABLED
25	0x0	**VGP2_INT_MASK**: VGP2 pin Interrupt Mask This bit controls interrupt when VGP2 rising/falling edge is detected. 0= Disabled 1= Enabled **0** = DISABLED **1** = ENABLED
24	0x0	**VGP1_INT_MASK**: VGP1 pin Interrupt Mask This bit controls interrupt when VGP1 rising/falling edge is detected. 0= Disabled 1= Enabled **0** = DISABLED **1** = ENABLED

Bit	Reset	Description
23	0x0	**CSBMASTER_CFGREG_REG_PARITY_ERR_MASK**: 0 = Disabled 1 = Enabled **0** = DISABLED **1** = ENABLED
22	0x0	**CSIMUX_REG_PARITY_ERR_MASK**: 0 = Disabled 1 = Enabled **0** = DISABLED **1** = ENABLED
21	0x0	**CHANSEL_SHADOW_REG_PARITY_ERR_MASK**: 0 = Disabled 1 = Enabled **0** = DISABLED **1** = ENABLED
20	0x0	**CHANSEL_DIRECT_REG_PARITY_ERR_MASK**: 0 = Disabled 1 = Enabled **0** = DISABLED **1** = ENABLED
19	0x0	**ATOMP_REG_PARITY_ERR_MASK**: 0 = Disabled 1 = Enabled **0** = DISABLED **1** = ENABLED
18	0x0	**PIXFMT_CFG_REG_PARITY_ERR_MASK**: 0 = Disabled 1 = Enabled **0** = DISABLED **1** = ENABLED
17	0x0	**PIXFMT_REG_PARITY_ERR_MASK**: 0 = Disabled 1 = Enabled **0** = DISABLED **1** = ENABLED
16	0x0	**CHANCMD_REG_PARITY_ERR_MASK**: 0 = Disabled 1 = Enabled **0** = DISABLED **1** = ENABLED
9	0x0	**ATOMP_TOSSED_ERR_MASK**: ATOMP Tossed happened 0 = Disabled 1 = Enabled **0** = DISABLED **1** = ENABLED

Bit	Reset	Description
8	0x0	**ATOMP_TRUNCATED_ERR_MASK**: ATOMP Truncated happened 0 = Disabled 1 = Enabled **0** = DISABLED **1** = ENABLED
7	0x0	**HOST_PKTINJECT_STALL_ERR_MASK**: Interrupt Mask for Host Controller packet inject stall error 0 = Disabled 1 = Enabled **0** = DISABLED **1** = ENABLED
6	0x0	**CSIMUX_FIFO_OVFL_ERR_MASK**: Interrupt Mask for CSIMUX Fifo overflow 0 = Disabled 1 = Enabled **0** = DISABLED **1** = ENABLED
5	0x0	**ATOMP_PACKER_OVFL_ERR_MASK**: Interrupt Mask for Atomp packer FIFO Overflow 0 = Disabled 1 = Enabled **0** = DISABLED **1** = ENABLED
3	0x0	**NOTIFY_FIFO_OVFL_ERR_MASK**: Interrupt Mask for Notify FIFO Overflow This bit enable generation of VI Interrupt when Notify FIFO overflow is asserted 0 = Disabled 1 = Enabled **0** = DISABLED **1** = ENABLED
2	0x0	**NOTIFY_TIMESTAMP_OVFL_ERR_MASK**: 0 = Disabled 1 = Enabled **0** = DISABLED **1** = ENABLED

VI_CFG_PWM_CONTROL_0

VI Pulse Width Modulation Control

PWM signal generation logic can generate up to 128 pulses per line internally and the PWM pulse select registers determines which of the 128 pulses will be output. Any of the 128 internally generated pulse can be independently selected as output if they occur within one line time. PWM

signal can be output on the VGP6 pin if VGP6 output is enabled and the output select is set to PWM. The PWM will be triggered by the first vsync after the PWM_ENABLE bit has been set.

Offset: 0x100f
Byte Offset: 0x403c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,xx00,xxxx,xxxx,xxxx,xxx0,xxx0)

Bit	Reset	Description
31:24	0x0	**PWM_COUNTER**: PWM Counter 8-bit value used when PWM_MODE is set to COUNTER to determine how many times the PWM will cycle through the 128 cycles before stopping.
21:20	0x0	**PWM_MODE**: PWM Mode Continous - after PWM is turned on, continue through the PWM's 128 cycles repeatedly until the pwm is turned off. Single - after PWM is turned on, cycle once through the 128 cycles and stop. Counter - after PWM is turned on, cycle through the 128 cycles PWM_COUNTER number of times then stop. 23:22 reserved **0** = CONTINUOUS **1** = SINGLE **2** = COUNTER
4	0x0	**PWM_DIRECTION**: PWM Direction 0= Incrementing 1= Decrementing **0** = INCR **1** = DECR
0	0x0	**PWM_ENABLE**: PWM Enable 0= Disabled 1= Enabled **0** = DISABLED **1** = ENABLED

VI_CFG_PWM_HIGH_PULSE_0

PWM High Pulse Period

Offset: 0x1010
Byte Offset: 0x4040
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PWM_HIGH_PULSE**: PWM High Pulse

VI_CFG_PWM_LOW_PULSE_0

PWM Low Pulse Period

Offset: 0x1011
Byte Offset: 0x4044
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PWM_LOW_PULSE**: PWM Low Pulse

VI_CFG_PWM_SELECT_PULSE_A_0

PWM Pulse Select A

The next 4 registers select which of the internal 128 pulses to be output. Each bit in the four registers correspond to one internal pulse.

Offset: 0x1012
Byte Offset: 0x4048
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**PWM_SELECT_A**: PWM Select bits 31 to 0

VI_CFG_PWM_SELECT_PULSE_B_0

PWM Pulse Select B

Offset: 0x1013
Byte Offset: 0x404c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**PWM_SELECT_B**: PWM Select bits 63 to 32

VI_CFG_PWM_SELECT_PULSE_C_0

PWM Pulse Select C

Offset: 0x1014
Byte Offset: 0x4050
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**PWM_SELECT_C**: PWM Select bits 95 to 64

VI_CFG_PWM_SELECT_PULSE_D_0

PWM Pulse Select D

Offset: 0x1015
Byte Offset: 0x4054

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**PWM_SELECT_D**: PWM Select bits 127 to 96

VI_CFG_VGP1_0

VI VGP1 input/output config/data

Offset: 0x101a
Byte Offset: 0x4068
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000X0X (0bxxxx,xxx0,xxxx,0000,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
24	RW	0x0	**VGP1_INPUT_ENABLE**: VGP1 pin Input Enable This bit is unused **0** = DISABLED **1** = ENABLED
19:17	RW	0x0	**PIN_OUTPUT_SELECT_VGP1**: Pin Output Select VGP1 0 = VGP1_OUTPUT_DATA 1 = PWM 2 = XVS0 3 = XHS0 4 = XVS1 5 = XHS1 6 = XVS2 7 = XHS2
16	RW	0x0	**VGP1_OUTPUT_ENABLE**: VGP1 pin Output Enable This bit controls VGP1 pin output. 0= Disabled 1= Enabled **0** = DISABLED **1** = ENABLED

Bit	R/W	Reset	Description
8	RO	X	**VGP1_INPUT_DATA**: VGP1 pin Input Data (effective if VGP1_INPUT_ENABLE is ENABLED) 0= VGP1 input low 1= VGP1 input high
0	RW	X	**VGP1_OUTPUT_DATA**: VGP1 pin Output Data (effective if VGP1_OUTPUT_ENABLE is ENABLED and VGP1_OUTPUT_SELECT is DATA)

VI_CFG_VGP2_0

VI VGP2 input/output config/data

Offset: 0x101b
Byte Offset: 0x406c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000X0X (0bxxxx,xxx0,xxxx,0000,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
24	RW	0x0	**VGP2_INPUT_ENABLE**: VGP2 pin Input Enable This bit is unused **0** = DISABLED **1** = ENABLED
19:17	RW	0x0	**PIN_OUTPUT_SELECT_VGP2**: Pin Output Select VGP2 0 = VGP2_OUTPUT_DATA 1 = PWM 2 = XVS0 3 = XHS0 4 = XVS1 5 = XHS1 6 = XVS2 7 = XHS2
16	RW	0x0	**VGP2_OUTPUT_ENABLE**: VGP2 pin Output Enable This bit controls VGP2 pin output. 0= Disabled 1= Enabled **0** = DISABLED **1** = ENABLED

Bit	R/W	Reset	Description
8	RO	X	**VGP2_INPUT_DATA**: VGP2 pin Input Data (effective if VGP2_INPUT_ENABLE is ENABLED) 0= VGP2 input low 1= VGP2 input high
0	RW	X	**VGP2_OUTPUT_DATA**: VGP2 pin Output Data (effective if VGP2_OUTPUT_ENABLE is ENABLED and VGP2_OUTPUT_SELECT is DATA)

VI_CFG_VGP3_0

VI VGP3 input/output config/data

Offset: 0x101c
Byte Offset: 0x4070
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000X0X (0bxxxx,xxx0,xxxx,0000,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
24	RW	0x0	**VGP3_INPUT_ENABLE**: VGP3 pin Input Enable This bit is unused **0** = DISABLED **1** = ENABLED
19:17	RW	0x0	**PIN_OUTPUT_SELECT_VGP3**: Pin Output Select VGP3 0 = VGP3_OUTPUT_DATA 1 = PWM 2 = XVS0 3 = XHS0 4 = XVS1 5 = XHS1 6 = XVS2 7 = XHS2
16	RW	0x0	**VGP3_OUTPUT_ENABLE**: VGP3 pin Output Enable This bit controls VGP3 pin output. 0= Disabled 1= Enabled **0** = DISABLED **1** = ENABLED

Bit	R/W	Reset	Description
8	RO	X	**VGP3_INPUT_DATA**: VGP3 pin Input Data (effective if VGP3_INPUT_ENABLE is ENABLED) 0= VGP3 input low 1= VGP3 input high
0	RW	X	**VGP3_OUTPUT_DATA**: VGP3 pin Output Data (effective if VGP3_OUTPUT_ENABLE is ENABLED and VGP3_OUTPUT_SELECT is DATA)

VI_CFG_VGP4_0

VI VGP4 input/output config/data

Offset: 0x101d
Byte Offset: 0x4074
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000X0X (0bxxxx,xxx0,xxxx,0000,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
24	RW	0x0	**VGP4_INPUT_ENABLE**: VGP4 pin Input Enable This bit is unused **0** = DISABLED **1** = ENABLED
19:17	RW	0x0	**PIN_OUTPUT_SELECT_VGP4**: Pin Output Select VGP4 0 = VGP4_OUTPUT_DATA 1 = PWM 2 = XVS0 3 = XHS0 4 = XVS1 5 = XHS1 6 = XVS2 7 = XHS2
16	RW	0x0	**VGP4_OUTPUT_ENABLE**: VGP4 pin Output Enable This bit controls VGP4 pin output. 0= Disabled 1= Enabled **0** = DISABLED **1** = ENABLED

Bit	R/W	Reset	Description
8	RO	X	**VGP4_INPUT_DATA**: VGP4 pin Input Data (effective if VGP4_INPUT_ENABLE is ENABLED) 0= VGP4 input low 1= VGP4 input high
0	RW	X	**VGP4_OUTPUT_DATA**: VGP4 pin Output Data (effective if VGP4_OUTPUT_ENABLE is ENABLED and VGP4_OUTPUT_SELECT is DATA)

VI_CFG_VGP5_0

VI VGP5 input/output config/data

Offset: 0x101e
Byte Offset: 0x4078
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000X0X (0bxxxx,xxx0,xxxx,0000,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
24	RW	0x0	**VGP5_INPUT_ENABLE**: VGP5 pin Input Enable This bit is unused **0** = DISABLED **1** = ENABLED
19:17	RW	0x0	**PIN_OUTPUT_SELECT_VGP5**: Pin Output Select VGP5 0 = VGP5_OUTPUT_DATA 1 = PWM 2 = XVS0 3 = XHS0 4 = XVS1 5 = XHS1 6 = XVS2 7 = XHS2
16	RW	0x0	**VGP5_OUTPUT_ENABLE**: VGP5 pin Output Enable This bit controls VGP5 pin output. 0= Disabled 1= Enabled **0** = DISABLED **1** = ENABLED

Bit	R/W	Reset	Description
8	RO	X	**VGP5_INPUT_DATA**: VGP5 pin Input Data (effective if VGP5_INPUT_ENABLE is ENABLED) 0= VGP5 input low 1= VGP5 input high
0	RW	X	**VGP5_OUTPUT_DATA**: VGP5 pin Output Data (effective if VGP5_OUTPUT_ENABLE is ENABLED and VGP5_OUTPUT_SELECT is DATA)

VI_CFG_VGP6_0

VI VGP6 input/output config/data

Offset: 0x101f
Byte Offset: 0x407c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000X0X (0bxxxx,xxx0,xxxx,0000,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
24	RW	0x0	**VGP6_INPUT_ENABLE**: VGP6 pin Input Enable This bit is unused **0** = DISABLED **1** = ENABLED
19:17	RW	0x0	**PIN_OUTPUT_SELECT_VGP6**: Pin Output Select VGP6 0 = VGP6_OUTPUT_DATA 1 = PWM 2 = XVS0 3 = XHS0 4 = XVS1 5 = XHS1 6 = XVS2 7 = XHS2
16	RW	0x0	**VGP6_OUTPUT_ENABLE**: VGP6 pin Output Enable This bit controls VGP6 pin output. 0= Disabled 1= Enabled **0** = DISABLED **1** = ENABLED

Bit	R/W	Reset	Description
8	RO	X	**VGP6_INPUT_DATA**: VGP6 pin Input Data (effective if VGP6_INPUT_ENABLE is ENABLED) 0= VGP6 input low 1= VGP6 input high
0	RW	X	**VGP6_OUTPUT_DATA**: VGP6 pin Output Data (effective if VGP6_OUTPUT_ENABLE is ENABLED and VGP6_OUTPUT_SELECT is DATA)

VI_CFG_INTERRUPT_TYPE_SELECT_0

Interrupt Type Select

Offset: 0x1020
Byte Offset: 0x4080
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000,000x)

Bit	Reset	Description
6	0x0	**VGP6_INT_TYPE**: VGP6 pin Interrupt Type This bit controls interrupt VGP6 0= Edge type 1= Level type **0** = EDGE **1** = LEVEL
5	0x0	**VGP5_INT_TYPE**: VGP5 pin Interrupt Type This bit controls interrupt VGP5 0= Edge type 1= Level type **0** = EDGE **1** = LEVEL
4	0x0	**VGP4_INT_TYPE**: VGP4 pin Interrupt Type This bit controls interrupt VGP4 0= Edge type 1= Level type **0** = EDGE **1** = LEVEL

Bit	Reset	Description
3	0x0	**VGP3_INT_TYPE**: VGP3 pin Interrupt Type This bit controls interrupt VGP3 0= Edge type 1= Level type **0** = EDGE **1** = LEVEL
2	0x0	**VGP2_INT_TYPE**: VGP2 pin Interrupt Type This bit controls interrupt VGP2 0= Edge type 1= Level type **0** = EDGE **1** = LEVEL
1	0x0	**VGP1_INT_TYPE**: VGP1 pin Interrupt Type This bit controls interrupt VGP1 0= Edge type 1= Level type **0** = EDGE **1** = LEVEL

VI_CFG_INTERRUPT_POLARITY_SELECT_0

Interrupt Polarity Select

Offset: 0x1021
Byte Offset: 0x4084
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000,000x)

Bit	Reset	Description
6	0x0	**VGP6_INT_POLARITY**: VGP6 pin Interrupt Type This bit controls interrupt VGP6 0= falling edge or low level 1= rising edge or high level **0** = LOW **1** = HIGH

Bit	Reset	Description
5	0x0	**VGP5_INT_POLARITY**: VGP5 pin Interrupt Type This bit controls interrupt VGP5 0= falling edge or low level 1= rising edge or high level **0** = LOW **1** = HIGH
4	0x0	**VGP4_INT_POLARITY**: VGP4 pin Interrupt Type This bit controls interrupt VGP4 0= falling edge or low level 1= rising edge or high level **0** = LOW **1** = HIGH
3	0x0	**VGP3_INT_POLARITY**: VGP3 pin Interrupt Type This bit controls interrupt VGP3 0= falling edge or low level 1= rising edge or high level **0** = LOW **1** = HIGH
2	0x0	**VGP2_INT_POLARITY**: VGP2 pin Interrupt Type This bit controls interrupt VGP2 0= falling edge or low level 1= rising edge or high level **0** = LOW **1** = HIGH
1	0x0	**VGP1_INT_POLARITY**: VGP1 pin Interrupt Type This bit controls interrupt VGP1 0= falling edge or low level 1= rising edge or high level **0** = LOW **1** = HIGH

VI_CFG_VGP_SYNCPT_CONFIG_0

Offset: 0x1022
Byte Offset: 0x4088
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000,x000)

Bit	Reset	Description
6:4	0x0	**SYNCPT_VGPY_SELECT**: Select the VGP (1-6) for SYNCPT condition VGP_1_RECD
2:0	0x0	**SYNCPT_VGPX_SELECT**: Select the VGP (1-6) for SYNCPT condition VGP_0_RECD

VI_CFG_SLCG_CTRL_0_0

Offset: 0x1028
Byte Offset: 0x40a0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x00x,0000,00xx,xxxx)

Bit	Reset	Description
14	0x0	**NVCSI_SWIZZLE_CLK_OVERRIDE**: Second level clock gating override for nvcsi_swizzle_clk 0: Enable second level clock gating 1: Disable second level clock gating **0** = DISABLE **1** = ENABLE
13	0x0	**VI_FSTAT_CLK_OVERRIDE**: Second level clock gating override for vi_fstat_clk and nvcsi_fstat_clk 0: Enable second level clock gating 1: Disable second level clock gating **0** = DISABLE **1** = ENABLE
11	0x0	**VI_PIXFMT_CLK_OVERRIDE**: Second level clock gating override for vi_pixfmt_clk 0: Enable second level clock gating 1: Disable second level clock gating **0** = DISABLE **1** = ENABLE
10	0x0	**VI_NOTIFY_CLK_OVERRIDE**: Second level clock gating override for vi_notify_clk 0: Enable second level clock gating 1: Disable second level clock gating **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
9	0x0	**VI_ATOMP_CLK_OVERRIDE**: Second level clock gating override for vi_atomp_clk 0: Enable second level clock gating 1: Disable second level clock gating **0** = DISABLE **1** = ENABLE
8	0x0	**VI_PWM_CLK_OVERRIDE**: Second level clock gating override for vi_pwm_clk 0: Enable second level clock gating 1: Disable second level clock gating **0** = DISABLE **1** = ENABLE
7	0x0	**VI_CHANSEL_CLK_OVERRIDE**: Second level clock gating override for vi_pwm_clk 0: Enable second level clock gating 1: Disable second level clock gating **0** = DISABLE **1** = ENABLE
6	0x0	**VI_CSIMUX_CLK_OVERRIDE**: Second level clock gating override for vi_pwm_clk 0: Enable second level clock gating 1: Disable second level clock gating **0** = DISABLE **1** = ENABLE

VI_CFG_SLCG_CTRL_1_0

Offset: 0x1029
Byte Offset: 0x40a4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**VI_CLK_OVR_ON**: VI Global Second level clock gating override 0: Enable all second level clock gating (assuming the individual OVR_ON is disabled as well) 1: Disable all second level clock gating **0** = DISABLE **1** = ENABLE

VI_CFG_READY_FOR_LATENCY_0

Offset: 0x102a
Byte Offset: 0x40a8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxx0,0000,0000,0000,0000,0101)

Bit	Reset	Description
20:11	0x0	**WAIT_FOR_BLANK**: Additional control for DVFS; if set, ready_for_latency is asserted only if COUNT_THRESHOLD is met and also all channels at CHANSEL are not in frame and is asserted for a limited time (the value of this field, in units of 2^8 VICLKs).
10:0	0x5	**COUNT_THRESHOLD**: Indicates that VI is ready for latency event if number of entries in the FIFO is less than COUNT_THRESHOLD

VI_CFG_MEMORY_BATCH_MASK_0

Offset: 0x102b
Byte Offset: 0x40ac
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xfffffc00 (0b1111,1111,1111,1111,1111,1100,0000,0000)

Bit	Reset	Description
31:0	0xfffffc00	**BATCH_MASK**: Specify when to flush the atom out to memory. In order to improve MC performace, before line break ATOMP always tries to buffer more atomps before it writes to MC until below equation satisfied. (cur_addr[39:0] & {8\'hff, batch_flush_mask[31:0]} ! = next_addr[39:0] & {8\'hff, batch_flush_mask[31:0]}) default value is 0xFFFFFC00 to make it flush out @ 1KB boundary

VI_CFG_PPC_CALCULATION_0

Offset: 0x102c
Byte Offset: 0x40b0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**USE_NVCSI2VI_BUS**: previously VI HW relied on data type to determine whether the current packet is 8ppc or 4ppc; there is a newly added bit on nvcsi->vi bus to indicate if the packet is 8ppc / 4ppc to avoid some DT overriding case this registers controls whether VI HW uses newly added bit to determine 8ppc/4ppc or still use legacy way which is DT based

VI_CFG_REGISTER_DEBUG_ACCESS_0

Offset: 0x102d
Byte Offset: 0x40b4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug) This is for those double buffered register which do not belong any channel space for example, FMLITE configuration registers
0	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug) This is for those double buffered register which do not belong any channel space for example, FMLITE configuration registers

VI_CFG_RESERVE_0_0

Offset: 0x102e
Byte Offset: 0x40b8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000XXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
15:12	X	**nc_RESERVE_0_3**
11:8	X	**nc_RESERVE_0_2**:

Bit	Reset	Description
7:4	X	**nc_RESERVE_0_1**:
3:0	X	**nc_RESERVE_0_0**:

VI_CFG_RESERVE_1_0

Offset: 0x102f
Byte Offset: 0x40bc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000XXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
15:12	X	**nc_RESERVE_1_3**
11:8	X	**nc_RESERVE_1_2**
7:4	X	**nc_RESERVE_1_1**:
3:0	X	**nc_RESERVE_1_0**:

VI_CFG_INTERRUPT_PIPELINE_STATUS_0

Interrupt Status Register for VI pipeline

Offset: 0x1030
Byte Offset: 0x40c0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
24	0x0	**ATOMP_DVFSFIFO_ADR_RAM_PARITY_ERR_STATUS**: **0** = NOINTR **1** = INTR

Bit	Reset	Description
23	0x0	**OFIF_FIFO_CTRL_ECC_DED_STATUS**: **0** = NOINTR **1** = INTR
22	0x0	**OFIF_FIFO_CTRL_ECC_SEC_STATUS**: **0** = NOINTR **1** = INTR
21	0x0	**ATOMP_STREAM5_CRC_ERR_STATUS**: **0** = NOINTR **1** = INTR
20	0x0	**ATOMP_STREAM4_CRC_ERR_STATUS**: **0** = NOINTR **1** = INTR
19	0x0	**ATOMP_STREAM3_CRC_ERR_STATUS**: **0** = NOINTR **1** = INTR
18	0x0	**ATOMP_STREAM2_CRC_ERR_STATUS**: **0** = NOINTR **1** = INTR
17	0x0	**ATOMP_STREAM1_CRC_ERR_STATUS**: **0** = NOINTR **1** = INTR
16	0x0	**ATOMP_STREAM0_CRC_ERR_STATUS**: **0** = NOINTR **1** = INTR
14	0x0	**PIXFMT_CTRL_ECC_DED_STATUS**: **0** = NOINTR **1** = INTR
13	0x0	**PIXFMT_CTRL_ECC_SEC_STATUS**: **0** = NOINTR **1** = INTR
12	0x0	**CHANSEL_DATA_PARITY_ERR_STATUS**: **0** = NOINTR **1** = INTR
11	0x0	**CSIMUX_STREAM5_ECC_DED_STATUS**: **0** = NOINTR **1** = INTR

Bit	Reset	Description
10	0x0	**CSIMUX_STREAM5_ECC_SEC_STATUS**: **0** = NOINTR **1** = INTR
9	0x0	**CSIMUX_STREAM4_ECC_DED_STATUS**: **0** = NOINTR **1** = INTR
8	0x0	**CSIMUX_STREAM4_ECC_SEC_STATUS**: **0** = NOINTR **1** = INTR
7	0x0	**CSIMUX_STREAM3_ECC_DED_STATUS**: **0** = NOINTR **1** = INTR
6	0x0	**CSIMUX_STREAM3_ECC_SEC_STATUS**: **0** = NOINTR **1** = INTR
5	0x0	**CSIMUX_STREAM2_ECC_DED_STATUS**: **0** = NOINTR **1** = INTR
4	0x0	**CSIMUX_STREAM2_ECC_SEC_STATUS**: **0** = NOINTR **1** = INTR
3	0x0	**CSIMUX_STREAM1_ECC_DED_STATUS**: **0** = NOINTR **1** = INTR
2	0x0	**CSIMUX_STREAM1_ECC_SEC_STATUS**: **0** = NOINTR **1** = INTR
1	0x0	**CSIMUX_STREAM0_ECC_DED_STATUS**: **0** = NOINTR **1** = INTR
0	0x0	**CSIMUX_STREAM0_ECC_SEC_STATUS**: **0** = NOINTR **1** = INTR

VI_CFG_INTERRUPT_PIPELINE_MASK_0

Interrupt Mask

Offset: 0x1031
Byte Offset: 0x40c4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
24	0x0	**ATOMP_DVFSFIFO_ADR_RAM_PARITY_ERR_MASK**: **0** = DISABLED **1** = ENABLED
23	0x0	**OFIF_FIFO_CTRL_ECC_DED_MASK**: **0** = DISABLED **1** = ENABLED
22	0x0	**OFIF_FIFO_CTRL_ECC_SEC_MASK**: **0** = DISABLED **1** = ENABLED
21	0x0	**ATOMP_STREAM5_CRC_ERR_MASK**: **0** = DISABLED **1** = ENABLED
20	0x0	**ATOMP_STREAM4_CRC_ERR_MASK**: **0** = DISABLED **1** = ENABLED
19	0x0	**ATOMP_STREAM3_CRC_ERR_MASK**: **0** = DISABLED **1** = ENABLED
18	0x0	**ATOMP_STREAM2_CRC_ERR_MASK**: **0** = DISABLED **1** = ENABLED
17	0x0	**ATOMP_STREAM1_CRC_ERR_MASK**: **0** = DISABLED **1** = ENABLED
16	0x0	**ATOMP_STREAM0_CRC_ERR_MASK**: **0** = DISABLED **1** = ENABLED
14	0x0	**PIXFMT_CTRL_ECC_DED_MASK**: **0** = DISABLED **1** = ENABLED
13	0x0	**PIXFMT_CTRL_ECC_SEC_MASK**: **0** = DISABLED **1** = ENABLED
12	0x0	**CHANSEL_DATA_PARITY_ERR_MASK**: **0** = DISABLED **1** = ENABLED

Bit	Reset	Description
11	0x0	**CSIMUX_STREAM5_ECC_DED_MASK**: **0** = DISABLED **1** = ENABLED
10	0x0	**CSIMUX_STREAM5_ECC_SEC_MASK**: **0** = DISABLED **1** = ENABLED
9	0x0	**CSIMUX_STREAM4_ECC_DED_MASK**: **0** = DISABLED **1** = ENABLED
8	0x0	**CSIMUX_STREAM4_ECC_SEC_MASK**: **0** = DISABLED **1** = ENABLED
7	0x0	**CSIMUX_STREAM3_ECC_DED_MASK**: **0** = DISABLED **1** = ENABLED
6	0x0	**CSIMUX_STREAM3_ECC_SEC_MASK**: **0** = DISABLED **1** = ENABLED
5	0x0	**CSIMUX_STREAM2_ECC_DED_MASK**: **0** = DISABLED **1** = ENABLED
4	0x0	**CSIMUX_STREAM2_ECC_SEC_MASK**: **0** = DISABLED **1** = ENABLED
3	0x0	**CSIMUX_STREAM1_ECC_DED_MASK**: **0** = DISABLED **1** = ENABLED
2	0x0	**CSIMUX_STREAM1_ECC_SEC_MASK**: **0** = DISABLED **1** = ENABLED
1	0x0	**CSIMUX_STREAM0_ECC_DED_MASK**: **0** = DISABLED **1** = ENABLED
0	0x0	**CSIMUX_STREAM0_ECC_SEC_MASK**: **0** = DISABLED **1** = ENABLED

VI_CFG_INTERRUPT_PIPELINE_EXT_STATUS_0

Interrupt Status Register for VI MISC

Offset: 0x1032
Byte Offset: 0x40c8
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOTIFY_TSC_ECC_DED_STATUS**: **0** = NOINTR **1** = INTR
8	0x0	**NOTIFY_TSC_ECC_SEC_STATUS**: **0** = NOINTR **1** = INTR
7	0x0	**HOST_SHIM_HRD_FIFO_ECC_DED_STATUS**: **0** = NOINTR **1** = INTR
6	0x0	**HOST_SHIM_HRD_FIFO_ECC_SEC_STATUS**: **0** = NOINTR **1** = INTR
5	0x0	**HOST_SHIM_HWR_FIFO_ECC_DED_STATUS**: **0** = NOINTR **1** = INTR
4	0x0	**HOST_SHIM_HWR_FIFO_ECC_SEC_STATUS**: **0** = NOINTR **1** = INTR
3	0x0	**NOTIFY_FIFO_ECC_DED_STATUS**: **0** = NOINTR **1** = INTR
2	0x0	**NOTIFY_FIFO_ECC_SEC_STATUS**: **0** = NOINTR **1** = INTR
1	0x0	**ATOMP_WR_ACK_FIFO_ECC_DED_STATUS**: **0** = NOINTR **1** = INTR
0	0x0	**ATOMP_WR_ACK_FIFO_ECC_SEC_STATUS**: **0** = NOINTR **1** = INTR

VI_CFG_INTERRUPT_PIPELINE_EXT_MASK_0

Interrupt Mask

Offset: 0x1033
Byte Offset: 0x40cc
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOTIFY_TSC_ECC_DED_MASK**: **0** = DISABLED **1** = ENABLED
8	0x0	**NOTIFY_TSC_ECC_SEC_MASK**: **0** = DISABLED **1** = ENABLED
7	0x0	**HOST_SHIM_HRD_FIFO_ECC_DED_MASK**: **0** = DISABLED **1** = ENABLED
6	0x0	**HOST_SHIM_HRD_FIFO_ECC_SEC_MASK**: **0** = DISABLED **1** = ENABLED
5	0x0	**HOST_SHIM_HWR_FIFO_ECC_DED_MASK**: **0** = DISABLED **1** = ENABLED
4	0x0	**HOST_SHIM_HWR_FIFO_ECC_SEC_MASK**: **0** = DISABLED **1** = ENABLED
3	0x0	**NOTIFY_FIFO_ECC_DED_MASK**: **0** = DISABLED **1** = ENABLED
2	0x0	**NOTIFY_FIFO_ECC_SEC_MASK**: **0** = DISABLED **1** = ENABLED
1	0x0	**ATOMP_WR_ACK_FIFO_ECC_DED_MASK**: **0** = DISABLED **1** = ENABLED
0	0x0	**ATOMP_WR_ACK_FIFO_ECC_SEC_MASK**: **0** = DISABLED **1** = ENABLED

VI_CFG_SW_ERR_CODE_0

SW Error Reporting

In certain cases, the IP driver may implement diagnostics in SW. Errors detected by such SW diagnostics need to be reported to HSM. To avoid time consuming IPCs, IP SW can directly write an IP SW defined Error Code into the SW_ERR_CODE register and assert the SW Error. The IP Error Collator adds an additional error input with the below options:

err_type = SW

err_user - Adds err__user [31:0] as an input to the error collator for the err_x.

Offset: 0x1034
Byte Offset: 0x40d0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SW_ERR_CODE**: SW Defined Error Code. Bits from this register connect to the IPs Error Collator err_<x>_user[31:0] signal.

VI_CFG_SW_ERR_ASSERT_0

Offset: 0x1035
Byte Offset: 0x40d4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SW_ERR_ASSERT**: When written with value 1'b1, this self clearing bit generates a pulse of 1 clock cycle from the unit. This pulse is dual_rail_converted by the IP {bit,~bit} and connected to the Error Collator err_x

VI_CFG_USER_DEFINED_UNCORRECTED_ERR_TRIGGER_0

Error Diagnose

Offset: 0x1036
Byte Offset: 0x40d8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**UNCORRECTED_ERR5**: When written with value 1, this self clearing bit generates a pulse of 1 clock cycle. This pulse is dual_rail_converted by the IP {bit,~bit} and connected to the Error Collator USER_DEFINED_UNCORRECTED_ERR5.
4	0x0	**UNCORRECTED_ERR4**: When written with value 1, this self clearing bit generates a pulse of 1 clock cycle. This pulse is dual_rail_converted by the IP {bit,~bit} and connected to the Error Collator USER_DEFINED_UNCORRECTED_ERR4.
3	0x0	**UNCORRECTED_ERR3**: When written with value 1, this self clearing bit generates a pulse of 1 clock cycle. This pulse is dual_rail_converted by the IP {bit,~bit} and connected to the Error Collator USER_DEFINED_UNCORRECTED_ERR3.
2	0x0	**UNCORRECTED_ERR2**: When written with value 1, this self clearing bit generates a pulse of 1 clock cycle. This pulse is dual_rail_converted by the IP {bit,~bit} and connected to the Error Collator USER_DEFINED_UNCORRECTED_ERR2.
1	0x0	**UNCORRECTED_ERR1**: When written with value 1, this self clearing bit generates a pulse of 1 clock cycle. This pulse is dual_rail_converted by the IP {bit,~bit} and connected to the Error Collator USER_DEFINED_UNCORRECTED_ERR1.
0	0x0	**UNCORRECTED_ERR0**: When written with value 1, this self clearing bit generates a pulse of 1 clock cycle. This pulse is dual_rail_converted by the IP {bit,~bit} and connected to the Error Collator USER_DEFINED_UNCORRECTED_ERR0.

VI_CFG_USER_DEFINED_CORRECTED_ERR_TRIGGER_0

Offset: 0x1037
Byte Offset: 0x40dc
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**CORRECTED_ERR5**: When written with value 1, this self clearing bit generates a pulse of 1 clock cycle. This pulse is dual_rail_converted by the IP {bit,~bit} and connected to the Error Collator USER_DEFINED_CORRECTED_ERR5.

Bit	Reset	Description
4	0x0	**CORRECTED_ERR4**: When written with value 1, this self clearing bit generates a pulse of 1 clock cycle. This pulse is dual_rail_converted by the IP {bit,~bit} and connected to the Error Collator USER_DEFINED_CORRECTED_ERR4.
3	0x0	**CORRECTED_ERR3**: When written with value 1, this self clearing bit generates a pulse of 1 clock cycle. This pulse is dual_rail_converted by the IP {bit,~bit} and connected to the Error Collator USER_DEFINED_CORRECTED_ERR3.
2	0x0	**CORRECTED_ERR2**: When written with value 1, this self clearing bit generates a pulse of 1 clock cycle. This pulse is dual_rail_converted by the IP {bit,~bit} and connected to the Error Collator USER_DEFINED_CORRECTED_ERR2.
1	0x0	**CORRECTED_ERR1**: When written with value 1, this self clearing bit generates a pulse of 1 clock cycle. This pulse is dual_rail_converted by the IP {bit,~bit} and connected to the Error Collator USER_DEFINED_CORRECTED_ERR1.
0	0x0	**CORRECTED_ERR0**: When written with value 1, this self clearing bit generates a pulse of 1 clock cycle. This pulse is dual_rail_converted by the IP {bit,~bit} and connected to the Error Collator USER_DEFINED_CORRECTED_ERR0.

VI_CFG_VI_CHANSEL_PARITY_ERR_STATUS1_0

Error Logging

Offset: 0x1038
Byte Offset: 0x40e0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**CH31_ERR**: Hot code channel ID to indicate if VI chansel 31 ecounters a parity error.
30	0x0	**CH30_ERR**: Hot code channel ID to indicate if VI chansel 30 ecounters a parity error.
29	0x0	**CH29_ERR**: Hot code channel ID to indicate if VI chansel 29 ecounters a parity error.

Bit	Reset	Description
28	0x0	**CH28_ERR**: Hot code channel ID to indicate if VI chansel 28 ecounters a parity error.
27	0x0	**CH27_ERR**: Hot code channel ID to indicate if VI chansel 27 ecounters a parity error.
26	0x0	**CH26_ERR**: Hot code channel ID to indicate if VI chansel 26 ecounters a parity error.
25	0x0	**CH25_ERR**: Hot code channel ID to indicate if VI chansel 25 ecounters a parity error.
24	0x0	**CH24_ERR**: Hot code channel ID to indicate if VI chansel 24 ecounters a parity error.
23	0x0	**CH23_ERR**: Hot code channel ID to indicate if VI chansel 23 ecounters a parity error.
22	0x0	**CH22_ERR**: Hot code channel ID to indicate if VI chansel 22 ecounters a parity error.
21	0x0	**CH21_ERR**: Hot code channel ID to indicate if VI chansel 21 ecounters a parity error.
20	0x0	**CH20_ERR**: Hot code channel ID to indicate if VI chansel 20 ecounters a parity error.
19	0x0	**CH19_ERR**: Hot code channel ID to indicate if VI chansel 19 ecounters a parity error.
18	0x0	**CH18_ERR**: Hot code channel ID to indicate if VI chansel 18 ecounters a parity error.
17	0x0	**CH17_ERR**: Hot code channel ID to indicate if VI chansel 17 ecounters a parity error.
16	0x0	**CH16_ERR**: Hot code channel ID to indicate if VI chansel 16 ecounters a parity error.
15	0x0	**CH15_ERR**: Hot code channel ID to indicate if VI chansel 15 ecounters a parity error.
14	0x0	**CH14_ERR**: Hot code channel ID to indicate if VI chansel 14 ecounters a parity error.
13	0x0	**CH13_ERR**: Hot code channel ID to indicate if VI chansel 13 ecounters a parity error.
12	0x0	**CH12_ERR**: Hot code channel ID to indicate if VI chansel 12 ecounters a parity error.
11	0x0	**CH11_ERR**: Hot code channel ID to indicate if VI chansel 11 ecounters a parity error.
10	0x0	**CH10_ERR**: Hot code channel ID to indicate if VI chansel 10 ecounters a parity error.
9	0x0	**CH9_ERR**: Hot code channel ID to indicate if VI chansel 9 ecounters a parity error.

Bit	Reset	Description
8	0x0	**CH8_ERR**: Hot code channel ID to indicate if VI chansel 8 ecounters a parity error.
7	0x0	**CH7_ERR**: Hot code channel ID to indicate if VI chansel 7 ecounters a parity error.
6	0x0	**CH6_ERR**: Hot code channel ID to indicate if VI chansel 6 ecounters a parity error.
5	0x0	**CH5_ERR**: Hot code channel ID to indicate if VI chansel 5 ecounters a parity error.
4	0x0	**CH4_ERR**: Hot code channel ID to indicate if VI chansel 4 ecounters a parity error.
3	0x0	**CH3_ERR**: Hot code channel ID to indicate if VI chansel 3 ecounters a parity error.
2	0x0	**CH2_ERR**: Hot code channel ID to indicate if VI chansel 2 ecounters a parity error.
1	0x0	**CH1_ERR**: Hot code channel ID to indicate if VI chansel 1 ecounters a parity error.
0	0x0	**CH0_ERR**: Hot code channel ID to indicate if VI chansel 0 ecounters a parity error.

VI_CFG_VI_CHANSEL_PARITY_ERR_STATUS2_0

Offset: 0x1039
Byte Offset: 0x40e4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	0x0	**CH35_ERR**: Hot code channel ID to indicate if VI chansel 35 ecounters a parity error.
2	0x0	**CH34_ERR**: Hot code channel ID to indicate if VI chansel 34 ecounters a parity error.
1	0x0	**CH33_ERR**: Hot code channel ID to indicate if VI chansel 33 ecounters a parity error.
0	0x0	**CH32_ERR**: Hot code channel ID to indicate if VI chansel 32 ecounters a parity error.

VI_CFG_PERFMON_SW_1_0

Offset: 0x103a
Byte Offset: 0x40e8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PERF_OUT_1**: FW read/write register that feeds as input to Perfmux. bit_0 for channel_0, bit_31 for channel_31. which can be routed to HWPM for calculating throughput, fps, overheads.

VI_CFG_PERFMON_SW_2_0

Offset: 0x103b
Byte Offset: 0x40ec
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	0x0	**PERF_OUT_2**: FW read/write register that feeds as input to Perfmux. bit_0 for channel_32, bit3 for channel_35. which can be routed to HWPM for calculating throughput, fps, overheads.

VI_CHANSEL_STATUS_NOMATCH_0

Register to indicate CHANSEL status for debug purpose.

Disable PROD chk as SW driver will set the right values.

Offset: 0x1060
Byte Offset: 0x4180
Read/Write: See table below

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31:16	RO	X	**FRAME_ID**: the Frame ID which triggers NOMATCH event; only saves the 1st frame ID if NOMATCH happened multi-times; Cleared by write 1 to 'NOMATCH' field
13:8	RO	X	**DT**: the CSI data type which triggers NOMATCH event; only saves the 1st data type if NOMATCH happened multi-times; Cleared by write 1 to 'NOMATCH' field
7:4	RO	X	**VC_ID**: the Virtual_Channnel_ID (binary) which triggers NOMATCH event; only saves the 1st Virtual_Channnel_ID if NOMATCH happened multi-times; Cleared by write 1 to 'NOMATCH' field
3:1	RO	X	**CSI_STREAM_ID**: the CSI_STREAM_ID (binary) which triggers NOMATCH event; only saves the 1st CSI_STREAM_ID if NOMATCH happened multi-times; Cleared by write 1 to 'NOMATCH' field
0	RW	X	**NOMATCH**: whether CHANSEL_NOMATCH happened; write 1 to clear

VI_PIXFMT_COMPAND_KNEE_CFG0_0

This is an array of 10 identical register entries; the register fields below apply to each entry.
Full register list is: VI_PIXFMT_COMPAND_KNEE_CFG0_<i>, among which <i> belongs to <0..9>.

Offset: 0x1080,..,0x1089
Byte Offset: 0x4200,..,0x4224
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,x000,0000,0000,0000,0000,0xxx,xxxx)

Bit	Reset	Description
26:7	0x0	**BASE**: input position for this knee point in U20 format

VI_PIXFMT_COMPAND_KNEE_CFG1_0

This is an array of 10 identical register entries; the register fields below apply to each entry.
Full register list is: VI_PIXFMT_COMPAND_KNEE_CFG0_<i>, among which <i> belongs to <0..9>.

Offset: 0x108a,..,0x1093
Byte Offset: 0x4228,..,0x424c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000080 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,1000,0000)

Bit	Reset	Description
16:0	0x80	**SCALE**: Scale above this knee point in U10.7 format

VI_PIXFMT_COMPAND_KNEE_CFG2_0

This is an array of 10 identical register entries; the register fields below apply to each entry.
Full register list is: VI_PIXFMT_COMPAND_KNEE_CFG0_<i>, among which <i> belongs to <0..9>.

Offset: 0x1094,..,0x109d
Byte Offset: 0x4250,..,0x4274
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,x000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
26:0	0x0	**OFFSET**: Output offset for this knee point in U20.7 format

VI_PIXFMT_ECC_STATUS0_0

Offset: 0x10a0
Byte Offset: 0x4280
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
30	RO	0x0	**IS_EMB_DATA**: 1 to indicate current packet is an embed data
29:25	RO	0x0	**PTYPE**: Value of Packet Type
24	RO	0x0	**VPR**: Value of VPR
23:8	RO	0x0	**FRAME_ID**: Value of FRAME_ID
7:2	RO	0x0	**NVCSI_STREAM_ID**: Value of NVCSI_STREAM_ID_ONE_HOT
1	RW	0x0	**DED_ERROR**: 1 to indicate a double bits error triggered
0	RW	0x0	**SEC_ERROR**: 1 to indicate a single bit error triggered

VI_PIXFMT_ECC_STATUS1_0

Offset: 0x10a1
Byte Offset: 0x4284
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VI_CHANNEL_ID_LO**: One hot channel id for lowest 32 channel. bit31: channel_31, bit0: channel_0.

VI_PIXFMT_ECC_STATUS2_0

Offset: 0x10a2
Byte Offset: 0x4288
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,x000,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
18	0x0	**FRAME_DONE**: Value of FRAME_DONE
17	0x0	**NLINES_DONE**: Value of NLINES_DONE
16	0x0	**EMB_DONE**: Value of EMB_DONE
3:0	0x0	**VI_CHANNEL_ID_HI**: One hot channel id for highest 4 channel. bit3: channel_35, bit0: channel_32.

VI_PIXFMT_PDAF_PATTERN_0

This is an array of 32 identical register entries; the register fields below apply to each entry.
Full register list is: VI_PIXFMT_PDAF_PATTERN_<i>, among which <i> belongs to <0..31>.

Offset: 0x10c0,..,0x10df
Byte Offset: 0x4300,..,0x437c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ENABLE**: Pixel bitmap, by line should be output to the PDAF surface

VI_PIXFMT_PDAF_PATTERN_REPLACE_0

This is an array of 32 identical register entries; the register fields below apply to each entry.
Full register list is: VI_PIXFMT_PDAF_PATTERN_REPLACE_<i>, among which <i> belongs to <0..31>.

Offset: 0x10e0,..,0x10ff
Byte Offset: 0x4380,..,0x43fc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**REPLACE_ENABLE**: Pixel bitmap to be used for Replace the pdaf pixel, by line should be output to the PDAF surface

VI_CSIMUX_CSIMUX_DEINT_0

All CSI MUX registers are non-shadowed, shadowing only starts at chansel.

Offset: 0x1100
Byte Offset: 0x4400
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**

VI_CSIMUX_CSIMUX_STREAM_0

This is an array of six identical register entries; the register fields below apply to each entry.
Full register list is: VI_CSIMUX_CSIMUX_STREAM_<i>, among which <i> belongs to <0..5>.

Offset: 0x1101,..,0x1106
Byte Offset: 0x4404,..,0x4418
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**

VI_CSIMUX_EIGHT_LANE_DEINT_CONFIG_0

EIGHT LANE DEINT CONFIG REGISTERS

To turn on OMNIVISION style deinterleaving, set MODE field to a non-zero value. Deinterleaving is done by muxing pixels 2 from a pair of streams: either streams 0 and 2, or streams 2 and 4.

When 0 and 2 are used stream 1 is not used. Similarly stream 3 is not used when 2 and 4 are in use. One is the master stream whereas the other is the slave stream. The least numbered stream in the pair is the master.

When the (0,2) stream pair is in use, stream 0 is master, stream 2 is slave. stream 1 is forced disabled. When the (2,4) stream pair is in use, stream 2 is master, stream 4 is slave. stream 3 is forced disabled.

The outgoing stream number associated with a deinterleaved output is the stream ID of the master channel. Deinterleaving is done by joining a HI (slave) and a LO stream (master), where the LO stream has the first very first pixel of each line. For the various modes the deinterleaving is done as follows, over 2 or 3 clock cycles which are numbered as beats:

	HI	LO	
Mode 2 x8 \ FIFO PXL POSITION	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	Deinterleaved Output
Cycle \			
Beat 0	7 5 3 1	6 4 2 0	7 6 5 4 3 2 1 0
Beat 1	15 13 11 9	14 12 10 8	15 14 13 12 11 10 9 8

	HI	LO	
Mode 3 x8 \ FIFO PXL POSITION	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	Deinterleaved Output
Cycle \			
Beat 0	7 6 5 4	3 2 1 0	7 6 5 4 3 2 1 0
Beat 1	15 14 13 12	11 10 9 8	15 14 13 12 11 10 9 8

Mode 3 x6 \ FIFO PXL POSITION	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	Deinterleaved Output
Cycle \			
Beat 0	5 4	^7 ^6 ^3 ^2 ^1 ^0	7 6 5 4 3 2 1 0
Beat 1	11 10	^9 ^8	15 14 13 12 | 15 14 13 12 11 10 9 8
Beat 2	23 22 17 16	21 20 19 18	23 22 21 20 19 18 17 16

Mode 3 x6: for each entry of the HI FIFO, there are two words from the LO FIFO. Pixel with a ^ prefix are on the first entry

from the LO FIFO whereas others are from the second entry. Since Beat 1 combines pixels from the first and second

FIFO word it follows that ^8 and ^9 and need to be saved when popping the low entry.

Constraint A: Omnivision interleaved stream can use only one virtual channel.

Constraint B: Omnivision interleaved stream can use only RAW modes.

Constraint C: Omnivision interleaved stream must at least be 9 pixels wide for Mode 3 X8, 7 pixels wide for Mode 3 X6, and 3 pixels wide for Mode 2 x8

Constraint D: Omnivision interleaved stream must use 8 PPC, 4 PPC is not supported.

If MODE==0 then WT is forced to zero and deinterleaved settings are used from the STREAM registers.

When WT set to 0 implies the stream is turned off and no traffic will be allowed out if the DEINT FIFOs.

If traffic is pushed in with DEINT in use and WT=0 this will eventually cause a FIFO overflow.

To block traffic from interleaved queues, set Q_BLOCK=1, this also resets the FIFO.

To reset deinterleaver in an errored state:

Set QBLOCK=1,

then trigger FEINJECT=1 (this terminate an ongoing frame, and marks it with error),

then trigger SRESET=1, (optional reset on state such as overflow detection and spurious data)

then release QBLOCK=0 to resume and unblock traffic.

To avoid hanging frames (a frame that has begun for which no Frame End is occurring within a reasonable amount of time set FE_MAX_TIME to a non-zero value. At each start of frame a timer is started. If the timer elapses before a frame end is seen then an errored frame end is forced down the pipeline.

For HW testing , FRAMEMAXTIME can also be used to inject error in the pipeline by setting the count expected to be smaller than the transmission duration of the expected frames.

Setting 0 on the FRAMEIDGEN bit resets the generated frame ID counter to 1 for the next time FRAMEIDGEN is set.

Offset: 0x1107
Byte Offset: 0x441c
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x0X400000 (0b0000,x00x,0100,0000,0000,0000,0000,0000)

Bit	R/W	Parity Protection	Reset	Description
31:30	RW	N	0x0	**MUXSEL**: Which CSI BRICKS Master Slave configuration is in use when 8 lane in use **0** = PAIR_MS_02 **1** = PAIR_MS_24
29:28	RW	Y	0x0	**MODE**: Whether deinterleaving mode is in use Mode3_x6 **0** = DISABLE **1** = MODE2 **2** = MODE3 **3** = MODE3_X6

Bit	R/W	Parity Protection	Reset	Description
26	RW	N	0x0	**FRAMEIDGEN**: Insert a sequential frame ID on each incoming frame, discard CSI frame ID field.
25	RW	N	0x0	**STICKYFAULT**: Some form of fault on the line has occurred if bit reads high
24	RO	N	X	**VPR**: VPR mode for DEINTERLAVED stream, there is at most one virtual channel
23	RW	N	0x0	**SRESET**: Reset some state associated with EIGHT_LANE deinterleave
22	RW	N	0x1	**QBLOCK**: Block traffic and reset both FIFO queues associated with EIGHT_LANE deinterleaving when MODE!=DISABLE
21	RW	N	0x0	**FEINJECT**: Manually Force a Frame End on stream stream is in a frame (always issue during a Q_RESET) when MODE!=DISABLE
20	RW	N	0x0	**FESHORTTIMER**: If 0 FE_MAX_TIME is in steps 2^20 VI cycles. If 1 then steps of 2^8 cycles (sims or error inject)
19:4	RW	N	0x0	**FEMAXTIME**: Force a frame end after a frame start when FE_MAX_TIME*2^20 clock cycles have elapsed, off if ==0
3:0	RW	N	0x0	**WT**: The weight of deint stream into arbiter

VI_CSIMUX_EIGHT_LANE_DEINT_NOTIFY_MASK_0

Offset: 0x1108
Byte Offset: 0x4420
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:20	0x0	**STREAM_EVENT_HI_MASK**: deint_stream_even_mask

Bit	Reset	Description
19:16	0x0	**STREAM_EVENT_LO_MASK**: deint_stream_even_mask
15:2	0x0	**FRAME_FAULT_MASK**: deint_frame_fault_mask
1:0	0x0	**FRAME_STATUS_MASK**: mask particular status events from being notified

VI_CSIMUX_CONFIG_STREAM_0

STREAM CONFIGURATION

Streams which are used by the Omnivision style interleaving have their configuration automatically disabled. For example, if pair (0,2) is interleaved then WT revers to 0 for streams 1,2 and WT for from the deinterleaved configuration register applies to stream 0.

When WT set to 0 implies the stream is turned off and no traffic will be allowed out if the correspnding stream FIFOs. If traffic is pushed in on a stream with no DEINT in use and WT=0 this will eventually cause a STREAM FIFO overflow.

Setting 0 on the FRAMEIDGEN bit resets the generated frame ID counter to 0 for the next time FRAMEIDGEN is set. There is one FRAMEIDGEN bit per VC.

This is an array of six identical register entries; the register fields below apply to each entry. Full register list is: VI_CSIMUX_CONFIG_STREAM_<i>, among which <i> belongs to <0..5>.

Offset: 0x1109,..,0x110e
Byte Offset: 0x4424,..,0x4438
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x0X400000 (0bxx00,000x,0100,0000,0000,0000,0000,0000)

Bit	R/W	Parity Protection	Reset	Description
29:26	RW	Y	0x0	**FRAMEIDGEN**: Insert a sequential frame ID on each incoming frame, discard CSI frame ID field.
25	RW	N	0x0	**STICKYFAULT**: Some form of fault on the stream if bit reads high
24	RO	N	X	**VPR**: PMC VPR mode for associated CSI brick

Bit	R/W	Parity Protection	Reset	Description
23	RW	N	0x0	**SRESET**: Reset some state associated with STREAM
22	RW	Y	0x1	**QBLOCK**: Block traffic and reset both FIFO queue associated with stream
21	RW	N	0x0	**FEINJECT**: Manually Force a Frame End on VC associated with stream, see CONFIG_FEINJECT_VC register
20	RW	N	0x0	**FESHORTTIMER**: If 0 FE_MAX_TIME is steps 2^20 VI cycles. If 1 then steps of 2^8 cycles (sims or error inject)
19:4	RW	Y	0x0	**FEMAXTIME**: Insert a frame end after a frame start when FE_MAX_TIME*2^20 clock cycles have elapsed, off if 0
3:0	RW	Y	0x0	**WT**: The weight of stream into arbiter

VI_CSIMUX_CONFIG_STREAM_VC_HI_0

This is an array of six identical register entries; the register fields below apply to each entry.
Full register list is: VI_CSIMUX_CONFIG_STREAM_VC_HI_<i>, among which <i> belongs to <0..5>.

Offset: 0x110f,..,0x1114
Byte Offset: 0x443c,..,0x4450
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00fff000 (0bxxxx,xxxx,1111,1111,1111,0000,0000,0000)

Bit	Reset	Description
23:12	0xfff	**FEINJECT_VC_HI**: which of the hi 12 VCs to inject FE on, bit 0 to 12 pos is VC number 4 to 16
11:0	0x0	**FRAMEIDGEN_HI**: Insert a sequential frame ID on each incoming frame for HI 12 VCs, discard CSI frame ID field.

VI_CSIMUX_CONFIG_FEINJECT_VC_0

When using FEINJECT field of CONFIG_STREAM register, select which subset of VCs to inject an FE on. Injection only occurs if the VC is in frame. The default 0xF is to inject on all four VCs of a stream. Such a field does not exist for deinterleaved case since deinterleaved stream can only have one VC. The purpose of this register is to abort as few frames as possible should a frame require a particular VC need to terminate.

Offset: 0x1115
Byte Offset: 0x4454
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00ffffff (0bxxxx,xxxx,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
23:20	0xf	**STREAM5_FEINJECT_VC**: stream 5: which of 4 VCs to inject FE on, bit 0 to 3 pos is VC number
19:16	0xf	**STREAM4_FEINJECT_VC**: stream 4: which of 4 VCs to inject FE on, bit 0 to 3 pos is VC number
15:12	0xf	**STREAM3_FEINJECT_VC**: stream 3: which of 4 VCs to inject FE on, bit 0 to 3 pos is VC number
11:8	0xf	**STREAM2_FEINJECT_VC**: stream 2: which of 4 VCs to inject FE on, bit 0 to 3 pos is VC number
7:4	0xf	**STREAM1_FEINJECT_VC**: stream 1: which of 4 VCs to inject FE on, bit 0 to 3 pos is VC number
3:0	0xf	**STREAM0_FEINJECT_VC**: stream 0: which of 4 VCs to inject FE on, bit 0 to 3 pos is VC number

VI_CSIMUX_NOTIFY_MASK_STREAM_0

STREAM NOTIFICATION MASK (6 registers)

This is an array of six identical register entries; the register fields below apply to each entry.
Full register list is: VI_CSIMUX_NOTIFY_MASK_STREAM_<i>, among which <i> belongs to <0..5>.

Offset: 0x1116,..,0x111b
Byte Offset: 0x4458,..,0x446c
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:16	0x0	**STREAM_EVENT_MASK**: deint_stream_event_mask
15:2	0x0	**FRAME_FAULT_MASK**: deint_frame_fault_mask
1:0	0x0	**FRAME_STATUS_MASK**: mask particular status events from being notified

VI_CSIMUX_CSI_STREAM_0

STREAM CSI FAULTS (6 registers)

Mask specific bits of the CSI error status so that only some type of errors result in failure. CSIMASK registers are applicable even when deinterleave mode is in use.

This is an array of six identical register entries; the register fields below apply to each entry. Full register list is: VI_CSIMUX_CSI_STREAM_<i>, among which <i> belongs to <0..5>.

Offset: 0x111c,..,0x1121
Byte Offset: 0x4470,..,0x4484
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**FAULT_MASK** Bit 0: PP FSM timeout Bit 1: PH Single bit error repaired Bit 2: CRC error on pixel line payload Bit 3: Line short error Bit 4: CPHY PH Single CRC error Bit 5: CRC error on emb line payload

VI_CSIMUX_STAT_CFG_0

STREAM FIFO STATS (6 registers + 1 CFG register)

The config register is used to power up or down the FIFO statistics collections.

Max read count resets when a write is issued to the register

Note that you should reset the value when the stream is quiescent.

Otherwise you might not capture the maximum between the last read prior to a write.

STATS registers are applicable even when deinterleave mode is in use.

Note that two sets of registers are provided to estimate the depth of the FIFO: RD and WR count.

Use the maximum of both to determine the deepest level the FIFO has reached.

This should normally come from the write side of the FIFO.

Note that after resetting the FIFO you should give time for the write side to propagate cleanly.

(gray codes are used and a reset event might violate a Gray code sequence)

The write counts are expected to be >= to FIFO levels reached, the read counts <= to them. If the FIFO DEPTH reads as 0x3F instead of 0x20 it is because you did not enable FSTAT.

Offset: 0x1122
Byte Offset: 0x4488
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE_FSTAT**: ENABLE FIFO STATistics computation **0** = DISABLE **1** = ENABLE

VI_CSIMUX_STAT_STREAM_0

This is an array of six identical register entries; the register fields below apply to each entry.
Full register list is: VI_CSIMUX_STAT_STREAM_<i>, among which <i> belongs to <0..5>.

Offset: 0x1123,..,0x1128
Byte Offset: 0x448c,..,0x44a0
Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
29:24	RO	X	**FIFO_MAX_WR_COUNT**
23:18	RO	X	**FIFO_WR_COUNT**
17:12	RW	X	**FIFO_MAX_RD_COUNT**
11:6	RO	X	**FIFO_RD_COUNT**
5:0	RO	X	**FIFO_DEPTH**

VI_CSIMUX_STAT_FRAME_0

When FRAMEIDGEN is in use FRAMEID can be used as a frame counter on a virtual channel. First frame is frame 1.

The registers are numbered as follows 4*STREAM_ID + VC_ID where VC_ID is numbered from 0-3. So the range is from 0 to 23. Deinterleave mode has its own specific register.

There is also an in frame status of all registers.

When using generated frame IDs:

If FRAMEID==0 and INFRAME==0 then no frame has been seen.

If FRAMEID==1 and INFRAME==1 then processing first frame.

If FRAMEID==1 and INFRAME==0 then first frame has been processed, waiting for second to start.

This is an array of 96 identical register entries; the register fields below apply to each entry.
Full register list is: VI_CSIMUX_STAT_FRAME_<i>, among which <i> belongs to <0..95>.

Offset: 0x1129,..,0x1188
Byte Offset: 0x44a4,..,0x4620
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description

16	X	**INFRAME:** Determine whether a virtual channel is actively processing a frame
15:0	X	**FRAMEID:** Determine the current or last frame ID seen on a viritual channel

VI_CSIMUX_STAT_FRAME_DEINT_0

Offset: 0x1189
Byte Offset: 0x4624
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
16	X	**INFRAME**
15:0	X	**FRAMEID**

VI_CSIMUX_FRAME_STATUS_0

This is an array of six identical register entries; the register fields below apply to each entry.
Full register list is: VI_CSIMUX_FRAME_STATUS_<i>, among which <i> belongs to <0..5>.

Offset: 0x118a,..,0x118f
Byte Offset: 0x4628,..,0x463c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000XXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
15:0	X	**VC_INFRAME**: VC in frame status, each bit for 1 VC

VI_CSIMUX_FRAME_STATUS_DEINT_0

Offset: 0x1190
Byte Offset: 0x4640
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
0	X	**DEINT_INFRAME**: Deinterleave is in frame

VI_CSIMUX_INJECT_DATA_0_0

Offset: 0x1191
Byte Offset: 0x4644
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PPC4_PXL0**: Or PPC8_RAWPXL0 and PPC8_RAWPXL1, or FRAME_ID

VI_CSIMUX_INJECT_DATA_1_0

Offset: 0x1192
Byte Offset: 0x4648
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PPC4_PXL1**: Or PPC8_RAWPXL2 and PPC8_RAWPXL3

VI_CSIMUX_INJECT_DATA_2_0

Offset: 0x1193
Byte Offset: 0x464c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PPC4_PXL2**: Or PPC8_RAWPXL4 and PPC8_RAWPXL5

VI_CSIMUX_INJECT_DATA_3_0

Offset: 0x1194
Byte Offset: 0x4650
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PPC4_PXL3**: Or PPC8_RAWPXL6 and PPC8_RAWPXL7

VI_CSIMUX_INJECT_HEADER_0

Offset: 0x1195
Byte Offset: 0x4654
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,x000,0000,0000,0000,0000,0000)

Bit	Reset	Description
22	0x0	**PPC_FLAG**: 8ppc=1, 4ppc=0
21:18	0x0	**CTYPE**: **1** = SP_FE **8** = SP_FS **6** = LP_LS_LE **2** = LP_LE **9** = LP_DATA **4** = LP_LS
17:14	0x0	**VC**
13:8	0x0	**DTYPE**: Or FE_ERROR
7:0	0x0	**PXL_ENABLE**

VI_CSIMUX_INJECT_CFG_0

Offset: 0x1196
Byte Offset: 0x4658
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0xXXXXX000 (0bxxxx,xxxx,xxxx,xxxx,xx00,xxxx,xx00,0000)

Bit	R/W	Parity Protection	Reset	Description
31:26	RO	N	X	**FIFO_DEPTH:** Number of programmable packets that can be stored before release into CSIMUX (32)
25:20	RO	N	X	**FIFO_WR_COUNT:** Number programmed packet entries waiting to be released as seen from READ side of FIFO.
19:14	RO	N	X	**FIFO_RD_COUNT:** Number programmed packet entries waiting to be released as seen from WRITE side of FIFO.
13	RW	Y	0x0	**PIPE_STALL:** Stall the CSIMUX pipeline while high, an effective way to test overflow using host interface.
12	RW	N	0x0	**ACCUMULATE:** When high: accumulate packets written by HOST. When lowered: release them as a burst

Bit	R/W	Parity Protection	Reset	Description
5:0	RW	Y	0x0	**STREAM_SWITCH:** Hot encoded packet source for each stream: CSI(0) or HOST(1), acts as a simple multiplexer select for packet source of each stream. Not all host assembled packets go to each stream listed in high in STREAM_SWITCH. The separate STREAM field of the INJECT register is used to direct packets to a particular stream.

VI_CSIMUX_INJECT_0

The following actually triggers an inject event; when INJECT is written

to, the effect is as if the CSI2VI_PKT in question appeared from the

NVCSI stream specified in STREAM.

The hot encoding determining which host streams should get packet of interest.

As such it is possible to send a same packet to all 6 streams at once.

The STREAM field should be a subset of STREAM_SWITCH since any stream switched to CSI input

will ignore input from the host. STREAM is ANDed with STREAM_SWITCH.

Note that any writes to this register are stalled should the host packet FIFO be full.

This has for effect stalling the host interface, this condition is not expected to occur under normal operation.

Any host stall automatically disables PIPE_STALL and ACCUMULATE until it is determined host packet FIFO is

no longer full.

Offset: 0x1197
Byte Offset: 0x465c
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
5:0	X	**STREAM**

VI_CSIMUX_STREAM_SWIZZLER_0

STREAM SWIZZLE

Allows mapping of any stream onto another stream. Allows to test multiple streams with a single sensor by copying data over.

This feature can only be used if all PMC VPR bits are set to 0.

Offset: 0x1198
Byte Offset: 0x4660
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00543210 (0bxxxx,xxxx,x101,x100,x011,x010,x001,x000)

Bit	Reset	Description
22:20	0x5	**STREAM5_SWIZZLE_SRC**: source CSI stream for VI stream 5
18:16	0x4	**STREAM4_SWIZZLE_SRC**: source CSI stream for VI stream 4
14:12	0x3	**STREAM3_SWIZZLE_SRC**: source CSI stream for VI stream 3
10:8	0x2	**STREAM2_SWIZZLE_SRC**: source CSI stream for VI stream 2
6:4	0x1	**STREAM1_SWIZZLE_SRC**: source CSI stream for VI stream 1
2:0	0x0	**STREAM0_SWIZZLE_SRC**: source CSI stream for VI stream 0

VI_CSIMUX_INPUT_STREAM_MUX_0

Offset: 0x1199
Byte Offset: 0x4664

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,x0x0,x0x0,x0x0)

Bit	Reset	Description
10	0x0	**STREAM5**: **0** = NVCSI_STREAM **1** = TERMINATE
8	0x0	**STREAM4**: **0** = NVCSI_STREAM **1** = TERMINATE
6	0x0	**STREAM3**: **0** = NVCSI_STREAM **1** = TERMINATE
4	0x0	**STREAM2**: **0** = NVCSI_STREAM **1** = TERMINATE
2	0x0	**STREAM1**: **0** = NVCSI_STREAM **1** = TERMINATE
0	0x0	**STREAM0**: **0** = NVCSI_STREAM **1** = TERMINATE

VI_ISPBUFA_ERROR_0

Offset: 0x1400
Byte Offset: 0x5000
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**FIFO_OVERFLOW**: Set by Hardware when the ISPBUF's internal FIFO has overflowed. (Generally due to clock speed mismatch b/w ISP and VI interfaces) Write 1 to clear. Also causes VI Master error.

VI_ISPBUFA_SW_RESET_0

Offset: 0x1401
Byte Offset: 0x5004
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ISPINTF_RESET**: Reset ISP interface This S/W reset is to be used when Asynchronously reset ISP unit due to an error condition (possibly FIFO_OVERFLOW) 1 : Assert reset on ISP interface 0 : De-asssert reset on ISP interface This reset needs to be held for the duration of atleast 200 isp_clk time in order to ensure that the internal buffer is completely flushed out.

VI_ATOMP_DVFS_LIMIT_WR_LIMIT_0

Offset: 0x1d00
Byte Offset: 0x7400
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x000001c2 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx1,1100,0010)

Bit	Reset	Description
8:0	0x1c2	**LIMIT**

VI_OFIF_FIFO_SIZE_0

Offset: 0x1e00
Byte Offset: 0x7800
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000XXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
10:0	X	**SIZE**: Total number of entries in OFIFs latency buffer. Read-only.

VI_OFIF_FIFO_OCCUPANCY_0

Offset: 0x1e01
Byte Offset: 0x7804
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXX0XXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
26:16	RW	X	**MAX**: All-time maximum number of entries occupied in OFIF FIFO buffer. Write any value to clear to CURRENT.
10:0	RO	X	**CURRENT**: Current number of entries occupied in OFIF FIFO buffer. Read-only.

VI_OFIF_FIFO_DROP_STATUS_0

Offset: 0x1e02
Byte Offset: 0x7808
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00XX00XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
22:17	RO	X	**TOSSED_CHANNEL_ID**: the channelID of ATOMP_TOSSED; only saves the 1st channelID if ATOMP_TOSSED happened multi-times
16	RW	X	**TOSSED**: whether ATOMP_TOSSED happened; write 1 to clear
6:1	RO	X	**TRUNCATED_CHANNEL_ID**: the channelID of ATOMP_TRUNCATED; only saves the 1st channelID if ATOMP_TRUNCATED happened multi-times
0	RW	X	**TRUNCATED**: whether ATOMP_TRUNCATED happened; write 1 to clear

VI_OFIF_DROPPED_ATOMS_BE0_PLANE0_0

Offset: 0x1e03
Byte Offset: 0x780c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**NUMBER**: Number of atoms dropped at ATOMP back-end. Each plane of ATOMP back-end has a counter. Write any value to clean to 0.

VI_OFIF_DROPPED_ATOMS_BE0_PLANE1_0

Offset: 0x1e04
Byte Offset: 0x7810
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**NUMBER**: Number of atoms dropped at ATOMP back-end. Each plane of ATOMP back-end has a counter. Write any value to clean to 0.

VI_OFIF_DROPPED_ATOMS_BE0_PLANE2_0

Offset: 0x1e05
Byte Offset: 0x7814
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**NUMBER**: Number of atoms dropped at ATOMP back-end. Each plane of ATOMP back-end has a counter. Write any value to clean to 0.

VI_OFIF_DROPPED_ATOMS_BE1_PLANE0_0

Offset: 0x1e06
Byte Offset: 0x7818
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**NUMBER**: Number of atoms dropped at ATOMP back-end. Each plane of ATOMP back-end has a counter. Write any value to clean to 0.

VI_OFIF_DROPPED_ATOMS_BE1_PLANE1_0

Offset: 0x1e07
Byte Offset: 0x781c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**NUMBER**: Number of atoms dropped at ATOMP back-end. Each plane of ATOMP back-end has a counter. Write any value to clean to 0.

VI_OFIF_DROPPED_ATOMS_BE1_PLANE2_0

Offset: 0x1e08
Byte Offset: 0x7820
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**NUMBER**: Number of atoms dropped at ATOMP back-end. Each plane of ATOMP back-end has a counter. Write any value to clean to 0.

VI_OFIF_DROPPED_ATOMS_BE2_PLANE0_0

Offset: 0x1e09
Byte Offset: 0x7824
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**NUMBER**: Number of atoms dropped at ATOMP back-end. Each plane of ATOMP back-end has a counter. Write any value to clean to 0.

VI_OFIF_DROPPED_ATOMS_BE2_PLANE1_0

Offset: 0x1e0a
Byte Offset: 0x7828
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**NUMBER**: Number of atoms dropped at ATOMP back-end. Each plane of ATOMP back-end has a counter. Write any value to clean to 0.

VI_OFIF_DROPPED_ATOMS_BE2_PLANE2_0

Offset: 0x1e0b
Byte Offset: 0x782c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**NUMBER**: Number of atoms dropped at ATOMP back-end. Each plane of ATOMP back-end has a counter. Write any value to clean to 0.

VI_OFIF_DROPPED_ATOMS_BE3_PLANE0_0

Offset: 0x1e0c
Byte Offset: 0x7830
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**NUMBER**: Number of atoms dropped at ATOMP back-end. Each plane of ATOMP back-end has a counter. Write any value to clean to 0.

VI_OFIF_DROPPED_ATOMS_BE3_PLANE1_0

Offset: 0x1e0d
Byte Offset: 0x7834
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**NUMBER**: Number of atoms dropped at ATOMP back-end. Each plane of ATOMP back-end has a counter. Write any value to clean to 0.

VI_OFIF_DROPPED_ATOMS_BE3_PLANE2_0

Offset: 0x1e0e
Byte Offset: 0x7838
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**NUMBER**: Number of atoms dropped at ATOMP back-end. Each plane of ATOMP back-end has a counter. Write any value to clean to 0.

VI_OFIF_DROPPED_ATOMS_BE4_PLANE0_0

Offset: 0x1e0f
Byte Offset: 0x783c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**NUMBER**: Number of atoms dropped at ATOMP back-end. Each plane of ATOMP back-end has a counter. Write any value to clean to 0.

VI_OFIF_DROPPED_ATOMS_BE4_PLANE1_0

Offset: 0x1e10
Byte Offset: 0x7840
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**NUMBER**: Number of atoms dropped at ATOMP back-end. Each plane of ATOMP back-end has a counter. Write any value to clean to 0.

VI_OFIF_DROPPED_ATOMS_BE4_PLANE2_0

Offset: 0x1e11
Byte Offset: 0x7844
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**NUMBER**: Number of atoms dropped at ATOMP back-end. Each plane of ATOMP back-end has a counter. Write any value to clean to 0.

VI_OFIF_DROPPED_ATOMS_BE5_PLANE0_0

Offset: 0x1e12
Byte Offset: 0x7848
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**NUMBER**: Number of atoms dropped at ATOMP back-end. Each plane of ATOMP back-end has a counter. Write any value to clean to 0.

VI_OFIF_DROPPED_ATOMS_BE5_PLANE1_0

Offset: 0x1e13
Byte Offset: 0x784c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**NUMBER**: Number of atoms dropped at ATOMP back-end. Each plane of ATOMP back-end has a counter. Write any value to clean to 0.

VI_OFIF_DROPPED_ATOMS_BE5_PLANE2_0

Offset: 0x1e14
Byte Offset: 0x7850
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**NUMBER**: Number of atoms dropped at ATOMP back-end. Each plane of ATOMP back-end has a counter. Write any value to clean to 0.

VI_OFIF_ECC_DEBUG_0

Offset: 0x1e15
Byte Offset: 0x7854
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:18	0x3fff	**ECC_RESERVED**: reserved for atomp ecc debug.
17:0	0x3ffff	**ECC_ERR_REPORT_EN**: one hot enable for 18 wr_ack_fifo. bit0 for wr_ack_fifo_0, bit17 for wr_ack_fifo_17.

VI_NOTIFY_FIFO_TAG_0_0

Offset: 0x2000
Byte Offset: 0x8000
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**FRAME_ID**: Frame ID from the popped event
15:8	X	**CHANNEL**: Channel from the popped event
5:1	X	**TAG**: Tag from the popped event
0	X	**VALID**: Whether the popped event was valid

VI_NOTIFY_FIFO_TIMESTAMP_0_0

Offset: 0x2001
Byte Offset: 0x8004
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**TIMESTAMP**: Timestamp from the popped event

VI_NOTIFY_FIFO_DATA_0_0

Offset: 0x2002
Byte Offset: 0x8008
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**: Payload data from the popped event

VI_NOTIFY_FIFO_EXT_DATA_0_0

Offset: 0x2003
Byte Offset: 0x800c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**: Extended payload data from the popped event

VI_NOTIFY_TAG_CLASSIFY_0_0

Offset: 0x2800
Byte Offset: 0xa000
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**NO_OUTPUT**: Tags that shall not be output to the CPU buffer FIFO (NO_OUTPUT[0] corresponds to tag 0, etc.)

VI_NOTIFY_TAG_CLASSIFY_1_0

Offset: 0x2801
Byte Offset: 0xa004
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**HIGH_PRI**: Tags that shall be considered to be high priority (HIGH[0] corresponds to tag 0, etc.)

VI_NOTIFY_FIFO_OCCUPANCY_0

Offset: 0x2802
Byte Offset: 0xa008
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
17:12	RW	X	**MAX**: Highest watermark reached for CPU buffer FIFO Write-1 to reset to 0
11:6	RO	X	**CURRENT**: Number of entries in CPU buffer FIFO
5:0	RO	X	**SIZE**: Size of CPU buffer FIFO

VI_NOTIFY_OCCUPANCY_URGENT_0

Offset: 0x2803
Byte Offset: 0xa00c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
5:0	X	**URGENT**: CPU buffer FIFO high watermark at which to trigger a high-priority interrupt, even if no high priority events are present in the FIFO

VI_NOTIFY_HIGHPRIO_0

Offset: 0x2804
Byte Offset: 0xa010
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000XXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
15:0	X	**HIGHPRIO**: Number of High priority events that the CPU has not yet processed.

VI_NOTIFY_INFIFO_OVRFLOW_STATUS_0

Offset: 0x2805
Byte Offset: 0xa014
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000XXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
8	X	**NOTIFY_BUS_MISC**: covers ATOMP_PACKER_OVERFLOW, VGP0_DONE, VGP1_DONE
7	X	**NOTIFY_BUS_ATOMP**: covers ATOMP_FRAME_DONE, ATOMP_EMB_DATA_DONE, ATOMP_FRAME_NLINES_DONE, ATOMP_PDAF_DATA_DONE
6	X	**NOTIFY_BUS_ATOMPSTATUS**: covers ATOMP_FRAME_TRUNCATED, ATOMP_FRAME_TOSSED
5	X	**NOTIFY_BUS_ATOMPDVFS**: covers ATOMP_FS, ATOMP_FE
4	X	**NOTIFY_BUS_CHANSEL_XCPTMISC**: covers CHANSEL_STALE_FRAME, CHANSEL_EMBED_SHORT
3	X	**NOTIFY_BUS_CHANSEL_XCPT**: covers CHANSEL_NOMATCH, CHANSEL_COLLISION, CHANSEL_PIX_SHORT, CHANSEL_LOAD_FRAMED
2	X	**NOTIFY_BUS_CHANSEL**: covers CHANSEL_{EMBED|PXL}_{SOF|EOF}, CHANSEL_NLINES, CHANSEL_FAULT, CHANSEL_FAULT_FE
1	X	**NOTIFY_BUS_CSIMUX_STREAM**: covers CSIMUX_STREAM
0	X	**NOTIFY_BUS_CSIMUX_FRAME**: covers FS, FE, CSIMUX_FRAME

VI_NOTIFY_CURRENT_TIMESTAMP_0_0

Offset: 0x2806
Byte Offset: 0xa018
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**TIMESTAMP**: return current timestamp value inside VI Notify

VI_NOTIFY_INFIFO_TIMESTAMP_OVRFLOW_ERR_STATUS_0

Offset: 0x2807
Byte Offset: 0xa01c

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
8	0x0	**TIMESTAMP_OVRFLOW_ERR_MISC**
7	0x0	**TIMESTAMP_OVRFLOW_ERR_ATOMP**
6	0x0	**TIMESTAMP_OVRFLOW_ERR_ATOMPSTATUS**
5	0x0	**TIMESTAMP_OVRFLOW_ERR_ATOMPDVFS**
4	0x0	**TIMESTAMP_OVRFLOW_ERR_CHANSEL_XCPTMISC**
3	0x0	**TIMESTAMP_OVRFLOW_ERR_CHANSEL_XCPT**
2	0x0	**TIMESTAMP_OVRFLOW_ERR_CHANSEL**
1	0x0	**TIMESTAMP_OVRFLOW_ERR_CSIMUX_STREAM**
0	0x0	**TIMESTAMP_OVRFLOW_ERR_CSIMUX_FRAME**:

VI_NOTIFY_INFIFO_TIMESTAMP_OVRFLOW_ERR_MASK_0

Offset: 0x2808
Byte Offset: 0xa020
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
8	0x0	**TIMESTAMP_OVRFLOW_ERR_MISC_MASK**: **0** = DISABLED **1** = ENABLED
7	0x0	**TIMESTAMP_OVRFLOW_ERR_ATOMP_MASK**: **0** = DISABLED **1** = ENABLED
6	0x0	**TIMESTAMP_OVRFLOW_ERR_ATOMPSTATUS_MASK**: **0** = DISABLED **1** = ENABLED
5	0x0	**TIMESTAMP_OVRFLOW_ERR_ATOMPDVFS_MASK**: **0** = DISABLED **1** = ENABLED

Bit	Reset	Description
4	0x0	**TIMESTAMP_OVRFLOW_ERR_CHANSEL_XCPTMISC_MASK**: **0** = DISABLED **1** = ENABLED
3	0x0	**TIMESTAMP_OVRFLOW_ERR_CHANSEL_XCPT_MASK**: **0** = DISABLED **1** = ENABLED
2	0x0	**TIMESTAMP_OVRFLOW_ERR_CHANSEL_MASK**: **0** = DISABLED **1** = ENABLED
1	0x0	**TIMESTAMP_OVRFLOW_ERR_CSIMUX_STREAM_MASK**: **0** = DISABLED **1** = ENABLED
0	0x0	**TIMESTAMP_OVRFLOW_ERR_CSIMUX_FRAME_MASK**: **0** = DISABLED **1** = ENABLED

VI_NOTIFY_ECC_ERR_INJ_0

Offset: 0x2809
Byte Offset: 0xa024
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x01ff01ff (0bxxxx,xxx1,1111,1111,xxxx,xxx1,1111,1111)

Bit	Reset	Description
24	0x1	**DED_ERR_INJ_ENABLE_MISC**
23	0x1	**DED_ERR_INJ_ENABLE_ATOMP**
22	0x1	**DED_ERR_INJ_ENABLE_ATOMPSTATUS**
21	0x1	**DED_ERR_INJ_ENABLE_ATOMPDVFS**
20	0x1	**DED_ERR_INJ_ENABLE_CHANSEL_XCPTMISC**
19	0x1	**DED_ERR_INJ_ENABLE_CHANSEL_XCPT**
18	0x1	**DED_ERR_INJ_ENABLE_CHANSEL**
17	0x1	**DED_ERR_INJ_ENABLE_CSIMUX_STREAM**
16	0x1	**DED_ERR_INJ_ENABLE_CSIMUX_FRAME**: double ecc error injection enable

Bit	Reset	Description
8	0x1	**SEC_ERR_INJ_ENABLE_MISC**
7	0x1	**SEC_ERR_INJ_ENABLE_ATOMP**
6	0x1	**SEC_ERR_INJ_ENABLE_ATOMPSTATUS**
5	0x1	**SEC_ERR_INJ_ENABLE_ATOMPDVFS**
4	0x1	**SEC_ERR_INJ_ENABLE_CHANSEL_XCPTMISC**
3	0x1	**SEC_ERR_INJ_ENABLE_CHANSEL_XCPT**
2	0x1	**SEC_ERR_INJ_ENABLE_CHANSEL**
1	0x1	**SEC_ERR_INJ_ENABLE_CSIMUX_STREAM**
0	0x1	**SEC_ERR_INJ_ENABLE_CSIMUX_FRAME**: single ecc error injection enable

VI_NOTIFY_TIMESTAMP_CTRL_0

Offset: 0x280a
Byte Offset: 0xa028
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: to enable notify TSC sync FIFO rd request propagate downwards. Enable this bit before data transition. trying to gate off unexpected ram X data when pwrbus_ram_pd is high during sync reset period.

VI_FALCON_CSB_HWR_DATA_1_0

Offset: 0x3000
Byte Offset: 0xc000
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
24:14	0x0	**SECGROUP**: SECGROUP bits from HWR bus.
13:4	0x0	**CHANNEL**: CHANNEL bits from HWR bus. (concatenation of VMID and channel)
3	0x0	**CTXSW**: CTXSW bit from HWR bus.
2	0x0	**COR**: COR bit from HWR bus. (class = 0, register = 1)
1	0x0	**RWN**: RWN bit from HWR bus. (write = 0, read = 1)
0	0x0	**VALID**: Whether the popped event was valid.

VI_FALCON_CSB_HWR_DATA_2_0

Offset: 0x3001
Byte Offset: 0xc004
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:20	0x0	**BE**: BE bits from HWR bus.
19:0	0x0	**OFFSET**: OFFSET bits from HWR bus.

VI_FALCON_CSB_HWR_DATA_3_0

Offset: 0x3002
Byte Offset: 0xc008
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**: DATA bits from HWR bus.

VI_FALCON_CSB_HRD_DATA_0

Offset: 0x3003
Byte Offset: 0xc00c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**: Writes to this register cause the DATA field to be sent as a read response.

VI_FALCON_CSB_HRD_FIFO_STATUS_0

Offset: 0x3004
Byte Offset: 0xc010
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,0000,0000,0000,0000,xxxx,xxxx)

Bit	R/W	Reset	Description
23:16	RO	0x0	**CURRENT**: Current occupancy in HRD FIFO.
15:8	RW	0x0	**MAX**: Maximum occupancy reached in HRD FIFO. (Resets to current on write.)
7:0	RO	X	**SIZE**: Number of entries total in HRD FIFO.

VI_FALCON_CSB_HWR_FIFO_STATUS_0

Offset: 0x3005
Byte Offset: 0xc014
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,0000,0000,0000,0000,xxxx,xxxx)

Bit	R/W	Reset	Description
23:16	RO	0x0	**CURRENT**: Current occupancy in HWR FIFO.
15:8	RW	0x0	**MAX**: Maximum occupancy reached in HWR FIFO. (Resets to current on write.)
7:0	RO	X	**SIZE**: Number of entries total in HWR FIFO.

VI_FALCON_CSB_HWR_STREAMID_OFFSET_0

Offset: 0x3006
Byte Offset: 0xc018
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**OFFSET**: Offset into HWR FIFO that secure-STREAMID logic should match on.

VI_FALCON_CSB_HWR_STREAMID_TRANSFER_0

Offset: 0x3007
Byte Offset: 0xc01c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
5:0	X	**INDEX**: Writes to this register cause a STREAMID to be transferred into the lookup table.

VI_FALCON_CSB_AFBIF_STREAMID_CTL_0

Offset: 0x3008
Byte Offset: 0xc020
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000,0000)

Bit	Reset	Description
6:1	0x0	**TABLESEL**
0	0x0	**MUXSEL**

VI_FALCON_CSB_INTERRUPT_0

Offset: 0x3009
Byte Offset: 0xc024
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**RTCPU**
8:0	0x0	**VMID**

VI_FALCON_CSB_STREAMID_LUT_INDEX0_0

Offset: 0x300a
Byte Offset: 0xc028
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**STREAMID**

VI_SYNCGEN0_HCLK_DIV_0

Offset: 0x3200
Byte Offset: 0xc800
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**HALF_CYCLE**: Unsigned floating point. Decimal point position is given by FRAC_BITS in HCLK_DIV_FMT. Frequency of HCLK = SYNCGEN_CLK / (HALF_CYCLE * 2)

VI_SYNCGEN0_HCLK_DIV_FMT_0

Offset: 0x3201
Byte Offset: 0xc804
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
4:0	X	**FRAC_BITS**: Number of fractional bits of HALF_CYCLE

VI_SYNCGEN0_XHS_0

Offset: 0x3202
Byte Offset: 0xc808
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**T_LOW**: Width of XHS, measured in SYNCGEN_CLK

VI_SYNCGEN0_XVS_0

Offset: 0x3203
Byte Offset: 0xc80c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x00XXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
21:8	X	**INTERVAL**: Interval of XVS assertion, measured in HCLK
7:0	X	**T_LOW**: Width of XVS, measured in SYNCGEN_CLK

VI_SYNCGEN0_XVS_TO_XHS_DELAY_0

Offset: 0x3204
Byte Offset: 0xc810
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
1:0	X	**T_DELAY**: Cycles to delay after XVS before assert XHS. 0 means no delay. Measured in SYNCGEN_CLK. Default to 1.

VI_SYNCGEN0_INT_STATUS_0

Offset: 0x3205
Byte Offset: 0xc814
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**SCAN_STOPPED**: Write-1 to clear
3	0x0	**XHS_TIMER**: Write-1 to clear
2	0x0	**STALE_FRAME**: Write-1 to clear
1	0x0	**LOAD_DONE**: Write-1 to clear
0	0x0	**XVS**: Write-1 to clear

VI_SYNCGEN0_INT_MASK_0

Offset: 0x3206
Byte Offset: 0xc818
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**SCAN_STOPPED**: 0=disable interrupt 1=pass interrupt **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
3	0x0	**XHS_TIMER**: 0=disable interrupt 1=pass interrupt **0** = DISABLE **1** = ENABLE
2	0x0	**STALE_FRAME**: 0=disable interrupt 1=pass interrupt **0** = DISABLE **1** = ENABLE
1	0x0	**LOAD_DONE**: 0=disable interrupt 1=pass interrupt **0** = DISABLE **1** = ENABLE
0	0x0	**XVS**: 0=disable interrupt 1=pass interrupt **0** = DISABLE **1** = ENABLE

VI_SYNCGEN0_XHS_TIMER_0

Offset: 0x3207
Byte Offset: 0xc81c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000XXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
12:0	X	**COUNT**: Number of XHS to count before assert XHS timer interrupt

VI_SYNCGEN0_CONTROL_0

Offset: 0x3208
Byte Offset: 0xc820
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00010000 (0bxxxx,xxxx,xxxx,xxx1,xxxx,xxx0,xxxx,xxx0)

Bit	Reset	Description
16	0x1	**CLAMP_SYNC_TO_0**: 0: no clamping, XVS/XHS is as being driven, high as inactive, low as active. 1: clamp XVS/XHS to low level (default)
8	0x0	**INSTANT_STOP**: 0: disable (default) 1: instant stops timing generation if ENABLE is written as 0
0	0x0	**ENABLE**: When INSTANT_STOP == 0, writing ENABLE as 0 does not necessarily stop timing generation immediately. XHS shall be still generated until end of frame is reached. When INSTANT_STOP == 1, writing ENABLE as 0 stops timing generation immediately. **0** = DISABLE **1** = ENABLE

VI_SYNCGEN0_COMMAND_0

Offset: 0x3209
Byte Offset: 0xc824
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Reset	Description
5	0x0	**WR_ACT_SEL**: 0: hosts write shadow registers (default) 1: host writes both shadow and active registers. Debug use only.
4	0x0	**RD_MUX_SEL**: 0: host reads active registers (default) 1: host reads shadow registers. Debug use only.
1	0x0	**AUTOLOAD**: 0: no auto load 1: automatically load a new configuration (as if LOAD were set) at the time that reaching end of current frame
0	0x0	**LOAD**: 0: no op 1: immediate load active register from shadow. LOAD is deferred to the end of the frame, if a frame is active. Always read as 0.

VI_SYNCGEN0_STATUS_0

Offset: 0x320a
Byte Offset: 0xc828
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**PENDING_FORCED_XVS**: Read as 1 when there is a forced XVS pending on XHS counting. 0 otherwise.
0	0x0	**SCANNING**: Read as 1 when timing generator is operative. 0 otherwise.

VI_SYNCGEN0_SCAN_STATUS_0

Offset: 0x320b
Byte Offset: 0xc82c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxx0,0000,0000,0000,xxx0,0000,0000,0000)

Bit	Reset	Description
28:16	0x0	**XVS_COUNT**: Reset to 0 when being written. Increase 1 when XVS changes from inactive to active.
12:0	0x0	**XHS_COUNT**: Reset to 0 when being written. Reset to 0 when XVS changes from inactive to active. Increase 1 when XHS changes from inactive to active.

VI_SYNCGEN0_FORCE_XVS_0

Offset: 0x320c
Byte Offset: 0xc830
Read/Write: R/W

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXX0000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
28:16	X	**XHS_COUNT**: Minimum number of XHS before interruptive XVS is asserted. Interruptive XVS shall not be asserted until XHS_COUNT is reached.
0	0x0	**ASSERT**: 0: no op 1: assert Always read as 0.

VI_SYNCGEN1_HCLK_DIV_0

Offset: 0x3300
Byte Offset: 0xcc00
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**HALF_CYCLE**: Unsigned floating point. Decimal point position is given by FRAC_BITS in HCLK_DIV_FMT. Frequency of HCLK = SYNCGEN_CLK / (HALF_CYCLE * 2)

VI_SYNCGEN1_HCLK_DIV_FMT_0

Offset: 0x3301
Byte Offset: 0xcc04
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
4:0	X	**FRAC_BITS**: Number of fractional bits of HALF_CYCLE

VI_SYNCGEN1_XHS_0

Offset: 0x3302
Byte Offset: 0xcc08
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**T_LOW**: Width of XHS, measured in SYNCGEN_CLK

VI_SYNCGEN1_XVS_0

Offset: 0x3303
Byte Offset: 0xcc0c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x00XXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
21:8	X	**INTERVAL**: Interval of XVS assertion, measured in HCLK
7:0	X	**T_LOW**: Width of XVS, measured in SYNCGEN_CLK

VI_SYNCGEN1_XVS_TO_XHS_DELAY_0

Offset: 0x3304
Byte Offset: 0xcc10
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
1:0	X	**T_DELAY**: Cycles to delay after XVS before assert XHS. 0 means no delay. Measured in SYNCGEN_CLK. Default to 1.

VI_SYNCGEN1_INT_STATUS_0

Offset: 0x3305
Byte Offset: 0xcc14
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**SCAN_STOPPED**: Write-1 to clear
3	0x0	**XHS_TIMER**: Write-1 to clear
2	0x0	**STALE_FRAME**: Write-1 to clear
1	0x0	**LOAD_DONE**: Write-1 to clear
0	0x0	**XVS**: Write-1 to clear

VI_SYNCGEN1_INT_MASK_0

Offset: 0x3306
Byte Offset: 0xcc18
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**SCAN_STOPPED**: 0=disable interrupt 1=pass interrupt **0** = DISABLE **1** = ENABLE
3	0x0	**XHS_TIMER**: 0=disable interrupt 1=pass interrupt **0** = DISABLE **1** = ENABLE
2	0x0	**STALE_FRAME**: 0=disable interrupt 1=pass interrupt **0** = DISABLE **1** = ENABLE
1	0x0	**LOAD_DONE**: 0=disable interrupt 1=pass interrupt **0** = DISABLE **1** = ENABLE
0	0x0	**XVS**: 0=disable interrupt 1=pass interrupt **0** = DISABLE **1** = ENABLE

VI_SYNCGEN1_XHS_TIMER_0

Offset: 0x3307
Byte Offset: 0xcc1c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000XXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
12:0	X	**COUNT**: Number of XHS to count before assert XHS timer interrupt

VI_SYNCGEN1_CONTROL_0

Offset: 0x3308
Byte Offset: 0xcc20
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00010000 (0bxxxx,xxxx,xxxx,xxx1,xxxx,xxx0,xxxx,xxx0)

Bit	Reset	Description
16	0x1	**CLAMP_SYNC_TO_0**: 0: no clamping, XVS/XHS is as being driven, high as inactive, low as active. 1: clamp XVS/XHS to low level (default)
8	0x0	**INSTANT_STOP**: 0: disable (default) 1: instant stops timing generation if ENABLE is written as 0
0	0x0	**ENABLE**: When INSTANT_STOP == 0, writing ENABLE as 0 does not necessarily stop timing generation immediately. XHS shall be still generated until end of frame is reached. When INSTANT_STOP == 1, writing ENABLE as 0 stops timing generation immediately. **0** = DISABLE **1** = ENABLE

VI_SYNCGEN1_COMMAND_0

Offset: 0x3309
Byte Offset: 0xcc24
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Reset	Description
5	0x0	**WR_ACT_SEL**: 0: hosts write shadow registers (default) 1: host writes both shadow and active registers. Debug use only.
4	0x0	**RD_MUX_SEL**: 0: host reads active registers (default) 1: host reads shadow registers. Debug use only.
1	0x0	**AUTOLOAD**: 0: no auto load 1: automatically load a new configuration (as if LOAD were set) at the time that reaching end of current frame

Bit	Reset	Description
0	0x0	**LOAD**: 0: no op 1: immediate load active register from shadow. LOAD is deferred to the end of the frame, if a frame is active. Always read as 0.

VI_SYNCGEN1_STATUS_0

Offset: 0x330a
Byte Offset: 0xcc28
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**PENDING_FORCED_XVS**: Read as 1 when there is a forced XVS pending on XHS counting. 0 otherwise.
0	0x0	**SCANNING**: Read as 1 when timing generator is operative. 0 otherwise.

VI_SYNCGEN1_SCAN_STATUS_0

Offset: 0x330b
Byte Offset: 0xcc2c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxx0,0000,0000,0000,xxx0,0000,0000,0000)

Bit	Reset	Description
28:16	0x0	**XVS_COUNT**: Reset to 0 when being written. Increase 1 when XVS changes from inactive to active.
12:0	0x0	**XHS_COUNT**: Reset to 0 when being written. Reset to 0 when XVS changes from inactive to active. Increase 1 when XHS changes from inactive to active.

VI_SYNCGEN1_FORCE_XVS_0

Offset: 0x330c
Byte Offset: 0xcc30
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXX0000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
28:16	X	**XHS_COUNT**: Minimum number of XHS before interruptive XVS is asserted. Interruptive XVS shall not be asserted until XHS_COUNT is reached.
0	0x0	**ASSERT**: 0: no op 1: assert Always read as 0.

VI_SYNCGEN2_HCLK_DIV_0

Offset: 0x3400
Byte Offset: 0xd000
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**HALF_CYCLE**: Unsigned floating point. Decimal point position is given by FRAC_BITS in HCLK_DIV_FMT. Frequency of HCLK = SYNCGEN_CLK / (HALF_CYCLE * 2)

VI_SYNCGEN2_HCLK_DIV_FMT_0

Offset: 0x3401
Byte Offset: 0xd004
Read/Write: R/W
Parity Protection: N

Shadow: Y
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
4:0	X	**FRAC_BITS**: Number of fractional bits of HALF_CYCLE

VI_SYNCGEN2_XHS_0

Offset: 0x3402
Byte Offset: 0xd008
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**T_LOW**: Width of XHS, measured in SYNCGEN_CLK

VI_SYNCGEN2_XVS_0

Offset: 0x3403
Byte Offset: 0xd00c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x00XXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
21:8	X	**INTERVAL**: Interval of XVS assertion, measured in HCLK
7:0	X	**T_LOW**: Width of XVS, measured in SYNCGEN_CLK

VI_SYNCGEN2_XVS_TO_XHS_DELAY_0

Offset: 0x3404
Byte Offset: 0xd010
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
1:0	X	**T_DELAY**: Cycles to delay after XVS before assert XHS. 0 means no delay. Measured in SYNCGEN_CLK. Default to 1.

VI_SYNCGEN2_INT_STATUS_0

Offset: 0x3405
Byte Offset: 0xd014
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**SCAN_STOPPED**: Write-1 to clear
3	0x0	**XHS_TIMER**: Write-1 to clear
2	0x0	**STALE_FRAME**: Write-1 to clear
1	0x0	**LOAD_DONE**: Write-1 to clear
0	0x0	**XVS**: Write-1 to clear

VI_SYNCGEN2_INT_MASK_0

Offset: 0x3406
Byte Offset: 0xd018
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**SCAN_STOPPED**: 0=disable interrupt 1=pass interrupt **0** = DISABLE **1** = ENABLE
3	0x0	**XHS_TIMER**: 0=disable interrupt 1=pass interrupt **0** = DISABLE **1** = ENABLE
2	0x0	**STALE_FRAME**: 0=disable interrupt 1=pass interrupt **0** = DISABLE **1** = ENABLE
1	0x0	**LOAD_DONE**: 0=disable interrupt 1=pass interrupt **0** = DISABLE **1** = ENABLE
0	0x0	**XVS**: 0=disable interrupt 1=pass interrupt **0** = DISABLE **1** = ENABLE

VI_SYNCGEN2_XHS_TIMER_0

Offset: 0x3407
Byte Offset: 0xd01c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000XXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
12:0	X	**COUNT**: Number of XHS to count before assert XHS timer interrupt

VI_SYNCGEN2_CONTROL_0

Offset: 0x3408
Byte Offset: 0xd020
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00010000 (0bxxxx,xxxx,xxxx,xxx1,xxxx,xxx0,xxxx,xxx0)

Bit	Reset	Description
16	0x1	**CLAMP_SYNC_TO_0**: 0: no clamping, XVS/XHS is as being driven, high as inactive, low as active. 1: clamp XVS/XHS to low level (default)
8	0x0	**INSTANT_STOP**: 0: disable (default) 1: instant stops timing generation if ENABLE is written as 0
0	0x0	**ENABLE**: When INSTANT_STOP == 0, writing ENABLE as 0 does not necessarily stop timing generation immediately. XHS shall be still generated until end of frame is reached. When INSTANT_STOP == 1, writing ENABLE as 0 stops timing generation immediately. **0** = DISABLE **1** = ENABLE

VI_SYNCGEN2_COMMAND_0

Offset: 0x3409
Byte Offset: 0xd024
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Reset	Description
5	0x0	**WR_ACT_SEL**: 0: hosts write shadow registers (default) 1: host writes both shadow and active registers. Debug use only.
4	0x0	**RD_MUX_SEL**: 0: host reads active registers (default) 1: host reads shadow registers. Debug use only.
1	0x0	**AUTOLOAD**: 0: no auto load 1: automatically load a new configuration (as if LOAD were set) at the time that reaching end of current frame
0	0x0	**LOAD**: 0: no op 1: immediate load active register from shadow. LOAD is deferred to the end of the frame, if a frame is active. Always read as 0.

VI_SYNCGEN2_STATUS_0

Offset: 0x340a
Byte Offset: 0xd028
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**PENDING_FORCED_XVS**: Read as 1 when there is a forced XVS pending on XHS counting. 0 otherwise.
0	0x0	**SCANNING**: Read as 1 when timing generator is operative. 0 otherwise.

VI_SYNCGEN2_SCAN_STATUS_0

Offset: 0x340b
Byte Offset: 0xd02c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxx0,0000,0000,0000,xxx0,0000,0000,0000)

Bit	Reset	Description
28:16	0x0	**XVS_COUNT**: Reset to 0 when being written. Increase 1 when XVS changes from inactive to active.
12:0	0x0	**XHS_COUNT**: Reset to 0 when being written. Reset to 0 when XVS changes from inactive to active. Increase 1 when XHS changes from inactive to active.

VI_SYNCGEN2_FORCE_XVS_0

Offset: 0x340c
Byte Offset: 0xd030
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXX0000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
28:16	X	**XHS_COUNT**: Minimum number of XHS before interruptive XVS is asserted. Interruptive XVS shall not be asserted until XHS_COUNT is reached.
0	0x0	**ASSERT**: 0: no op 1: assert Always read as 0.

VI_PM_VI_PERFMON_LOCALTRIG_A_0

Offset: 0x3600
Byte Offset: 0xd800
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:16	0x0	**LOCAL_BKMKA**: perfmon local bookmark A

Bit	Reset	Description
0	0x0	**LOCAL_TRIGA**: perfmon local trigger A

VI_PM_VI_PERFMON_LOCALTRIG_B_0

Offset: 0x3601
Byte Offset: 0xd804
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:16	0x0	**LOCAL_BKMKB**: perfmon local bookmark B
0	0x0	**LOCAL_TRIGB**: perfmon local trigger B

VI_PM_VI_PERFMUX_CONTROL_0

Offset: 0x3602
Byte Offset: 0xd808
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,00xx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**PM_EN**: **0** = OFF **1** = ON
15:14	0x0	**PM_V_BLANK_SEL**: when this sel is 2'h1, vi2pm_chx_in_v_blank is actually connected to chansel output chx Frame Start (FS) signal. when this sel is 2'h2, vi2pm_chx_in_v_blank is actually connected to chansel output chx Frame End (FE) signal.

VI_PM_VI_PERFMUX_CONTROL_1_0

Offset: 0x3603
Byte Offset: 0xd80c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**PM_SEL_3**: select signal for perfmon_mux bit 3.
23:16	0x0	**PM_SEL_2**: select signal for perfmon_mux bit 2.
15:8	0x0	**PM_SEL_1**: sel9ect signal for perfmon_mux bit 1.
7:0	0x0	**PM_SEL_0**: select signal for perfmon_mux bit 0. **0** = VI2PM_STATIC_PATTERN_1_1 **1** = VI2PM_STATIC_PATTERN_0_1 **2** = VI2PM_VI2MSS_WRITE_REQUEST **3** = VI2PM_MSS2VI_WRITE_RESPONSE **4** = VI2PM_VI2MSS_WDATA **5** = VI2PM_VI2MSS_WSTALLS **6** = VI2PM_VI_BUSY **7** = VI2PM_CH0_PIXEL_VALID **8** = VI2PM_CH1_PIXEL_VALID **9** = VI2PM_CH2_PIXEL_VALID **10** = VI2PM_CH3_PIXEL_VALID **11** = VI2PM_CH4_PIXEL_VALID **12** = VI2PM_CH5_PIXEL_VALID **13** = VI2PM_CH6_PIXEL_VALID **14** = VI2PM_CH7_PIXEL_VALID **15** = VI2PM_CH8_PIXEL_VALID **16** = VI2PM_CH9_PIXEL_VALID **17** = VI2PM_CH10_PIXEL_VALID **18** = VI2PM_CH11_PIXEL_VALID **19** = VI2PM_CH12_PIXEL_VALID **20** = VI2PM_CH13_PIXEL_VALID **21** = VI2PM_CH14_PIXEL_VALID **22** = VI2PM_CH15_PIXEL_VALID **23** = VI2PM_CH16_PIXEL_VALID **24** = VI2PM_CH17_PIXEL_VALID **25** = VI2PM_CH18_PIXEL_VALID **26** = VI2PM_CH19_PIXEL_VALID **27** = VI2PM_CH20_PIXEL_VALID **28** = VI2PM_CH21_PIXEL_VALID **29** = VI2PM_CH22_PIXEL_VALID **30** = VI2PM_CH23_PIXEL_VALID **31** = VI2PM_CH24_PIXEL_VALID **32** = VI2PM_CH25_PIXEL_VALID **33** = VI2PM_CH26_PIXEL_VALID **34** = VI2PM_CH27_PIXEL_VALID **35** = VI2PM_CH28_PIXEL_VALID **36** = VI2PM_CH29_PIXEL_VALID **37** = VI2PM_CH30_PIXEL_VALID **38** = VI2PM_CH31_PIXEL_VALID **39** = VI2PM_CH32_PIXEL_VALID **40** = VI2PM_CH33_PIXEL_VALID **41** = VI2PM_CH34_PIXEL_VALID

Bit	Reset	Description
		42 = VI2PM_CH35_PIXEL_VALID **43** = VI2PM_CH0_IN_H_BLANK **44** = VI2PM_CH1_IN_H_BLANK **45** = VI2PM_CH2_IN_H_BLANK **46** = VI2PM_CH3_IN_H_BLANK **47** = VI2PM_CH4_IN_H_BLANK **48** = VI2PM_CH5_IN_H_BLANK **49** = VI2PM_CH6_IN_H_BLANK **50** = VI2PM_CH7_IN_H_BLANK **51** = VI2PM_CH8_IN_H_BLANK **52** = VI2PM_CH9_IN_H_BLANK **53** = VI2PM_CH10_IN_H_BLANK **54** = VI2PM_CH11_IN_H_BLANK **55** = VI2PM_CH12_IN_H_BLANK **56** = VI2PM_CH13_IN_H_BLANK **57** = VI2PM_CH14_IN_H_BLANK **58** = VI2PM_CH15_IN_H_BLANK **59** = VI2PM_CH16_IN_H_BLANK **60** = VI2PM_CH17_IN_H_BLANK **61** = VI2PM_CH18_IN_H_BLANK **62** = VI2PM_CH19_IN_H_BLANK **63** = VI2PM_CH20_IN_H_BLANK **64** = VI2PM_CH21_IN_H_BLANK **65** = VI2PM_CH22_IN_H_BLANK **66** = VI2PM_CH23_IN_H_BLANK **67** = VI2PM_CH24_IN_H_BLANK **68** = VI2PM_CH25_IN_H_BLANK **69** = VI2PM_CH26_IN_H_BLANK **70** = VI2PM_CH27_IN_H_BLANK **71** = VI2PM_CH28_IN_H_BLANK **72** = VI2PM_CH29_IN_H_BLANK **73** = VI2PM_CH30_IN_H_BLANK **74** = VI2PM_CH31_IN_H_BLANK **75** = VI2PM_CH32_IN_H_BLANK **76** = VI2PM_CH33_IN_H_BLANK **77** = VI2PM_CH34_IN_H_BLANK **78** = VI2PM_CH35_IN_H_BLANK **79** = VI2PM_CH0_IN_V_BLANK **80** = VI2PM_CH1_IN_V_BLANK **81** = VI2PM_CH2_IN_V_BLANK **82** = VI2PM_CH3_IN_V_BLANK **83** = VI2PM_CH4_IN_V_BLANK **84** = VI2PM_CH5_IN_V_BLANK **85** = VI2PM_CH6_IN_V_BLANK **86** = VI2PM_CH7_IN_V_BLANK **87** = VI2PM_CH8_IN_V_BLANK **88** = VI2PM_CH9_IN_V_BLANK **89** = VI2PM_CH10_IN_V_BLANK **90** = VI2PM_CH11_IN_V_BLANK **91** = VI2PM_CH12_IN_V_BLANK **92** = VI2PM_CH13_IN_V_BLANK **93** = VI2PM_CH14_IN_V_BLANK **94** = VI2PM_CH15_IN_V_BLANK **95** = VI2PM_CH16_IN_V_BLANK **96** = VI2PM_CH17_IN_V_BLANK **97** = VI2PM_CH18_IN_V_BLANK **98** = VI2PM_CH19_IN_V_BLANK **99** = VI2PM_CH20_IN_V_BLANK **100** = VI2PM_CH21_IN_V_BLANK **101** = VI2PM_CH22_IN_V_BLANK **102** = VI2PM_CH23_IN_V_BLANK **103** = VI2PM_CH24_IN_V_BLANK **104** = VI2PM_CH25_IN_V_BLANK **105** = VI2PM_CH26_IN_V_BLANK **106** = VI2PM_CH27_IN_V_BLANK **107** = VI2PM_CH28_IN_V_BLANK

Bit	Reset	Description
		108 = VI2PM_CH29_IN_V_BLANK **109** = VI2PM_CH30_IN_V_BLANK **110** = VI2PM_CH31_IN_V_BLANK **111** = VI2PM_CH32_IN_V_BLANK **112** = VI2PM_CH33_IN_V_BLANK **113** = VI2PM_CH34_IN_V_BLANK **114** = VI2PM_CH35_IN_V_BLANK **115** = VI2PM_STREAM_FIRST_ATOM_LATENCY_0 **116** = VI2PM_STREAM_FIRST_ATOM_LATENCY_1 **117** = VI2PM_STREAM_FIRST_ATOM_LATENCY_2 **118** = VI2PM_STREAM_FIRST_ATOM_LATENCY_3 **119** = VI2PM_STREAM_FIRST_ATOM_LATENCY_4 **120** = VI2PM_STREAM_FIRST_ATOM_LATENCY_5 **121** = VI2PM_STREAM_FIRST_ATOM_LATENCY_6 **122** = VI2PM_STREAM_FIRST_ATOM_LATENCY_7 **123** = VI2PM_STREAM_FIRST_ATOM_LATENCY_8 **124** = VI2PM_STREAM_FIRST_ATOM_LATENCY_9 **125** = VI2PM_STREAM_FIRST_ATOM_LATENCY_10 **126** = VI2PM_STREAM_FIRST_ATOM_LATENCY_11 **127** = VI2PM_STREAM_FIRST_ATOM_LATENCY_12 **128** = VI2PM_STREAM_FIRST_ATOM_LATENCY_13 **129** = VI2PM_STREAM_FIRST_ATOM_LATENCY_14 **130** = VI2PM_STREAM_FIRST_ATOM_LATENCY_15 **131** = VI2PM_STREAM_FIRST_ATOM_LATENCY_16 **132** = VI2PM_STREAM_FIRST_ATOM_LATENCY_17 **133** = VI2PM_STREAM_DONE_FLAG_LATENCY_0 **134** = VI2PM_STREAM_DONE_FLAG_LATENCY_1 **135** = VI2PM_STREAM_DONE_FLAG_LATENCY_2 **136** = VI2PM_STREAM_DONE_FLAG_LATENCY_3 **137** = VI2PM_STREAM_DONE_FLAG_LATENCY_4 **138** = VI2PM_STREAM_DONE_FLAG_LATENCY_5 **139** = VI2PM_STREAM_DONE_FLAG_LATENCY_6 **140** = VI2PM_STREAM_DONE_FLAG_LATENCY_7 **141** = VI2PM_STREAM_DONE_FLAG_LATENCY_8 **142** = VI2PM_STREAM_DONE_FLAG_LATENCY_9 **143** = VI2PM_STREAM_DONE_FLAG_LATENCY_10 **144** = VI2PM_STREAM_DONE_FLAG_LATENCY_11 **145** = VI2PM_STREAM_DONE_FLAG_LATENCY_12 **146** = VI2PM_STREAM_DONE_FLAG_LATENCY_13 **147** = VI2PM_STREAM_DONE_FLAG_LATENCY_14 **148** = VI2PM_STREAM_DONE_FLAG_LATENCY_15 **149** = VI2PM_STREAM_DONE_FLAG_LATENCY_16 **150** = VI2PM_STREAM_DONE_FLAG_LATENCY_17 **151** = VI2PM_CH0_BUSY_WITH_LATENCY **152** = VI2PM_CH1_BUSY_WITH_LATENCY **153** = VI2PM_CH2_BUSY_WITH_LATENCY **154** = VI2PM_CH3_BUSY_WITH_LATENCY **155** = VI2PM_CH4_BUSY_WITH_LATENCY **156** = VI2PM_CH5_BUSY_WITH_LATENCY **157** = VI2PM_CH6_BUSY_WITH_LATENCY **158** = VI2PM_CH7_BUSY_WITH_LATENCY **159** = VI2PM_CH8_BUSY_WITH_LATENCY **160** = VI2PM_CH9_BUSY_WITH_LATENCY **161** = VI2PM_CH10_BUSY_WITH_LATENCY **162** = VI2PM_CH11_BUSY_WITH_LATENCY **163** = VI2PM_CH12_BUSY_WITH_LATENCY **164** = VI2PM_CH13_BUSY_WITH_LATENCY **165** = VI2PM_CH14_BUSY_WITH_LATENCY **166** = VI2PM_CH15_BUSY_WITH_LATENCY **167** = VI2PM_CH16_BUSY_WITH_LATENCY **168** = VI2PM_CH17_BUSY_WITH_LATENCY **169** = VI2PM_CH18_BUSY_WITH_LATENCY **170** = VI2PM_CH19_BUSY_WITH_LATENCY **171** = VI2PM_CH20_BUSY_WITH_LATENCY **172** = VI2PM_CH21_BUSY_WITH_LATENCY **173** = VI2PM_CH22_BUSY_WITH_LATENCY

Bit	Reset	Description
		174 = VI2PM_CH23_BUSY_WITH_LATENCY **175** = VI2PM_CH24_BUSY_WITH_LATENCY **176** = VI2PM_CH25_BUSY_WITH_LATENCY **177** = VI2PM_CH26_BUSY_WITH_LATENCY **178** = VI2PM_CH27_BUSY_WITH_LATENCY **179** = VI2PM_CH28_BUSY_WITH_LATENCY **180** = VI2PM_CH29_BUSY_WITH_LATENCY **181** = VI2PM_CH30_BUSY_WITH_LATENCY **182** = VI2PM_CH31_BUSY_WITH_LATENCY **183** = VI2PM_CH32_BUSY_WITH_LATENCY **184** = VI2PM_CH33_BUSY_WITH_LATENCY **185** = VI2PM_CH34_BUSY_WITH_LATENCY **186** = VI2PM_CH35_BUSY_WITH_LATENCY **187** = VI2PM_STREAM0_PIXEL_VALID **188** = VI2PM_STREAM1_PIXEL_VALID **189** = VI2PM_STREAM2_PIXEL_VALID **190** = VI2PM_STREAM3_PIXEL_VALID **191** = VI2PM_STREAM4_PIXEL_VALID **192** = VI2PM_STREAM5_PIXEL_VALID **193** = VI2PM_FW_PERF_CH0 **194** = VI2PM_FW_PERF_CH1 **195** = VI2PM_FW_PERF_CH2 **196** = VI2PM_FW_PERF_CH3 **197** = VI2PM_FW_PERF_CH4 **198** = VI2PM_FW_PERF_CH5 **199** = VI2PM_FW_PERF_CH6 **200** = VI2PM_FW_PERF_CH7 **201** = VI2PM_FW_PERF_CH8 **202** = VI2PM_FW_PERF_CH9 **203** = VI2PM_FW_PERF_CH10 **204** = VI2PM_FW_PERF_CH11 **205** = VI2PM_FW_PERF_CH12 **206** = VI2PM_FW_PERF_CH13 **207** = VI2PM_FW_PERF_CH14 **208** = VI2PM_FW_PERF_CH15 **209** = VI2PM_FW_PERF_CH16 **210** = VI2PM_FW_PERF_CH17 **211** = VI2PM_FW_PERF_CH18 **212** = VI2PM_FW_PERF_CH19 **213** = VI2PM_FW_PERF_CH20 **214** = VI2PM_FW_PERF_CH21 **215** = VI2PM_FW_PERF_CH22 **216** = VI2PM_FW_PERF_CH23 **217** = VI2PM_FW_PERF_CH24 **218** = VI2PM_FW_PERF_CH25 **219** = VI2PM_FW_PERF_CH26 **220** = VI2PM_FW_PERF_CH27 **221** = VI2PM_FW_PERF_CH28 **222** = VI2PM_FW_PERF_CH29 **223** = VI2PM_FW_PERF_CH30 **224** = VI2PM_FW_PERF_CH31 **225** = VI2PM_FW_PERF_CH32 **226** = VI2PM_FW_PERF_CH33 **227** = VI2PM_FW_PERF_CH34 **228** = VI2PM_FW_PERF_CH35 **229** = VI2PM_WR_OR_CNT_0 **230** = VI2PM_WR_OR_CNT_1 **231** = VI2PM_WR_OR_CNT_2 **232** = VI2PM_WR_OR_CNT_3 **233** = VI2PM_WR_OR_CNT_4 **234** = VI2PM_WR_OR_CNT_5 **235** = VI2PM_WR_OR_CNT_6 **236** = VI2PM_WR_OR_CNT_7 **237** = VI2PM_WR_OR_CNT_8 **238** = VI2PM_WR_OR_CNT_LT_16 **239** = VI2PM_WR_TAOR_CNT_MSB

VI_PM_VI_PERFMUX_CONTROL_2_0

Offset: 0x3604
Byte Offset: 0xd810
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**PM_SEL_7**: select signal for perfmon_mux bit 7.
23:16	0x0	**PM_SEL_6**: select signal for perfmon_mux bit 6.
15:8	0x0	**PM_SEL_5**: select signal for perfmon_mux bit 5.
7:0	0x0	**PM_SEL_4**: select signal for perfmon_mux bit 4.

VI_PM_VI_PERFMUX_CONTROL_3_0

Offset: 0x3605
Byte Offset: 0xd814
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**PM_SEL_11**: select signal for perfmon_mux bit 11.
23:16	0x0	**PM_SEL_10**: select signal for perfmon_mux bit 10.
15:8	0x0	**PM_SEL_9**: select signal for perfmon_mux bit 9.
7:0	0x0	**PM_SEL_8**: select signal for perfmon_mux bit 8.

VI_PM_VI_PERFMUX_CONTROL_4_0

Offset: 0x3606
Byte Offset: 0xd818
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**PM_SEL_15**: select signal for perfmon_mux bit 15.
23:16	0x0	**PM_SEL_14**: select signal for perfmon_mux bit 14.
15:8	0x0	**PM_SEL_13**: select signal for perfmon_mux bit 13.
7:0	0x0	**PM_SEL_12**: select signal for perfmon_mux bit 12.

7.2.2.3.2 Video Input Channel 0-10 Registers

Channel 0 Specific Registers

VI_CH0_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x4000
Byte Offset: 0x10000
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)

Bit	Parity Protection	Reset	Description
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH0_CHANSEL_0

Offset: 0x4004
Byte Offset: 0x10010
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH0_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x4005
Byte Offset: 0x10014
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH0_CONTROL_0

Channel prefix gets added to register name (VI_CH*)

Offset: 0x4006
Byte Offset: 0x10018
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.

Bit	Parity Protection	Reset	Description
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH0_MATCH_0

SHADOWED CHANSEL REGISTERS

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x4007
Byte Offset: 0x1001c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH0_MATCH_DOL_0

Offset: 0x4008
Byte Offset: 0x10020
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH0_MATCH_VC_HI_0

Offset: 0x4009
Byte Offset: 0x10024
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH0_MATCH_DATATYPE_0

Offset: 0x400a
Byte Offset: 0x10028
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH0_MATCH_FRAMEID_0

Offset: 0x400b
Byte Offset: 0x1002c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH0_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT.

The one exception is embedded data type which does not get remapped through this register.

If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on.

You are also likely not to have a wildcard but an exact match for data type.

It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}).

Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined prior to any override mapping. Should any incoming pixel after LS not match the lock in type then a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type. This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x400c
Byte Offset: 0x10030
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH0_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x400d
Byte Offset: 0x10034
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH0_FRAME_Y_0

Expected frame y dimension

Offset: 0x400e
Byte Offset: 0x10038
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH0_EMBED_X_0

Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x400f
Byte Offset: 0x1003c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH0_EMBED_Y_0

Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x4010
Byte Offset: 0x10040
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH0_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached.

The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x4011
Byte Offset: 0x10044
Read/Write: R/W
Parity Protection: Y

Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH0_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started.

For example starting a flash in anticipation of an upcoming frame, or setup up a focusing process.This is done setting a tripline count which flags a notification once the number of lines has been reached.

The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

The line timer events are not filtered by the cropped region, they are solely based on the input frame.The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x4012
Byte Offset: 0x10048
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.

Bit	Reset	Description
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH0_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x4013
Byte Offset: 0x1004c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH0_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use.

If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x4014
Byte Offset: 0x10050
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH0_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that:

1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x4015
Byte Offset: 0x10054
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH0_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that:

1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x4016
Byte Offset: 0x10058
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH0_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) - SKIP_Y_LINES

Offset: 0x4017
Byte Offset: 0x1005c
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH0_OUT_Y_0

Offset: 0x4018
Byte Offset: 0x10060
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH0_NOTIFY_MASK_0

Offset: 0x4019
Byte Offset: 0x10064
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line

Bit	Reset	Description
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH0_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED. The expected use of NOMATCH is for debug purposes.

If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification.

To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x401a
Byte Offset: 0x10068
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH0_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x401b
Byte Offset: 0x1006c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH0_FRAME_COUNT_0

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame.

If in frame then FS=FE+1. FE can be perceived as a checksum check for FS.

Write any value to register to clear all counts.

Offset: 0x401c
Byte Offset: 0x10070
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH0_LINE_TIMER_FIRST_0

Offset: 0x401d
Byte Offset: 0x10074
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH0_FLUSH_FIRST_0

Offset: 0x401e
Byte Offset: 0x10078
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH0_DOL_CTRL_0

Offset: 0x401f
Byte Offset: 0x1007c
Read/Write: R/W
Parity Protection: Y

Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH0_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x4020
Byte Offset: 0x10080
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH0_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x4021
Byte Offset: 0x10084
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH0_PIXFMT_FORMAT_0

Offset: 0x4022
Byte Offset: 0x10088
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/ RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0

Bit	Reset	Description
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH0_PIXFMT_WIDE_0

Offset: 0x4023
Byte Offset: 0x1008c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH0_PIXFMT_PDAF_CROP_X_0

Offset: 0x4024
Byte Offset: 0x10090
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH0_PIXFMT_PDAF_CROP_Y_0

Offset: 0x4025
Byte Offset: 0x10094
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends

Bit	Reset	Description
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH0_PIXFMT_PDAF_CONFIG0_0

Offset: 0x4026
Byte Offset: 0x10098
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH0_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x4027
Byte Offset: 0x1009c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH0_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x4028
Byte Offset: 0x100a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH0_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.
Offset: 0x4029
Byte Offset: 0x100a4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH0_PIXFMT_PDAF_CONFIG1_0

Offset: 0x402a
Byte Offset: 0x100a8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH0_DPCM_STRIP_0

Offset: 0x402f
Byte Offset: 0x100bc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH0_DPCM_CHUNK_FIRST_0

Offset: 0x4030
Byte Offset: 0x100c0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH0_DPCM_CHUNK_BODY_0

Offset: 0x4031
Byte Offset: 0x100c4
Read/Write: R/W
Parity Protection: Y

Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH0_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x4032
Byte Offset: 0x100c8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH0_DPCM_CHUNK_LAST_0

Offset: 0x4033
Byte Offset: 0x100cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH0_DPCM_STATISTICS_0_0

Offset: 0x4034
Byte Offset: 0x100d0
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH0_DPCM_STATISTICS_1_0

Offset: 0x4035
Byte Offset: 0x100d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH0_DPCM_MODE_0

Offset: 0x4036
Byte Offset: 0x100d8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH0_DPCM_CLAMP_HIGH_0

Offset: 0x4037
Byte Offset: 0x100dc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH0_DPCM_CLAMP_LOW_0

Offset: 0x4038
Byte Offset: 0x100e0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH0_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x403b
Byte Offset: 0x100ec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH0_ATOMP_SURFACE_OFFSET0_0

Offset: 0x403c
Byte Offset: 0x100f0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH0_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x403d
Byte Offset: 0x100f4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH0_ATOMP_SURFACE_STRIDE0_0

Offset: 0x403e
Byte Offset: 0x100f8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH0_ATOMP_DPCM_CHUNK_0

Offset: 0x403f
Byte Offset: 0x100fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH0_ATOMP_SURFACE_OFFSET1_0

Offset: 0x4040
Byte Offset: 0x10100
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH0_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x4041
Byte Offset: 0x10104
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH0_ATOMP_SURFACE_STRIDE1_0

Offset: 0x4042
Byte Offset: 0x10108
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH0_ATOMP_SURFACE_OFFSET2_0

Offset: 0x4043
Byte Offset: 0x1010c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH0_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x4044
Byte Offset: 0x10110
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH0_ATOMP_SURFACE_STRIDE2_0

Offset: 0x4045
Byte Offset: 0x10114
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH0_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x4046
Byte Offset: 0x10118
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH0_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x4047
Byte Offset: 0x1011c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH0_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x4048
Byte Offset: 0x10120
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH0_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x4049
Byte Offset: 0x10124
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH0_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x404a
Byte Offset: 0x10128
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH0_ATOMP_HIGH_PRI_REQ_0

Offset: 0x404b
Byte Offset: 0x1012c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH0_ATOMP_RESERVE_0

Offset: 0x404c
Byte Offset: 0x10130
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 1 Specific Registers

VI_CH1_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x4100
Byte Offset: 0x10400
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH1_CHANSEL_0

Offset: 0x4104
Byte Offset: 0x10410
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH1_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x4105
Byte Offset: 0x10414
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH1_CONTROL_0

Channel prefix gets added to register name (VI_CH*)

Offset: 0x4106
Byte Offset: 0x10418
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.

Bit	Parity Protection	Reset	Description
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH1_MATCH_0

SHADOWED CHANSEL REGISTERS

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x4107
Byte Offset: 0x1041c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH1_MATCH_DOL_0

Offset: 0x4108
Byte Offset: 0x10420
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH1_MATCH_VC_HI_0

Offset: 0x4109
Byte Offset: 0x10424
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH1_MATCH_DATATYPE_0

Offset: 0x410a
Byte Offset: 0x10428
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH1_MATCH_FRAMEID_0

Offset: 0x410b
Byte Offset: 0x1042c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH1_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT. The one exception is embedded data type which does not get remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type. Note that if a channel uses data type interleaving it is most likely not to have this feature on. You are also likely not to have a wildcard but an exact match for data type. It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined prior to any override mapping. Should any incoming pixel after LS not match the lock in type then a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type. This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x410c
Byte Offset: 0x10430
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH1_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x410d
Byte Offset: 0x10434
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH1_FRAME_Y_0

Expected frame y dimension

Offset: 0x410e
Byte Offset: 0x10438
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH1_EMBED_X_0

Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x410f
Byte Offset: 0x1043c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH1_EMBED_Y_0

Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x4110
Byte Offset: 0x10440
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH1_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data.This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached.

The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed. Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x4111
Byte Offset: 0x10444
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH1_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started.

For example starting a flash in anticipation of an upcoming frame, or setup up a focusing process.

This is done setting a tripline count which flags a notification once the number of lines has been reached.

The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

The line timer events are not filtered by the cropped region, they are solely based on the input frame.

The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x4112
Byte Offset: 0x10448
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH1_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that:

1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x4113
Byte Offset: 0x1044c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH1_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that:

1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use.

If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x4114
Byte Offset: 0x10450
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH1_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that:

1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x4115
Byte Offset: 0x10454
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH1_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that:

1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x4116
Byte Offset: 0x10458
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH1_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) - SKIP_Y_LINES

Offset: 0x4117
Byte Offset: 0x1045c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH1_OUT_Y_0

Offset: 0x4118
Byte Offset: 0x10460
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH1_NOTIFY_MASK_0

Offset: 0x4119
Byte Offset: 0x10464
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH1_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED.

If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification.

To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x411a
Byte Offset: 0x10468
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels

Bit	Reset	Description
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH1_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel Status of current or last frame seen.

Offset: 0x411b
Byte Offset: 0x1046c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH1_FRAME_COUNT_0

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame.

If in frame then FS=FE+1. FE can be perceived as a checksum check for FS.

Write any value to register to clear all counts.

Offset: 0x411c
Byte Offset: 0x10470

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH1_LINE_TIMER_FIRST_0

Offset: 0x411d
Byte Offset: 0x10474
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH1_FLUSH_FIRST_0

Offset: 0x411e
Byte Offset: 0x10478
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH1_DOL_CTRL_0

Offset: 0x411f
Byte Offset: 0x1047c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH1_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x4120
Byte Offset: 0x10480
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

Bit	R/W	Reset	Description
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH1_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x4121
Byte Offset: 0x10484
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH1_PIXFMT_FORMAT_0

Offset: 0x4122
Byte Offset: 0x10488
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/ RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH1_PIXFMT_WIDE_0

Offset: 0x4123
Byte Offset: 0x1048c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH1_PIXFMT_PDAF_CROP_X_0

Offset: 0x4124
Byte Offset: 0x10490
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH1_PIXFMT_PDAF_CROP_Y_0

Offset: 0x4125
Byte Offset: 0x10494
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH1_PIXFMT_PDAF_CONFIG0_0

Offset: 0x4126
Byte Offset: 0x10498
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH1_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x4127
Byte Offset: 0x1049c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH1_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x4128
Byte Offset: 0x104a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH1_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.

Offset: 0x4129
Byte Offset: 0x104a4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH1_PIXFMT_PDAF_CONFIG1_0

Offset: 0x412a
Byte Offset: 0x104a8
Read/Write: R/W
Parity Protection: Y

Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH1_DPCM_STRIP_0

Offset: 0x412f
Byte Offset: 0x104bc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH1_DPCM_CHUNK_FIRST_0

Offset: 0x4130
Byte Offset: 0x104c0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH1_DPCM_CHUNK_BODY_0

Offset: 0x4131
Byte Offset: 0x104c4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH1_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x4132
Byte Offset: 0x104c8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH1_DPCM_CHUNK_LAST_0

Offset: 0x4133
Byte Offset: 0x104cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH1_DPCM_STATISTICS_0_0

Offset: 0x4134
Byte Offset: 0x104d0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH1_DPCM_STATISTICS_1_0

Offset: 0x4135
Byte Offset: 0x104d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH1_DPCM_MODE_0

Offset: 0x4136
Byte Offset: 0x104d8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH1_DPCM_CLAMP_HIGH_0

Offset: 0x4137
Byte Offset: 0x104dc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH1_DPCM_CLAMP_LOW_0

Offset: 0x4138
Byte Offset: 0x104e0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH1_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x413b
Byte Offset: 0x104ec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH1_ATOMP_SURFACE_OFFSET0_0

Offset: 0x413c
Byte Offset: 0x104f0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH1_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x413d
Byte Offset: 0x104f4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH1_ATOMP_SURFACE_STRIDE0_0

Offset: 0x413e
Byte Offset: 0x104f8
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH1_ATOMP_DPCM_CHUNK_0

Offset: 0x413f
Byte Offset: 0x104fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH1_ATOMP_SURFACE_OFFSET1_0

Offset: 0x4140
Byte Offset: 0x10500
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH1_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x4141
Byte Offset: 0x10504
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH1_ATOMP_SURFACE_STRIDE1_0

Offset: 0x4142
Byte Offset: 0x10508
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH1_ATOMP_SURFACE_OFFSET2_0

Offset: 0x4143
Byte Offset: 0x1050c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH1_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x4144
Byte Offset: 0x10510
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH1_ATOMP_SURFACE_STRIDE2_0

Offset: 0x4145
Byte Offset: 0x10514
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH1_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x4146
Byte Offset: 0x10518
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH1_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x4147
Byte Offset: 0x1051c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH1_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x4148
Byte Offset: 0x10520
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH1_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x4149
Byte Offset: 0x10524
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH1_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x414a
Byte Offset: 0x10528
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH1_ATOMP_HIGH_PRI_REQ_0

Offset: 0x414b
Byte Offset: 0x1052c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH1_ATOMP_RESERVE_0

Offset: 0x414c
Byte Offset: 0x10530
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 2 Specific Registers

VI_CH2_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x4200
Byte Offset: 0x10800
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH2_CHANSEL_0

Offset: 0x4204
Byte Offset: 0x10810
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH2_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x4205
Byte Offset: 0x10814
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH2_CONTROL_0

Channel prefix gets added to register name (VI_CH*)

Offset: 0x4206
Byte Offset: 0x10818
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH2_MATCH_0

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x4207
Byte Offset: 0x1081c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH2_MATCH_DOL_0

Offset: 0x4208
Byte Offset: 0x10820
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH2_MATCH_VC_HI_0

Offset: 0x4209
Byte Offset: 0x10824
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH2_MATCH_DATATYPE_0

Offset: 0x420a
Byte Offset: 0x10828
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH2_MATCH_FRAMEID_0

Offset: 0x420b
Byte Offset: 0x1082c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH2_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT. The one exception is embedded data type which does not get remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on.

You are also likely not to have a wildcard but an exact match for data type. It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x420c
Byte Offset: 0x10830
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH2_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x420d
Byte Offset: 0x10834
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH2_FRAME_Y_0

Expected frame y dimension

Offset: 0x420e
Byte Offset: 0x10838
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH2_EMBED_X_0
Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x420f
Byte Offset: 0x1083c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH2_EMBED_Y_0

Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x4210
Byte Offset: 0x10840
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH2_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed. Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x4211
Byte Offset: 0x10844
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH2_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started. For example starting a flash in anticipation of an upcoming frame, or setup up a focusing process. This is done setting a tripline count which flags a notification once the number of lines has been reached. The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. The line timer events are not filtered by the cropped region, they are solely based on the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x4212
Byte Offset: 0x10848
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH2_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x4213
Byte Offset: 0x1084c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH2_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use. If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x4214
Byte Offset: 0x10850
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH2_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x4215
Byte Offset: 0x10854
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH2_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop. 0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x4216
Byte Offset: 0x10858
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH2_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) - SKIP_Y_LINES

Offset: 0x4217
Byte Offset: 0x1085c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH2_OUT_Y_0

Offset: 0x4218
Byte Offset: 0x10860
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH2_NOTIFY_MASK_0

Offset: 0x4219
Byte Offset: 0x10864
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH2_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED.

If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification.

To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x421a
Byte Offset: 0x10868
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels

Bit	Reset	Description
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH2_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x421b
Byte Offset: 0x1086c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH2_FRAME_COUNT_0

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame.

If in frame then FS=FE+1. FE can be perceived as a checksum check for FS. Write any value to register to clear all counts.

Offset: 0x421c
Byte Offset: 0x10870
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH2_LINE_TIMER_FIRST_0

Offset: 0x421d
Byte Offset: 0x10874
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH2_FLUSH_FIRST_0

Offset: 0x421e
Byte Offset: 0x10878
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH2_DOL_CTRL_0

Offset: 0x421f
Byte Offset: 0x1087c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH2_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x4220
Byte Offset: 0x10880
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;

Bit	R/W	Reset	Description
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH2_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x4221
Byte Offset: 0x10884
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH2_PIXFMT_FORMAT_0

Offset: 0x4222
Byte Offset: 0x10888
Read/Write: R/W
Parity Protection: Y

Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/ RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH2_PIXFMT_WIDE_0

Offset: 0x4223
Byte Offset: 0x1088c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH2_PIXFMT_PDAF_CROP_X_0

Offset: 0x4224
Byte Offset: 0x10890
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH2_PIXFMT_PDAF_CROP_Y_0

Offset: 0x4225
Byte Offset: 0x10894
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH2_PIXFMT_PDAF_CONFIG0_0

Offset: 0x4226
Byte Offset: 0x10898
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH2_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x4227
Byte Offset: 0x1089c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends

Bit	Reset	Description
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH2_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x4228
Byte Offset: 0x108a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH2_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.

Offset: 0x4229
Byte Offset: 0x108a4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH2_PIXFMT_PDAF_CONFIG1_0

Offset: 0x422a
Byte Offset: 0x108a8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH2_DPCM_STRIP_0

Offset: 0x422f
Byte Offset: 0x108bc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH2_DPCM_CHUNK_FIRST_0

Offset: 0x4230
Byte Offset: 0x108c0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH2_DPCM_CHUNK_BODY_0

Offset: 0x4231
Byte Offset: 0x108c4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH2_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x4232
Byte Offset: 0x108c8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH2_DPCM_CHUNK_LAST_0

Offset: 0x4233
Byte Offset: 0x108cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH2_DPCM_STATISTICS_0_0

Offset: 0x4234
Byte Offset: 0x108d0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH2_DPCM_STATISTICS_1_0

Offset: 0x4235
Byte Offset: 0x108d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH2_DPCM_MODE_0

Offset: 0x4236
Byte Offset: 0x108d8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH2_DPCM_CLAMP_HIGH_0

Offset: 0x4237
Byte Offset: 0x108dc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH2_DPCM_CLAMP_LOW_0

Offset: 0x4238
Byte Offset: 0x108e0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH2_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x423b
Byte Offset: 0x108ec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH2_ATOMP_SURFACE_OFFSET0_0

Offset: 0x423c
Byte Offset: 0x108f0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH2_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x423d
Byte Offset: 0x108f4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH2_ATOMP_SURFACE_STRIDE0_0

Offset: 0x423e
Byte Offset: 0x108f8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH2_ATOMP_DPCM_CHUNK_0

Offset: 0x423f
Byte Offset: 0x108fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH2_ATOMP_SURFACE_OFFSET1_0

Offset: 0x4240
Byte Offset: 0x10900
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH2_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x4241
Byte Offset: 0x10904
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH2_ATOMP_SURFACE_STRIDE1_0

Offset: 0x4242
Byte Offset: 0x10908
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH2_ATOMP_SURFACE_OFFSET2_0

Offset: 0x4243
Byte Offset: 0x1090c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH2_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x4244
Byte Offset: 0x10910
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH2_ATOMP_SURFACE_STRIDE2_0

Offset: 0x4245
Byte Offset: 0x10914
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH2_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x4246
Byte Offset: 0x10918
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH2_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x4247
Byte Offset: 0x1091c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH2_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x4248
Byte Offset: 0x10920
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH2_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x4249
Byte Offset: 0x10924
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH2_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x424a
Byte Offset: 0x10928
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH2_ATOMP_HIGH_PRI_REQ_0

Offset: 0x424b
Byte Offset: 0x1092c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH2_ATOMP_RESERVE_0

Offset: 0x424c
Byte Offset: 0x10930
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 3 Specific Registers

VI_CH3_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x4300
Byte Offset: 0x10c00
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH3_CHANSEL_0

Offset: 0x4304
Byte Offset: 0x10c10
Read/Write: R/W

Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH3_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race. condition.

Offset: 0x4305
Byte Offset: 0x10c14
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH3_CONTROL_0

Channel prefix gets added to register name (VI_CH*)

Offset: 0x4306
Byte Offset: 0x10c18
Read/Write: R/W
Parity Protection: See table below
Shadow: N

SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH3_MATCH_0

SHADOWED CHANSEL REGISTERS

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x4307
Byte Offset: 0x10c1c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH3_MATCH_DOL_0

Offset: 0x4308
Byte Offset: 0x10c20
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH3_MATCH_VC_HI_0

Offset: 0x4309
Byte Offset: 0x10c24
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH3_MATCH_DATATYPE_0

Offset: 0x430a
Byte Offset: 0x10c28
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH3_MATCH_FRAMEID_0

Offset: 0x430b
Byte Offset: 0x10c2c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH3_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT.

The one exception is embedded data type which does not get remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type. Note that if a channel uses data type interleaving it is most likely not to have this feature on. You are also likely not to have a wildcard but an exact match for data type. It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x430c
Byte Offset: 0x10c30
Read/Write: R/W
Parity Protection: Y

Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH3_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x430d
Byte Offset: 0x10c34
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH3_FRAME_Y_0

Expected frame y dimension

Offset: 0x430e
Byte Offset: 0x10c38
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH3_EMBED_X_0

Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x430f
Byte Offset: 0x10c3c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH3_EMBED_Y_0

Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x4310
Byte Offset: 0x10c40
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH3_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x4311
Byte Offset: 0x10c44
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH3_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started. For example starting a flash in anticipation of an upcoming frame, or setup up a focusing process.This is done setting a tripline count which flags a notification once the number of lines has been reached. The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.The line timer events are not filtered by the cropped region, they are solely based on the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x4312
Byte Offset: 0x10c48
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH3_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x4313
Byte Offset: 0x10c4c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH3_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use.

If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x4314
Byte Offset: 0x10c50
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH3_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x4315
Byte Offset: 0x10c54
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH3_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x4316
Byte Offset: 0x10c58
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH3_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) - SKIP_Y_LINES

Offset: 0x4317
Byte Offset: 0x10c5c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH3_OUT_Y_0

Offset: 0x4318
Byte Offset: 0x10c60
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH3_NOTIFY_MASK_0

Offset: 0x4319
Byte Offset: 0x10c64
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped

Bit	Reset	Description
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH3_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED.The expected use of NOMATCH is for debug purposes.

If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification. To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x431a
Byte Offset: 0x10c68
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)

Bit	Reset	Description
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH3_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x431b
Byte Offset: 0x10c6c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH3_FRAME_COUNT_0

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame.

If in frame then FS=FE+1. FE can be perceived as a checksum check for FS.Write any value to register to clear all counts.

Offset: 0x431c
Byte Offset: 0x10c70
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH3_LINE_TIMER_FIRST_0

Offset: 0x431d
Byte Offset: 0x10c74
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH3_FLUSH_FIRST_0

Offset: 0x431e
Byte Offset: 0x10c78
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH3_DOL_CTRL_0

Offset: 0x431f
Byte Offset: 0x10c7c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH3_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x4320
Byte Offset: 0x10c80
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;

Bit	R/W	Reset	Description
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH3_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x4321
Byte Offset: 0x10c84
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH3_PIXFMT_FORMAT_0

Offset: 0x4322
Byte Offset: 0x10c88
Read/Write: R/W
Parity Protection: Y

Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH3_PIXFMT_WIDE_0

Offset: 0x4323
Byte Offset: 0x10c8c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH3_PIXFMT_PDAF_CROP_X_0

Offset: 0x4324
Byte Offset: 0x10c90
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH3_PIXFMT_PDAF_CROP_Y_0

Offset: 0x4325
Byte Offset: 0x10c94
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH3_PIXFMT_PDAF_CONFIG0_0

Offset: 0x4326
Byte Offset: 0x10c98
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH3_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x4327
Byte Offset: 0x10c9c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends

Bit	Reset	Description
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH3_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x4328
Byte Offset: 0x10ca0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH3_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.

Offset: 0x4329
Byte Offset: 0x10ca4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH3_PIXFMT_PDAF_CONFIG1_0

Offset: 0x432a
Byte Offset: 0x10ca8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH3_DPCM_STRIP_0

Offset: 0x432f
Byte Offset: 0x10cbc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH3_DPCM_CHUNK_FIRST_0

Offset: 0x4330
Byte Offset: 0x10cc0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH3_DPCM_CHUNK_BODY_0

Offset: 0x4331
Byte Offset: 0x10cc4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH3_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x4332
Byte Offset: 0x10cc8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH3_DPCM_CHUNK_LAST_0

Offset: 0x4333
Byte Offset: 0x10ccc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH3_DPCM_STATISTICS_0_0

Offset: 0x4334
Byte Offset: 0x10cd0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH3_DPCM_STATISTICS_1_0

Offset: 0x4335
Byte Offset: 0x10cd4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH3_DPCM_MODE_0

Offset: 0x4336
Byte Offset: 0x10cd8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH3_DPCM_CLAMP_HIGH_0

Offset: 0x4337
Byte Offset: 0x10cdc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH3_DPCM_CLAMP_LOW_0

Offset: 0x4338
Byte Offset: 0x10ce0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH3_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x433b
Byte Offset: 0x10cec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH3_ATOMP_SURFACE_OFFSET0_0

Offset: 0x433c
Byte Offset: 0x10cf0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH3_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x433d
Byte Offset: 0x10cf4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH3_ATOMP_SURFACE_STRIDE0_0

Offset: 0x433e
Byte Offset: 0x10cf8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH3_ATOMP_DPCM_CHUNK_0

Offset: 0x433f
Byte Offset: 0x10cfc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH3_ATOMP_SURFACE_OFFSET1_0

Offset: 0x4340
Byte Offset: 0x10d00
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH3_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x4341
Byte Offset: 0x10d04
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH3_ATOMP_SURFACE_STRIDE1_0

Offset: 0x4342
Byte Offset: 0x10d08
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH3_ATOMP_SURFACE_OFFSET2_0

Offset: 0x4343
Byte Offset: 0x10d0c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH3_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x4344
Byte Offset: 0x10d10
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH3_ATOMP_SURFACE_STRIDE2_0

Offset: 0x4345
Byte Offset: 0x10d14
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH3_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x4346
Byte Offset: 0x10d18
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH3_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x4347

Byte Offset: 0x10d1c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH3_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x4348
Byte Offset: 0x10d20
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH3_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x4349
Byte Offset: 0x10d24
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH3_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x434a
Byte Offset: 0x10d28

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH3_ATOMP_HIGH_PRI_REQ_0

Offset: 0x434b
Byte Offset: 0x10d2c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH3_ATOMP_RESERVE_0

Offset: 0x434c
Byte Offset: 0x10d30
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 4 Specific Registers

VI_CH4_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x4400
Byte Offset: 0x11000
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH4_CHANSEL_0

Offset: 0x4404
Byte Offset: 0x11010
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH4_ENABLE_0
Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from

the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is

properly terminated. The two step process ensures no races in hardware.

put channel enable to a separate register field to avoid SW race condition.

Offset: 0x4405
Byte Offset: 0x11014
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH4_CONTROL_0
Channel prefix gets added to register name (VI_CH*)
Offset: 0x4406
Byte Offset: 0x11018
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed

Bit	Parity Protection	Reset	Description
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH4_MATCH_0

SHADOWED CHANSEL REGISTERS

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x4407
Byte Offset: 0x1101c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH4_MATCH_DOL_0

Offset: 0x4408
Byte Offset: 0x11020
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH4_MATCH_VC_HI_0

Offset: 0x4409
Byte Offset: 0x11024
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH4_MATCH_DATATYPE_0

Offset: 0x440a

Byte Offset: 0x11028
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH4_MATCH_FRAMEID_0

Offset: 0x440b
Byte Offset: 0x1102c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH4_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT.

The one exception is embedded data type which does not get remapped through this register.

If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on.

You are also likely not to have a wildcard but an exact match for data type.

It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}).

Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x440c
Byte Offset: 0x11030
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH4_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x440d
Byte Offset: 0x11034
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH4_FRAME_Y_0

Expected frame y dimension

Offset: 0x440e
Byte Offset: 0x11038
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH4_EMBED_X_0
Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x440f
Byte Offset: 0x1103c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH4_EMBED_Y_0
Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x4410
Byte Offset: 0x11040
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.

Bit	Reset	Description
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH4_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data.

This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached.

The count can be setup to occur periodically. This feature does not count embedded data, however any embedded

data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output

cropped region.

Offset: 0x4411
Byte Offset: 0x11044
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH4_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started.

For example starting a flash in anticipation of an upcoming frame, or setup up a focusing process.

This is done setting a tripline count which flags a notification once the number of lines has been reached.

The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

The line timer events are not filtered by the cropped region, they are solely based on the input frame.

The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x4412
Byte Offset: 0x11048
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH4_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x4413
Byte Offset: 0x1104c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH4_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use.

If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x4414
Byte Offset: 0x11050
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH4_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x4415
Byte Offset: 0x11054
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH4_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x4416
Byte Offset: 0x11058
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH4_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) - SKIP_Y_LINES

Offset: 0x4417
Byte Offset: 0x1105c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH4_OUT_Y_0

Offset: 0x4418
Byte Offset: 0x11060
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH4_NOTIFY_MASK_0

Offset: 0x4419
Byte Offset: 0x11064
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register

Bit	Reset	Description
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH4_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED. The expected use of NOMATCH is for debug purposes.

If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification.

To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x441a
Byte Offset: 0x11068
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line

Bit	Reset	Description
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH4_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen.

IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x441b
Byte Offset: 0x1106c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status

Bit	Reset	Description
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH4_FRAME_COUNT_0

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame.

If in frame then FS=FE+1. FE can be perceived as a checksum check for FS.

Write any value to register to clear all counts.

Offset: 0x441c
Byte Offset: 0x11070
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH4_LINE_TIMER_FIRST_0

Offset: 0x441d
Byte Offset: 0x11074
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH4_FLUSH_FIRST_0

Offset: 0x441e
Byte Offset: 0x11078
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH4_DOL_CTRL_0

Offset: 0x441f
Byte Offset: 0x1107c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH4_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x4420
Byte Offset: 0x11080
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH4_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x4421
Byte Offset: 0x11084
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH4_PIXFMT_FORMAT_0

Offset: 0x4422
Byte Offset: 0x11088
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/ RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0

Bit	Reset	Description
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH4_PIXFMT_WIDE_0

Offset: 0x4423
Byte Offset: 0x1108c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH4_PIXFMT_PDAF_CROP_X_0

Offset: 0x4424
Byte Offset: 0x11090
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH4_PIXFMT_PDAF_CROP_Y_0

Offset: 0x4425
Byte Offset: 0x11094
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH4_PIXFMT_PDAF_CONFIG0_0

Offset: 0x4426
Byte Offset: 0x11098
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH4_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x4427
Byte Offset: 0x1109c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH4_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x4428
Byte Offset: 0x110a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH4_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.
Offset: 0x4429
Byte Offset: 0x110a4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH4_PIXFMT_PDAF_CONFIG1_0

Offset: 0x442a
Byte Offset: 0x110a8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH4_DPCM_STRIP_0

Offset: 0x442f
Byte Offset: 0x110bc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH4_DPCM_CHUNK_FIRST_0

Offset: 0x4430
Byte Offset: 0x110c0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH4_DPCM_CHUNK_BODY_0

Offset: 0x4431
Byte Offset: 0x110c4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH4_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x4432
Byte Offset: 0x110c8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH4_DPCM_CHUNK_LAST_0

Offset: 0x4433
Byte Offset: 0x110cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH4_DPCM_STATISTICS_0_0

Offset: 0x4434
Byte Offset: 0x110d0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH4_DPCM_STATISTICS_1_0

Offset: 0x4435
Byte Offset: 0x110d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH4_DPCM_MODE_0

Offset: 0x4436
Byte Offset: 0x110d8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH4_DPCM_CLAMP_HIGH_0

Offset: 0x4437
Byte Offset: 0x110dc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH4_DPCM_CLAMP_LOW_0

Offset: 0x4438
Byte Offset: 0x110e0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH4_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x443b
Byte Offset: 0x110ec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH4_ATOMP_SURFACE_OFFSET0_0

Offset: 0x443c
Byte Offset: 0x110f0
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH4_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x443d
Byte Offset: 0x110f4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH4_ATOMP_SURFACE_STRIDE0_0

Offset: 0x443e
Byte Offset: 0x110f8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH4_ATOMP_DPCM_CHUNK_0

Offset: 0x443f
Byte Offset: 0x110fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH4_ATOMP_SURFACE_OFFSET1_0

Offset: 0x4440
Byte Offset: 0x11100
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH4_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x4441
Byte Offset: 0x11104
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH4_ATOMP_SURFACE_STRIDE1_0

Offset: 0x4442
Byte Offset: 0x11108
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH4_ATOMP_SURFACE_OFFSET2_0

Offset: 0x4443
Byte Offset: 0x1110c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH4_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x4444
Byte Offset: 0x11110
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH4_ATOMP_SURFACE_STRIDE2_0

Offset: 0x4445
Byte Offset: 0x11114
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH4_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x4446
Byte Offset: 0x11118
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH4_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x4447
Byte Offset: 0x1111c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH4_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x4448
Byte Offset: 0x11120
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH4_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x4449
Byte Offset: 0x11124
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH4_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x444a
Byte Offset: 0x11128
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH4_ATOMP_HIGH_PRI_REQ_0

Offset: 0x444b
Byte Offset: 0x1112c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH4_ATOMP_RESERVE_0

Offset: 0x444c
Byte Offset: 0x11130
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 5 Specific Registers

VI_CH5_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS(non-shadowed)

Offset: 0x4500
Byte Offset: 0x11400
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)

Bit	Parity Protection	Reset	Description
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH5_CHANSEL_0

Offset: 0x4504
Byte Offset: 0x11410
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH5_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from

the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is

properly terminated. The two step process ensures no races in hardware.

put channel enable to a separate register field to avoid SW race condition.

Offset: 0x4505
Byte Offset: 0x11414
Read/Write: R/W

Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH5_CONTROL_0

Channel prefix gets added to register name (VI_CH*)
Offset: 0x4506
Byte Offset: 0x11418
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.

Bit	Parity Protection	Reset	Description
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH5_MATCH_0

SHADOWED CHANSEL REGISTERS

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x4507
Byte Offset: 0x1141c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH5_MATCH_DOL_0

Offset: 0x4508
Byte Offset: 0x11420
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH5_MATCH_VC_HI_0
Offset: 0x4509
Byte Offset: 0x11424
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH5_MATCH_DATATYPE_0
Offset: 0x450a
Byte Offset: 0x11428
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH5_MATCH_FRAMEID_0
Offset: 0x450b
Byte Offset: 0x1142c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH5_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT.

The one exception is embedded data type which does not get remapped through this register.

If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on.

You are also likely not to have a wildcard but an exact match for data type.

It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}).

Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x450c
Byte Offset: 0x11430
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH5_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x450d
Byte Offset: 0x11434
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH5_FRAME_Y_0

Expected frame y dimension

Offset: 0x450e
Byte Offset: 0x11438
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH5_EMBED_X_0

Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x450f
Byte Offset: 0x1143c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH5_EMBED_Y_0
Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x4510
Byte Offset: 0x11440
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH5_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data.

This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x4511
Byte Offset: 0x11444
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH5_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started.

For example starting a flash in anticipation of an upcoming frame, or setup up a focusing process.

This is done setting a tripline count which flags a notification once the number of lines has been reached.

The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

The line timer events are not filtered by the cropped region, they are solely based on the input frame.

The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x4512
Byte Offset: 0x11448
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH5_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x4513
Byte Offset: 0x1144c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH5_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use.

If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x4514
Byte Offset: 0x11450
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH5_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x4515
Byte Offset: 0x11454
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH5_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x4516
Byte Offset: 0x11458
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH5_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) - SKIP_Y_LINES

Offset: 0x4517
Byte Offset: 0x1145c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH5_OUT_Y_0

Offset: 0x4518
Byte Offset: 0x11460
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH5_NOTIFY_MASK_0

Offset: 0x4519
Byte Offset: 0x11464
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events

Bit	Reset	Description
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH5_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED.

The expected use of NOMATCH is for debug purposes.

If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification. To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x451a
Byte Offset: 0x11468
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH5_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x451b
Byte Offset: 0x1146c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH5_FRAME_COUNT_0

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame.

If in frame then FS=FE+1. FE can be perceived as a checksum check for FS.

Write any value to register to clear all counts.

Offset: 0x451c
Byte Offset: 0x11470
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095

Bit	Reset	Description
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH5_LINE_TIMER_FIRST_0

Offset: 0x451d
Byte Offset: 0x11474
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH5_FLUSH_FIRST_0

Offset: 0x451e
Byte Offset: 0x11478
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH5_DOL_CTRL_0

Offset: 0x451f
Byte Offset: 0x1147c
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH5_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x4520
Byte Offset: 0x11480
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH5_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x4521
Byte Offset: 0x11484
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH5_PIXFMT_FORMAT_0

Offset: 0x4522
Byte Offset: 0x11488
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/ RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0

Bit	Reset	Description
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH5_PIXFMT_WIDE_0

Offset: 0x4523
Byte Offset: 0x1148c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH5_PIXFMT_PDAF_CROP_X_0

Offset: 0x4524
Byte Offset: 0x11490
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH5_PIXFMT_PDAF_CROP_Y_0

Offset: 0x4525
Byte Offset: 0x11494
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH5_PIXFMT_PDAF_CONFIG0_0

Offset: 0x4526
Byte Offset: 0x11498
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH5_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x4527
Byte Offset: 0x1149c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH5_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x4528
Byte Offset: 0x114a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH5_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.
Offset: 0x4529
Byte Offset: 0x114a4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH5_PIXFMT_PDAF_CONFIG1_0

Offset: 0x452a
Byte Offset: 0x114a8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH5_DPCM_STRIP_0

Offset: 0x452f
Byte Offset: 0x114bc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH5_DPCM_CHUNK_FIRST_0

Offset: 0x4530
Byte Offset: 0x114c0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH5_DPCM_CHUNK_BODY_0

Offset: 0x4531
Byte Offset: 0x114c4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH5_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x4532
Byte Offset: 0x114c8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH5_DPCM_CHUNK_LAST_0

Offset: 0x4533
Byte Offset: 0x114cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH5_DPCM_STATISTICS_0_0

Offset: 0x4534
Byte Offset: 0x114d0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH5_DPCM_STATISTICS_1_0

Offset: 0x4535
Byte Offset: 0x114d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH5_DPCM_MODE_0

Offset: 0x4536
Byte Offset: 0x114d8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH5_DPCM_CLAMP_HIGH_0

Offset: 0x4537
Byte Offset: 0x114dc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH5_DPCM_CLAMP_LOW_0

Offset: 0x4538
Byte Offset: 0x114e0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH5_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x453b
Byte Offset: 0x114ec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH5_ATOMP_SURFACE_OFFSET0_0

Offset: 0x453c
Byte Offset: 0x114f0
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH5_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x453d
Byte Offset: 0x114f4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH5_ATOMP_SURFACE_STRIDE0_0

Offset: 0x453e
Byte Offset: 0x114f8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH5_ATOMP_DPCM_CHUNK_0

Offset: 0x453f
Byte Offset: 0x114fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH5_ATOMP_SURFACE_OFFSET1_0

Offset: 0x4540
Byte Offset: 0x11500
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH5_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x4541
Byte Offset: 0x11504
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH5_ATOMP_SURFACE_STRIDE1_0

Offset: 0x4542
Byte Offset: 0x11508
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH5_ATOMP_SURFACE_OFFSET2_0

Offset: 0x4543
Byte Offset: 0x1150c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH5_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x4544
Byte Offset: 0x11510
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH5_ATOMP_SURFACE_STRIDE2_0

Offset: 0x4545
Byte Offset: 0x11514
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH5_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x4546
Byte Offset: 0x11518
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH5_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x4547
Byte Offset: 0x1151c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH5_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x4548
Byte Offset: 0x11520
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH5_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x4549
Byte Offset: 0x11524
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH5_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x454a
Byte Offset: 0x11528
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH5_ATOMP_HIGH_PRI_REQ_0

Offset: 0x454b
Byte Offset: 0x1152c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH5_ATOMP_RESERVE_0

Offset: 0x454c
Byte Offset: 0x11530
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 6 Specific Registers

VI_CH6_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x4600
Byte Offset: 0x11800
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.

Bit	Parity Protection	Reset	Description
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH6_CHANSEL_0

Offset: 0x4604
Byte Offset: 0x11810
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH6_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x4605
Byte Offset: 0x11814
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH6_CONTROL_0

Channel prefix gets added to register name (VI_CH*)
Offset: 0x4606
Byte Offset: 0x11818
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)

Bit	Parity Protection	Reset	Description
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH6_MATCH_0

SHADOWED CHANSEL REGISTERS

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x4607
Byte Offset: 0x1181c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH6_MATCH_DOL_0

Offset: 0x4608
Byte Offset: 0x11820
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**

Bit	Reset	Description
13:0	0x0	**DOL_MASK**

VI_CH6_MATCH_VC_HI_0
Offset: 0x4609
Byte Offset: 0x11824
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH6_MATCH_DATATYPE_0
Offset: 0x460a
Byte Offset: 0x11828
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH6_MATCH_FRAMEID_0
Offset: 0x460b
Byte Offset: 0x1182c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH6_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT. The one exception is embedded data type which does not get remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on.

You are also likely not to have a wildcard but an exact match for data type. It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x460c
Byte Offset: 0x11830
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH6_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x460d
Byte Offset: 0x11834
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH6_FRAME_Y_0

Expected frame y dimension

Offset: 0x460e
Byte Offset: 0x11838
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH6_EMBED_X_0
Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x460f
Byte Offset: 0x1183c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH6_EMBED_Y_0
Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x4610
Byte Offset: 0x11840
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH6_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x4611
Byte Offset: 0x11844
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH6_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started. For example starting a flash in anticipation of an upcoming frame, or setup up a focusing process. This is done setting a tripline count which flags a notification once the number of lines has been reached. The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

The line timer events are not filtered by the cropped region, they are solely based on the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x4612
Byte Offset: 0x11848
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH6_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x4613
Byte Offset: 0x1184c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH6_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use. If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x4614
Byte Offset: 0x11850
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH6_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x4615
Byte Offset: 0x11854
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH6_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x4616
Byte Offset: 0x11858
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH6_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) - SKIP_Y_LINES

Offset: 0x4617
Byte Offset: 0x1185c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH6_OUT_Y_0

Offset: 0x4618
Byte Offset: 0x11860
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH6_NOTIFY_MASK_0

Offset: 0x4619
Byte Offset: 0x11864
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register

Bit	Reset	Description
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH6_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED.

The expected use of NOMATCH is for debug purposes. If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification.

To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x461a
Byte Offset: 0x11868
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line

Bit	Reset	Description
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH6_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x461b
Byte Offset: 0x1186c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame

Bit	Reset	Description
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH6_FRAME_COUNT_0

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame.

If in frame then FS=FE+1. FE can be perceived as a checksum check for FS.

Write any value to register to clear all counts.

Offset: 0x461c
Byte Offset: 0x11870
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH6_LINE_TIMER_FIRST_0

Offset: 0x461d
Byte Offset: 0x11874
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH6_FLUSH_FIRST_0

Offset: 0x461e
Byte Offset: 0x11878
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH6_DOL_CTRL_0

Offset: 0x461f
Byte Offset: 0x1187c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH6_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x4620
Byte Offset: 0x11880

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH6_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x4621
Byte Offset: 0x11884
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH6_PIXFMT_FORMAT_0

Offset: 0x4622
Byte Offset: 0x11888
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0

Bit	Reset	Description
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH6_PIXFMT_WIDE_0

Offset: 0x4623
Byte Offset: 0x1188c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH6_PIXFMT_PDAF_CROP_X_0

Offset: 0x4624
Byte Offset: 0x11890
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH6_PIXFMT_PDAF_CROP_Y_0

Offset: 0x4625
Byte Offset: 0x11894
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH6_PIXFMT_PDAF_CONFIG0_0

Offset: 0x4626
Byte Offset: 0x11898
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH6_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x4627
Byte Offset: 0x1189c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH6_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x4628
Byte Offset: 0x118a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH6_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.
Offset: 0x4629
Byte Offset: 0x118a4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH6_PIXFMT_PDAF_CONFIG1_0

Offset: 0x462a
Byte Offset: 0x118a8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH6_DPCM_STRIP_0

Offset: 0x462f
Byte Offset: 0x118bc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH6_DPCM_CHUNK_FIRST_0

Offset: 0x4630
Byte Offset: 0x118c0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH6_DPCM_CHUNK_BODY_0

Offset: 0x4631
Byte Offset: 0x118c4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH6_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x4632
Byte Offset: 0x118c8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH6_DPCM_CHUNK_LAST_0

Offset: 0x4633
Byte Offset: 0x118cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH6_DPCM_STATISTICS_0_0

Offset: 0x4634
Byte Offset: 0x118d0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH6_DPCM_STATISTICS_1_0

Offset: 0x4635
Byte Offset: 0x118d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH6_DPCM_MODE_0

Offset: 0x4636
Byte Offset: 0x118d8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH6_DPCM_CLAMP_HIGH_0

Offset: 0x4637
Byte Offset: 0x118dc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH6_DPCM_CLAMP_LOW_0

Offset: 0x4638
Byte Offset: 0x118e0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH6_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x463b
Byte Offset: 0x118ec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH6_ATOMP_SURFACE_OFFSET0_0

Offset: 0x463c
Byte Offset: 0x118f0
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH6_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x463d
Byte Offset: 0x118f4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH6_ATOMP_SURFACE_STRIDE0_0

Offset: 0x463e
Byte Offset: 0x118f8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH6_ATOMP_DPCM_CHUNK_0

Offset: 0x463f
Byte Offset: 0x118fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH6_ATOMP_SURFACE_OFFSET1_0

Offset: 0x4640
Byte Offset: 0x11900
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH6_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x4641
Byte Offset: 0x11904
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH6_ATOMP_SURFACE_STRIDE1_0

Offset: 0x4642
Byte Offset: 0x11908
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH6_ATOMP_SURFACE_OFFSET2_0

Offset: 0x4643
Byte Offset: 0x1190c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH6_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x4644
Byte Offset: 0x11910
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH6_ATOMP_SURFACE_STRIDE2_0

Offset: 0x4645
Byte Offset: 0x11914
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH6_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x4646
Byte Offset: 0x11918
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH6_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x4647
Byte Offset: 0x1191c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH6_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x4648
Byte Offset: 0x11920
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH6_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x4649
Byte Offset: 0x11924
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH6_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x464a
Byte Offset: 0x11928
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH6_ATOMP_HIGH_PRI_REQ_0

Offset: 0x464b
Byte Offset: 0x1192c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH6_ATOMP_RESERVE_0

Offset: 0x464c
Byte Offset: 0x11930
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 7 Specific Registers

VI_CH7_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x4700
Byte Offset: 0x11c00
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH7_CHANSEL_0

Offset: 0x4704
Byte Offset: 0x11c10
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH7_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x4705
Byte Offset: 0x11c14
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH7_CONTROL_0

Channel prefix gets added to register name (VI_CH*)

Offset: 0x4706
Byte Offset: 0x11c18
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.

Bit	Parity Protection	Reset	Description
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH7_MATCH_0

SHADOWED CHANSEL REGISTERS

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x4707
Byte Offset: 0x11c1c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH7_MATCH_DOL_0

Offset: 0x4708
Byte Offset: 0x11c20
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH7_MATCH_VC_HI_0

Offset: 0x4709
Byte Offset: 0x11c24
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH7_MATCH_DATATYPE_0

Offset: 0x470a
Byte Offset: 0x11c28
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH7_MATCH_FRAMEID_0

Offset: 0x470b
Byte Offset: 0x11c2c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH7_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT.

The one exception is embedded data type which does not get remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on.

You are also likely not to have a wildcard but an exact match for data type. It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x470c
Byte Offset: 0x11c30
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH7_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x470d
Byte Offset: 0x11c34
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH7_FRAME_Y_0

Expected frame y dimension

Offset: 0x470e
Byte Offset: 0x11c38
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH7_EMBED_X_0

Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x470f
Byte Offset: 0x11c3c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH7_EMBED_Y_0

Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x4710
Byte Offset: 0x11c40
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH7_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data,

however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x4711
Byte Offset: 0x11c44
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH7_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started. For example starting a flash in anticipation of an upcoming frame, or setup up a focusing process. This is done setting a tripline count which flags a notification once the number of lines has been reached. The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

The line timer events are not filtered by the cropped region, they are solely based on the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x4712
Byte Offset: 0x11c48
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH7_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x4713
Byte Offset: 0x11c4c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH7_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use. If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x4714
Byte Offset: 0x11c50
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH7_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x4715
Byte Offset: 0x11c54
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH7_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x4716
Byte Offset: 0x11c58
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH7_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) - SKIP_Y_LINES

Offset: 0x4717
Byte Offset: 0x11c5c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH7_OUT_Y_0

Offset: 0x4718
Byte Offset: 0x11c60
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH7_NOTIFY_MASK_0

Offset: 0x4719
Byte Offset: 0x11c64
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame

Bit	Reset	Description
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH7_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED.

The expected use of NOMATCH is for debug purposes.

If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification. To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x471a
Byte Offset: 0x11c68
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH7_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x471b
Byte Offset: 0x11c6c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH7_FRAME_COUNT_0

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame.

If in frame then FS=FE+1. FE can be perceived as a checksum check for FS.

Write any value to register to clear all counts.

Offset: 0x471c
Byte Offset: 0x11c70
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095

Bit	Reset	Description
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH7_LINE_TIMER_FIRST_0

Offset: 0x471d
Byte Offset: 0x11c74
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH7_FLUSH_FIRST_0

Offset: 0x471e
Byte Offset: 0x11c78
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH7_DOL_CTRL_0

Offset: 0x471f
Byte Offset: 0x11c7c

Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH7_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x4720
Byte Offset: 0x11c80
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH7_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x4721
Byte Offset: 0x11c84
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH7_PIXFMT_FORMAT_0

Offset: 0x4722
Byte Offset: 0x11c88
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/ RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0

Bit	Reset	Description
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH7_PIXFMT_WIDE_0

Offset: 0x4723
Byte Offset: 0x11c8c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH7_PIXFMT_PDAF_CROP_X_0

Offset: 0x4724
Byte Offset: 0x11c90
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH7_PIXFMT_PDAF_CROP_Y_0

Offset: 0x4725
Byte Offset: 0x11c94
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH7_PIXFMT_PDAF_CONFIG0_0

Offset: 0x4726
Byte Offset: 0x11c98
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH7_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x4727
Byte Offset: 0x11c9c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH7_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x4728
Byte Offset: 0x11ca0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH7_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.
Offset: 0x4729
Byte Offset: 0x11ca4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH7_PIXFMT_PDAF_CONFIG1_0

Offset: 0x472a
Byte Offset: 0x11ca8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH7_DPCM_STRIP_0

Offset: 0x472f
Byte Offset: 0x11cbc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH7_DPCM_CHUNK_FIRST_0

Offset: 0x4730
Byte Offset: 0x11cc0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH7_DPCM_CHUNK_BODY_0

Offset: 0x4731
Byte Offset: 0x11cc4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH7_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x4732
Byte Offset: 0x11cc8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH7_DPCM_CHUNK_LAST_0

Offset: 0x4733
Byte Offset: 0x11ccc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH7_DPCM_STATISTICS_0_0

Offset: 0x4734
Byte Offset: 0x11cd0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH7_DPCM_STATISTICS_1_0

Offset: 0x4735
Byte Offset: 0x11cd4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH7_DPCM_MODE_0

Offset: 0x4736
Byte Offset: 0x11cd8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH7_DPCM_CLAMP_HIGH_0

Offset: 0x4737
Byte Offset: 0x11cdc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH7_DPCM_CLAMP_LOW_0

Offset: 0x4738
Byte Offset: 0x11ce0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH7_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x473b
Byte Offset: 0x11cec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH7_ATOMP_SURFACE_OFFSET0_0

Offset: 0x473c
Byte Offset: 0x11cf0
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH7_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x473d
Byte Offset: 0x11cf4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH7_ATOMP_SURFACE_STRIDE0_0

Offset: 0x473e
Byte Offset: 0x11cf8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH7_ATOMP_DPCM_CHUNK_0

Offset: 0x473f
Byte Offset: 0x11cfc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH7_ATOMP_SURFACE_OFFSET1_0

Offset: 0x4740
Byte Offset: 0x11d00
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH7_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x4741
Byte Offset: 0x11d04
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH7_ATOMP_SURFACE_STRIDE1_0

Offset: 0x4742
Byte Offset: 0x11d08
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH7_ATOMP_SURFACE_OFFSET2_0

Offset: 0x4743
Byte Offset: 0x11d0c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH7_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x4744
Byte Offset: 0x11d10
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH7_ATOMP_SURFACE_STRIDE2_0

Offset: 0x4745
Byte Offset: 0x11d14
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH7_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x4746
Byte Offset: 0x11d18
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH7_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x4747
Byte Offset: 0x11d1c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH7_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x4748
Byte Offset: 0x11d20
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH7_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x4749
Byte Offset: 0x11d24
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH7_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x474a
Byte Offset: 0x11d28
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH7_ATOMP_HIGH_PRI_REQ_0

Offset: 0x474b
Byte Offset: 0x11d2c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH7_ATOMP_RESERVE_0

Offset: 0x474c
Byte Offset: 0x11d30
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 8 Specific Registers

VI_CH8_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x4800
Byte Offset: 0x12000
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.

Bit	Parity Protection	Reset	Description
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH8_CHANSEL_0

Offset: 0x4804
Byte Offset: 0x12010
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH8_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x4805
Byte Offset: 0x12014
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH8_CONTROL_0

Channel prefix gets added to register name (VI_CH*)
Offset: 0x4806
Byte Offset: 0x12018
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)

Bit	Parity Protection	Reset	Description
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH8_MATCH_0

SHADOWED CHANSEL REGISTERS

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x4807
Byte Offset: 0x1201c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH8_MATCH_DOL_0
Offset: 0x4808
Byte Offset: 0x12020
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**

Bit	Reset	Description
13:0	0x0	**DOL_MASK**

VI_CH8_MATCH_VC_HI_0
Offset: 0x4809
Byte Offset: 0x12024
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH8_MATCH_DATATYPE_0
Offset: 0x480a
Byte Offset: 0x12028
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH8_MATCH_FRAMEID_0
Offset: 0x480b
Byte Offset: 0x1202c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH8_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT. The one exception is embedded data type which does not get remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on.

You are also likely not to have a wildcard but an exact match for data type. It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x480c
Byte Offset: 0x12030
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH8_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x480d
Byte Offset: 0x12034
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH8_FRAME_Y_0

Expected frame y dimension

Offset: 0x480e
Byte Offset: 0x12038
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH8_EMBED_X_0
Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x480f
Byte Offset: 0x1203c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH8_EMBED_Y_0
Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x4810
Byte Offset: 0x12040
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH8_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x4811
Byte Offset: 0x12044
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH8_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started. For example starting a flash in anticipation of an upcoming frame, or setup up a focusing process.This is done setting a tripline count which flags a notification once the number of lines has been reached. The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. The line timer events are not filtered by the cropped region, they are solely based on the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x4812
Byte Offset: 0x12048
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH8_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x4813
Byte Offset: 0x1204c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH8_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use. If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x4814
Byte Offset: 0x12050
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH8_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x4815
Byte Offset: 0x12054
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH8_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x4816
Byte Offset: 0x12058
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH8_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) - SKIP_Y_LINES

Offset: 0x4817
Byte Offset: 0x1205c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH8_OUT_Y_0

Offset: 0x4818
Byte Offset: 0x12060
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH8_NOTIFY_MASK_0

Offset: 0x4819
Byte Offset: 0x12064
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register

Bit	Reset	Description
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH8_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED. The expected use of NOMATCH is for debug purposes.

If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification. To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x481a
Byte Offset: 0x12068
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line

Bit	Reset	Description
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH8_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x481b
Byte Offset: 0x1206c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)

Bit	Reset	Description
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH8_FRAME_COUNT_0

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame.

If in frame then FS=FE+1. FE can be perceived as a checksum check for FS. Write any value to register to clear all counts.

Offset: 0x481c
Byte Offset: 0x12070
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH8_LINE_TIMER_FIRST_0

Offset: 0x481d
Byte Offset: 0x12074
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH8_FLUSH_FIRST_0

Offset: 0x481e
Byte Offset: 0x12078
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH8_DOL_CTRL_0

Offset: 0x481f
Byte Offset: 0x1207c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH8_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x4820
Byte Offset: 0x12080
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH8_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x4821
Byte Offset: 0x12084
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH8_PIXFMT_FORMAT_0

Offset: 0x4822
Byte Offset: 0x12088
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0

Bit	Reset	Description
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH8_PIXFMT_WIDE_0

Offset: 0x4823
Byte Offset: 0x1208c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH8_PIXFMT_PDAF_CROP_X_0

Offset: 0x4824
Byte Offset: 0x12090
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH8_PIXFMT_PDAF_CROP_Y_0

Offset: 0x4825
Byte Offset: 0x12094
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends

Bit	Reset	Description
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH8_PIXFMT_PDAF_CONFIG0_0

Offset: 0x4826
Byte Offset: 0x12098
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH8_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x4827
Byte Offset: 0x1209c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH8_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x4828
Byte Offset: 0x120a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH8_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.

Offset: 0x4829
Byte Offset: 0x120a4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH8_PIXFMT_PDAF_CONFIG1_0

Offset: 0x482a
Byte Offset: 0x120a8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH8_DPCM_STRIP_0

Offset: 0x482f
Byte Offset: 0x120bc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH8_DPCM_CHUNK_FIRST_0

Offset: 0x4830
Byte Offset: 0x120c0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH8_DPCM_CHUNK_BODY_0

Offset: 0x4831
Byte Offset: 0x120c4

Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH8_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x4832
Byte Offset: 0x120c8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH8_DPCM_CHUNK_LAST_0

Offset: 0x4833
Byte Offset: 0x120cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH8_DPCM_STATISTICS_0_0

Offset: 0x4834
Byte Offset: 0x120d0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH8_DPCM_STATISTICS_1_0

Offset: 0x4835
Byte Offset: 0x120d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH8_DPCM_MODE_0

Offset: 0x4836
Byte Offset: 0x120d8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH8_DPCM_CLAMP_HIGH_0

Offset: 0x4837
Byte Offset: 0x120dc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH8_DPCM_CLAMP_LOW_0

Offset: 0x4838
Byte Offset: 0x120e0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH8_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x483b
Byte Offset: 0x120ec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH8_ATOMP_SURFACE_OFFSET0_0

Offset: 0x483c
Byte Offset: 0x120f0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH8_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x483d
Byte Offset: 0x120f4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH8_ATOMP_SURFACE_STRIDE0_0

Offset: 0x483e
Byte Offset: 0x120f8
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH8_ATOMP_DPCM_CHUNK_0

Offset: 0x483f
Byte Offset: 0x120fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH8_ATOMP_SURFACE_OFFSET1_0

Offset: 0x4840
Byte Offset: 0x12100
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH8_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x4841
Byte Offset: 0x12104
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH8_ATOMP_SURFACE_STRIDE1_0

Offset: 0x4842
Byte Offset: 0x12108
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH8_ATOMP_SURFACE_OFFSET2_0

Offset: 0x4843
Byte Offset: 0x1210c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH8_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x4844
Byte Offset: 0x12110
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH8_ATOMP_SURFACE_STRIDE2_0

Offset: 0x4845
Byte Offset: 0x12114
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH8_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x4846
Byte Offset: 0x12118
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH8_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x4847
Byte Offset: 0x1211c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH8_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x4848
Byte Offset: 0x12120
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH8_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x4849
Byte Offset: 0x12124
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH8_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x484a
Byte Offset: 0x12128
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH8_ATOMP_HIGH_PRI_REQ_0

Offset: 0x484b
Byte Offset: 0x1212c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH8_ATOMP_RESERVE_0

Offset: 0x484c
Byte Offset: 0x12130
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 9 Specific Registers

VI_CH9_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x4900
Byte Offset: 0x12400
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH9_CHANSEL_0

Offset: 0x4904
Byte Offset: 0x12410
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH9_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. put channel enable to a separate register field to avoid SW race condition.

Offset: 0x4905
Byte Offset: 0x12414
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH9_CONTROL_0

Channel prefix gets added to register name (VI_CH*)

Offset: 0x4906
Byte Offset: 0x12418
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH9_MATCH_0

SHADOWED CHANSEL REGISTERS

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x4907
Byte Offset: 0x1241c
Read/Write: R/W
Parity Protection: Y

Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH9_MATCH_DOL_0

Offset: 0x4908
Byte Offset: 0x12420
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH9_MATCH_VC_HI_0

Offset: 0x4909
Byte Offset: 0x12424
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH9_MATCH_DATATYPE_0

Offset: 0x490a
Byte Offset: 0x12428
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH9_MATCH_FRAMEID_0

Offset: 0x490b
Byte Offset: 0x1242c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH9_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT.

The one exception is embedded data type which does not get remapped through this register.If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on. You are also likely not to have a wildcard but an exact match for data type. It is illegal to program

OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x490c
Byte Offset: 0x12430
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH9_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x490d
Byte Offset: 0x12434
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH9_FRAME_Y_0

Expected frame y dimension

Offset: 0x490e
Byte Offset: 0x12438
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH9_EMBED_X_0

Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x490f
Byte Offset: 0x1243c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH9_EMBED_Y_0

Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x4910
Byte Offset: 0x12440
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.

Bit	Reset	Description
24	0x0	**EXPECT**: Embedded data expected within channel.
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH9_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x4911
Byte Offset: 0x12444
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH9_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started. For example starting a flash in anticipation of an upcoming frame, or setup up a

focusing process.This is done setting a tripline count which flags a notification once the number of lines has been reached. The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. The line timer events are not filtered by the cropped region, they are solely based on the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x4912
Byte Offset: 0x12448
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH9_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x4913
Byte Offset: 0x1244c
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH9_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use.

If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x4914
Byte Offset: 0x12450
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH9_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x4915
Byte Offset: 0x12454
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH9_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x4916
Byte Offset: 0x12458
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH9_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) - SKIP_Y_LINES

Offset: 0x4917
Byte Offset: 0x1245c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH9_OUT_Y_0

Offset: 0x4918
Byte Offset: 0x12460
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH9_NOTIFY_MASK_0

Offset: 0x4919
Byte Offset: 0x12464
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH9_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED.

The expected use of NOMATCH is for debug purposes. If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification. To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x491a
Byte Offset: 0x12468
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels

Bit	Reset	Description
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH9_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x491b
Byte Offset: 0x1246c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH9_FRAME_COUNT_0

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame.

If in frame then FS=FE+1. FE can be perceived as a checksum check for FS.

Write any value to register to clear all counts.

Offset: 0x491c
Byte Offset: 0x12470
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH9_LINE_TIMER_FIRST_0

Offset: 0x491d
Byte Offset: 0x12474
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH9_FLUSH_FIRST_0

Offset: 0x491e
Byte Offset: 0x12478
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH9_DOL_CTRL_0

Offset: 0x491f
Byte Offset: 0x1247c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH9_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x4920
Byte Offset: 0x12480
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

Bit	R/W	Reset	Description
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH9_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x4921
Byte Offset: 0x12484
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH9_PIXFMT_FORMAT_0

Offset: 0x4922
Byte Offset: 0x12488
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/ RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH9_PIXFMT_WIDE_0

Offset: 0x4923
Byte Offset: 0x1248c

Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH9_PIXFMT_PDAF_CROP_X_0

Offset: 0x4924
Byte Offset: 0x12490
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH9_PIXFMT_PDAF_CROP_Y_0

Offset: 0x4925
Byte Offset: 0x12494
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH9_PIXFMT_PDAF_CONFIG0_0

Offset: 0x4926
Byte Offset: 0x12498
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH9_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x4927
Byte Offset: 0x1249c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH9_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x4928

Byte Offset: 0x124a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH9_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.
Offset: 0x4929
Byte Offset: 0x124a4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH9_PIXFMT_PDAF_CONFIG1_0

Offset: 0x492a
Byte Offset: 0x124a8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH9_DPCM_STRIP_0

Offset: 0x492f
Byte Offset: 0x124bc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH9_DPCM_CHUNK_FIRST_0

Offset: 0x4930
Byte Offset: 0x124c0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH9_DPCM_CHUNK_BODY_0

Offset: 0x4931
Byte Offset: 0x124c4
Read/Write: R/W
Parity Protection: Y

Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH9_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x4932
Byte Offset: 0x124c8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH9_DPCM_CHUNK_LAST_0

Offset: 0x4933
Byte Offset: 0x124cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH9_DPCM_STATISTICS_0_0

Offset: 0x4934
Byte Offset: 0x124d0
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH9_DPCM_STATISTICS_1_0

Offset: 0x4935
Byte Offset: 0x124d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH9_DPCM_MODE_0

Offset: 0x4936
Byte Offset: 0x124d8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH9_DPCM_CLAMP_HIGH_0

Offset: 0x4937
Byte Offset: 0x124dc

Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH9_DPCM_CLAMP_LOW_0
Offset: 0x4938
Byte Offset: 0x124e0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH9_AXISTREAMID_CFG_0
NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x493b
Byte Offset: 0x124ec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH9_ATOMP_SURFACE_OFFSET0_0

Offset: 0x493c
Byte Offset: 0x124f0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH9_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x493d
Byte Offset: 0x124f4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH9_ATOMP_SURFACE_STRIDE0_0

Offset: 0x493e
Byte Offset: 0x124f8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH9_ATOMP_DPCM_CHUNK_0

Offset: 0x493f
Byte Offset: 0x124fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH9_ATOMP_SURFACE_OFFSET1_0

Offset: 0x4940
Byte Offset: 0x12500
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH9_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x4941
Byte Offset: 0x12504
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH9_ATOMP_SURFACE_STRIDE1_0

Offset: 0x4942
Byte Offset: 0x12508

Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH9_ATOMP_SURFACE_OFFSET2_0

Offset: 0x4943
Byte Offset: 0x1250c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH9_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x4944
Byte Offset: 0x12510
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH9_ATOMP_SURFACE_STRIDE2_0

Offset: 0x4945
Byte Offset: 0x12514
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH9_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x4946
Byte Offset: 0x12518
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH9_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x4947
Byte Offset: 0x1251c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH9_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x4948
Byte Offset: 0x12520
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH9_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x4949
Byte Offset: 0x12524
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH9_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x494a
Byte Offset: 0x12528
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH9_ATOMP_HIGH_PRI_REQ_0

Offset: 0x494b

Byte Offset: 0x1252c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH9_ATOMP_RESERVE_0

Offset: 0x494c
Byte Offset: 0x12530
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 10 Specific Registers

VI_CH10_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x4a00
Byte Offset: 0x12800
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)

Bit	Parity Protection	Reset	Description
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH10_CHANSEL_0

Offset: 0x4a04
Byte Offset: 0x12810
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH10_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x4a05
Byte Offset: 0x12814
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH10_CONTROL_0

Channel prefix gets added to register name (VI_CH*)
Offset: 0x4a06
Byte Offset: 0x12818
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.

Bit	Parity Protection	Reset	Description
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH10_MATCH_0

SHADOWED CHANSEL REGISTERS

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x4a07
Byte Offset: 0x1281c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH10_MATCH_DOL_0

Offset: 0x4a08
Byte Offset: 0x12820
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH10_MATCH_VC_HI_0
Offset: 0x4a09
Byte Offset: 0x12824
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH10_MATCH_DATATYPE_0
Offset: 0x4a0a
Byte Offset: 0x12828
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH10_MATCH_FRAMEID_0
Offset: 0x4a0b
Byte Offset: 0x1282c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH10_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT. The one exception is embedded data type which does not get remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on.

You are also likely not to have a wildcard but an exact match for data type. It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error.The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x4a0c
Byte Offset: 0x12830
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH10_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x4a0d
Byte Offset: 0x12834
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH10_FRAME_Y_0

Expected frame y dimension

Offset: 0x4a0e
Byte Offset: 0x12838
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH10_EMBED_X_0

Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x4a0f
Byte Offset: 0x1283c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH10_EMBED_Y_0

Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x4a10
Byte Offset: 0x12840
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH10_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x4a11
Byte Offset: 0x12844
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH10_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started. For example starting a flash in anticipation of an upcoming frame, or setup up a focusing process. This is done setting a tripline count which flags a notification once the number of lines has been reached. The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

The line timer events are not filtered by the cropped region, they are solely based on the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x4a12
Byte Offset: 0x12848
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH10_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x4a13
Byte Offset: 0x1284c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH10_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use. If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x4a14
Byte Offset: 0x12850
Read/Write: R/W
Parity Protection: Y

Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH10_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x4a15
Byte Offset: 0x12854
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH10_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x4a16
Byte Offset: 0x12858
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH10_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) - SKIP_Y_LINES

Offset: 0x4a17
Byte Offset: 0x1285c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH10_OUT_Y_0

Offset: 0x4a18
Byte Offset: 0x12860
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH10_NOTIFY_MASK_0

Offset: 0x4a19
Byte Offset: 0x12864
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register

Bit	Reset	Description
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH10_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED.The expected use of NOMATCH is for debug purposes. If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification. To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x4a1a
Byte Offset: 0x12868
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line

Bit	Reset	Description
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH10_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x4a1b
Byte Offset: 0x1286c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)

Bit	Reset	Description
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH10_FRAME_COUNT_0

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame.

If in frame then FS=FE+1. FE can be perceived as a checksum check for FS.

Write any value to register to clear all counts.

Offset: 0x4a1c
Byte Offset: 0x12870
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH10_LINE_TIMER_FIRST_0

Offset: 0x4a1d
Byte Offset: 0x12874
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH10_FLUSH_FIRST_0

Offset: 0x4a1e
Byte Offset: 0x12878
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH10_DOL_CTRL_0

Offset: 0x4a1f
Byte Offset: 0x1287c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH10_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x4a20
Byte Offset: 0x12880
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH10_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x4a21
Byte Offset: 0x12884
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH10_PIXFMT_FORMAT_0

Offset: 0x4a22
Byte Offset: 0x12888
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/ RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0

Bit	Reset	Description
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH10_PIXFMT_WIDE_0

Offset: 0x4a23
Byte Offset: 0x1288c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH10_PIXFMT_PDAF_CROP_X_0

Offset: 0x4a24
Byte Offset: 0x12890
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH10_PIXFMT_PDAF_CROP_Y_0

Offset: 0x4a25
Byte Offset: 0x12894
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH10_PIXFMT_PDAF_CONFIG0_0

Offset: 0x4a26
Byte Offset: 0x12898
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH10_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x4a27
Byte Offset: 0x1289c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH10_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x4a28
Byte Offset: 0x128a0
Read/Write: R/W
Parity Protection: N
Shadow: Y

SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH10_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.

Offset: 0x4a29
Byte Offset: 0x128a4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH10_PIXFMT_PDAF_CONFIG1_0

Offset: 0x4a2a
Byte Offset: 0x128a8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH10_DPCM_STRIP_0

Offset: 0x4a2f
Byte Offset: 0x128bc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH10_DPCM_CHUNK_FIRST_0

Offset: 0x4a30
Byte Offset: 0x128c0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH10_DPCM_CHUNK_BODY_0

Offset: 0x4a31
Byte Offset: 0x128c4

Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH10_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x4a32
Byte Offset: 0x128c8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH10_DPCM_CHUNK_LAST_0

Offset: 0x4a33
Byte Offset: 0x128cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH10_DPCM_STATISTICS_0_0

Offset: 0x4a34
Byte Offset: 0x128d0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH10_DPCM_STATISTICS_1_0

Offset: 0x4a35
Byte Offset: 0x128d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH10_DPCM_MODE_0

Offset: 0x4a36
Byte Offset: 0x128d8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH10_DPCM_CLAMP_HIGH_0

Offset: 0x4a37
Byte Offset: 0x128dc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH10_DPCM_CLAMP_LOW_0

Offset: 0x4a38
Byte Offset: 0x128e0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH10_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x4a3b
Byte Offset: 0x128ec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH10_ATOMP_SURFACE_OFFSET0_0

Offset: 0x4a3c
Byte Offset: 0x128f0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH10_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x4a3d
Byte Offset: 0x128f4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH10_ATOMP_SURFACE_STRIDE0_0

Offset: 0x4a3e
Byte Offset: 0x128f8
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH10_ATOMP_DPCM_CHUNK_0

Offset: 0x4a3f
Byte Offset: 0x128fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH10_ATOMP_SURFACE_OFFSET1_0

Offset: 0x4a40
Byte Offset: 0x12900
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH10_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x4a41
Byte Offset: 0x12904
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH10_ATOMP_SURFACE_STRIDE1_0

Offset: 0x4a42
Byte Offset: 0x12908
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH10_ATOMP_SURFACE_OFFSET2_0

Offset: 0x4a43
Byte Offset: 0x1290c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH10_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x4a44
Byte Offset: 0x12910
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH10_ATOMP_SURFACE_STRIDE2_0

Offset: 0x4a45
Byte Offset: 0x12914
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH10_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x4a46
Byte Offset: 0x12918
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH10_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x4a47
Byte Offset: 0x1291c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH10_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x4a48
Byte Offset: 0x12920
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH10_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x4a49
Byte Offset: 0x12924
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH10_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x4a4a
Byte Offset: 0x12928
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH10_ATOMP_HIGH_PRI_REQ_0

Offset: 0x4a4b
Byte Offset: 0x1292c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH10_ATOMP_RESERVE_0

Offset: 0x4a4c
Byte Offset: 0x12930
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

7.2.2.3.3 Video Input Channels 11-20 Registers

Channel 11 Specific Registers

VI_CH11_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x4b00
Byte Offset: 0x12c00
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH11_CHANSEL_0

Offset: 0x4b04
Byte Offset: 0x12c10
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH11_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x4b05
Byte Offset: 0x12c14
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH11_CONTROL_0

Channel prefix gets added to register name (VI_CH*)

Offset: 0x4b06
Byte Offset: 0x12c18
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH11_MATCH_0

SHADOWED CHANSEL REGISTERS

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x4b07
Byte Offset: 0x12c1c
Read/Write: R/W
Parity Protection: Y

Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH11_MATCH_DOL_0

Offset: 0x4b08
Byte Offset: 0x12c20
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH11_MATCH_VC_HI_0

Offset: 0x4b09
Byte Offset: 0x12c24
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH11_MATCH_DATATYPE_0

Offset: 0x4b0a
Byte Offset: 0x12c28
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH11_MATCH_FRAMEID_0

Offset: 0x4b0b
Byte Offset: 0x12c2c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH11_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT. The one exception is embedded data type which does not get remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on.

You are also likely not to have a wildcard but an exact match for data type. It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x4b0c
Byte Offset: 0x12c30
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH11_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x4b0d
Byte Offset: 0x12c34
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH11_FRAME_Y_0

Expected frame y dimension

Offset: 0x4b0e
Byte Offset: 0x12c38
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH11_EMBED_X_0
Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x4b0f
Byte Offset: 0x12c3c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH11_EMBED_Y_0
Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x4b10
Byte Offset: 0x12c40
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH11_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x4b11
Byte Offset: 0x12c44
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH11_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started. For example starting a flash in anticipation of an upcoming frame, or setup up a focusing process. This is done setting a tripline count which flags a notification once the number of lines has been reached. The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

The line timer events are not filtered by the cropped region, they are solely based on the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x4b12
Byte Offset: 0x12c48
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH11_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x4b13
Byte Offset: 0x12c4c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH11_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use. If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x4b14
Byte Offset: 0x12c50
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH11_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x4b15
Byte Offset: 0x12c54
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH11_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x4b16
Byte Offset: 0x12c58
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH11_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) - SKIP_Y_LINES

Offset: 0x4b17
Byte Offset: 0x12c5c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH11_OUT_Y_0

Offset: 0x4b18
Byte Offset: 0x12c60
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH11_NOTIFY_MASK_0

General warning

Offset: 0x4b19
Byte Offset: 0x12c64
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped

Bit	Reset	Description
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH11_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED.

The expected use of NOMATCH is for debug purposes.

If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification. To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x4b1a
Byte Offset: 0x12c68
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line

Bit	Reset	Description
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH11_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x4b1b
Byte Offset: 0x12c6c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)

Bit	Reset	Description
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH11_FRAME_COUNT_0

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame.

If in frame then FS=FE+1. FE can be perceived as a checksum check for FS.

Write any value to register to clear all counts.

Offset: 0x4b1c
Byte Offset: 0x12c70
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH11_LINE_TIMER_FIRST_0

Offset: 0x4b1d
Byte Offset: 0x12c74
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH11_FLUSH_FIRST_0

Offset: 0x4b1e
Byte Offset: 0x12c78
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH11_DOL_CTRL_0

Offset: 0x4b1f
Byte Offset: 0x12c7c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH11_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x4b20
Byte Offset: 0x12c80
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH11_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x4b21
Byte Offset: 0x12c84
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH11_PIXFMT_FORMAT_0

Offset: 0x4b22
Byte Offset: 0x12c88
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0

Bit	Reset	Description
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH11_PIXFMT_WIDE_0

Offset: 0x4b23
Byte Offset: 0x12c8c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH11_PIXFMT_PDAF_CROP_X_0

Offset: 0x4b24
Byte Offset: 0x12c90
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH11_PIXFMT_PDAF_CROP_Y_0

Offset: 0x4b25
Byte Offset: 0x12c94
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends

Bit	Reset	Description
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH11_PIXFMT_PDAF_CONFIG0_0

Offset: 0x4b26
Byte Offset: 0x12c98
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH11_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x4b27
Byte Offset: 0x12c9c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH11_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x4b28
Byte Offset: 0x12ca0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH11_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.

Offset: 0x4b29
Byte Offset: 0x12ca4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH11_PIXFMT_PDAF_CONFIG1_0

Offset: 0x4b2a
Byte Offset: 0x12ca8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH11_DPCM_STRIP_0

Offset: 0x4b2f
Byte Offset: 0x12cbc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH11_DPCM_CHUNK_FIRST_0

Offset: 0x4b30
Byte Offset: 0x12cc0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH11_DPCM_CHUNK_BODY_0

Offset: 0x4b31
Byte Offset: 0x12cc4

Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH11_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x4b32
Byte Offset: 0x12cc8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH11_DPCM_CHUNK_LAST_0

Offset: 0x4b33
Byte Offset: 0x12ccc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH11_DPCM_STATISTICS_0_0

Offset: 0x4b34
Byte Offset: 0x12cd0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH11_DPCM_STATISTICS_1_0

Offset: 0x4b35
Byte Offset: 0x12cd4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH11_DPCM_MODE_0

Offset: 0x4b36
Byte Offset: 0x12cd8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH11_DPCM_CLAMP_HIGH_0

Offset: 0x4b37
Byte Offset: 0x12cdc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH11_DPCM_CLAMP_LOW_0

Offset: 0x4b38
Byte Offset: 0x12ce0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH11_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x4b3b
Byte Offset: 0x12cec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH11_ATOMP_SURFACE_OFFSET0_0

Offset: 0x4b3c
Byte Offset: 0x12cf0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH11_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x4b3d
Byte Offset: 0x12cf4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH11_ATOMP_SURFACE_STRIDE0_0

Offset: 0x4b3e
Byte Offset: 0x12cf8
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH11_ATOMP_DPCM_CHUNK_0

Offset: 0x4b3f
Byte Offset: 0x12cfc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH11_ATOMP_SURFACE_OFFSET1_0

Offset: 0x4b40
Byte Offset: 0x12d00
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH11_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x4b41
Byte Offset: 0x12d04
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH11_ATOMP_SURFACE_STRIDE1_0

Offset: 0x4b42
Byte Offset: 0x12d08
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH11_ATOMP_SURFACE_OFFSET2_0

Offset: 0x4b43
Byte Offset: 0x12d0c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH11_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x4b44
Byte Offset: 0x12d10
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH11_ATOMP_SURFACE_STRIDE2_0

Offset: 0x4b45
Byte Offset: 0x12d14
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH11_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x4b46
Byte Offset: 0x12d18
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH11_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x4b47
Byte Offset: 0x12d1c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH11_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x4b48
Byte Offset: 0x12d20
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH11_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x4b49
Byte Offset: 0x12d24
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH11_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x4b4a
Byte Offset: 0x12d28
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH11_ATOMP_HIGH_PRI_REQ_0

Offset: 0x4b4b
Byte Offset: 0x12d2c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH11_ATOMP_RESERVE_0

Offset: 0x4b4c
Byte Offset: 0x12d30
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 12 Specific Registers

VI_CH12_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x4c00
Byte Offset: 0x13000
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH12_CHANSEL_0

Offset: 0x4c04
Byte Offset: 0x13010
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH12_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x4c05
Byte Offset: 0x13014
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH12_CONTROL_0

Channel prefix gets added to register name (VI_CH*)

Offset: 0x4c06
Byte Offset: 0x13018
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.

Bit	Parity Protection	Reset	Description
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH12_MATCH_0

SHADOWED CHANSEL REGISTERS

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x4c07
Byte Offset: 0x1301c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH12_MATCH_DOL_0

Offset: 0x4c08
Byte Offset: 0x13020
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH12_MATCH_VC_HI_0

Offset: 0x4c09
Byte Offset: 0x13024
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH12_MATCH_DATATYPE_0

Offset: 0x4c0a
Byte Offset: 0x13028
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH12_MATCH_FRAMEID_0

Offset: 0x4c0b
Byte Offset: 0x1302c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH12_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT. The one exception is embedded data type which does not get remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on.

You are also likely not to have a wildcard but an exact match for data type. It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x4c0c
Byte Offset: 0x13030
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.

Bit	Reset	Description
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH12_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x4c0d
Byte Offset: 0x13034
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH12_FRAME_Y_0

Expected frame y dimension

Offset: 0x4c0e
Byte Offset: 0x13038
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH12_EMBED_X_0

Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x4c0f
Byte Offset: 0x1303c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH12_EMBED_Y_0

Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x4c10
Byte Offset: 0x13040
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.

Bit	Reset	Description
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH12_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x4c11
Byte Offset: 0x13044
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH12_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started. For example starting a flash in anticipation of an upcoming frame, or setup up a

focusing process. This is done setting a tripline count which flags a notification once the number of lines has been reached. The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

The line timer events are not filtered by the cropped region, they are solely based on the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x4c12
Byte Offset: 0x13048
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH12_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x4c13
Byte Offset: 0x1304c
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH12_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use. If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x4c14
Byte Offset: 0x13050
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH12_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x4c15
Byte Offset: 0x13054
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH12_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x4c16
Byte Offset: 0x13058
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH12_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) - SKIP_Y_LINES

Offset: 0x4c17
Byte Offset: 0x1305c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH12_OUT_Y_0

Offset: 0x4c18
Byte Offset: 0x13060
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH12_NOTIFY_MASK_0

Offset: 0x4c19
Byte Offset: 0x13064
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH12_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED.

The expected use of NOMATCH is for debug purposes.

If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification. To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x4c1a
Byte Offset: 0x13068
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels

Bit	Reset	Description
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH12_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x4c1b
Byte Offset: 0x1306c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH12_FRAME_COUNT_0

FRAME COUNTS:

FS: a count of all valid frame start seen for channel saturates at 4095

FE: a count of all valid frame end seen for channel (including errored frames), saturates at 4095

FAULT: a count of all faulty frame end seen for channel, saturates at 255

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame. If in frame then FS=FE+1. FE can be perceived as a checksum check for FS.

Write any value to register to clear all counts.

Offset: 0x4c1c
Byte Offset: 0x13070
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH12_LINE_TIMER_FIRST_0

Offset: 0x4c1d
Byte Offset: 0x13074
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH12_FLUSH_FIRST_0

Offset: 0x4c1e
Byte Offset: 0x13078
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH12_DOL_CTRL_0

Offset: 0x4c1f
Byte Offset: 0x1307c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH12_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x4c20
Byte Offset: 0x13080
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;

Bit	R/W	Reset	Description
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH12_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x4c21
Byte Offset: 0x13084
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH12_PIXFMT_FORMAT_0

Offset: 0x4c22
Byte Offset: 0x13088
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/ RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH12_PIXFMT_WIDE_0

Offset: 0x4c23
Byte Offset: 0x1308c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH12_PIXFMT_PDAF_CROP_X_0

Offset: 0x4c24
Byte Offset: 0x13090
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH12_PIXFMT_PDAF_CROP_Y_0

Offset: 0x4c25
Byte Offset: 0x13094
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH12_PIXFMT_PDAF_CONFIG0_0

Offset: 0x4c26
Byte Offset: 0x13098
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH12_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x4c27
Byte Offset: 0x1309c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends

Bit	Reset	Description
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH12_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x4c28
Byte Offset: 0x130a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH12_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.

Offset: 0x4c29
Byte Offset: 0x130a4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH12_PIXFMT_PDAF_CONFIG1_0

Offset: 0x4c2a
Byte Offset: 0x130a8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH12_DPCM_STRIP_0

Offset: 0x4c2f
Byte Offset: 0x130bc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH12_DPCM_CHUNK_FIRST_0

Offset: 0x4c30
Byte Offset: 0x130c0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH12_DPCM_CHUNK_BODY_0

Offset: 0x4c31
Byte Offset: 0x130c4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH12_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x4c32
Byte Offset: 0x130c8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH12_DPCM_CHUNK_LAST_0

Offset: 0x4c33
Byte Offset: 0x130cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH12_DPCM_STATISTICS_0_0

Offset: 0x4c34
Byte Offset: 0x130d0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH12_DPCM_STATISTICS_1_0

Offset: 0x4c35
Byte Offset: 0x130d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH12_DPCM_MODE_0

Offset: 0x4c36
Byte Offset: 0x130d8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH12_DPCM_CLAMP_HIGH_0

Offset: 0x4c37
Byte Offset: 0x130dc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH12_DPCM_CLAMP_LOW_0

Offset: 0x4c38
Byte Offset: 0x130e0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH12_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x4c3b
Byte Offset: 0x130ec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH12_ATOMP_SURFACE_OFFSET0_0

Offset: 0x4c3c
Byte Offset: 0x130f0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH12_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x4c3d
Byte Offset: 0x130f4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH12_ATOMP_SURFACE_STRIDE0_0

Offset: 0x4c3e
Byte Offset: 0x130f8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH12_ATOMP_DPCM_CHUNK_0

Offset: 0x4c3f
Byte Offset: 0x130fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH12_ATOMP_SURFACE_OFFSET1_0

Offset: 0x4c40
Byte Offset: 0x13100
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH12_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x4c41
Byte Offset: 0x13104
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH12_ATOMP_SURFACE_STRIDE1_0

Offset: 0x4c42
Byte Offset: 0x13108
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH12_ATOMP_SURFACE_OFFSET2_0

Offset: 0x4c43
Byte Offset: 0x1310c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH12_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x4c44
Byte Offset: 0x13110
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH12_ATOMP_SURFACE_STRIDE2_0

Offset: 0x4c45
Byte Offset: 0x13114
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH12_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x4c46
Byte Offset: 0x13118
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH12_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x4c47
Byte Offset: 0x1311c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH12_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x4c48
Byte Offset: 0x13120
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH12_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x4c49
Byte Offset: 0x13124
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH12_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x4c4a
Byte Offset: 0x13128
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH12_ATOMP_HIGH_PRI_REQ_0

Offset: 0x4c4b
Byte Offset: 0x1312c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH12_ATOMP_RESERVE_0

Offset: 0x4c4c
Byte Offset: 0x13130
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 13 Specific Registers

VI_CH13_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x4d00
Byte Offset: 0x13400
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH13_CHANSEL_0

Offset: 0x4d04
Byte Offset: 0x13410
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH13_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x4d05
Byte Offset: 0x13414
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH13_CONTROL_0

Channel prefix gets added to register name (VI_CH*)

Offset: 0x4d06
Byte Offset: 0x13418
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH13_MATCH_0

SHADOWED CHANSEL REGISTERS

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x4d07
Byte Offset: 0x1341c
Read/Write: R/W
Parity Protection: Y

Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH13_MATCH_DOL_0

Offset: 0x4d08
Byte Offset: 0x13420
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH13_MATCH_VC_HI_0

Offset: 0x4d09
Byte Offset: 0x13424
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH13_MATCH_DATATYPE_0

Offset: 0x4d0a
Byte Offset: 0x13428
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH13_MATCH_FRAMEID_0

Offset: 0x4d0b
Byte Offset: 0x1342c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH13_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT. The one exception is embedded data type which does not get remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on.

You are also likely not to have a wildcard but an exact match for data type. It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x4d0c
Byte Offset: 0x13430
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH13_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x4d0d
Byte Offset: 0x13434
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH13_FRAME_Y_0

Expected frame y dimension

Offset: 0x4d0e
Byte Offset: 0x13438
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH13_EMBED_X_0
Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x4d0f
Byte Offset: 0x1343c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH13_EMBED_Y_0
Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x4d10
Byte Offset: 0x13440
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH13_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x4d11
Byte Offset: 0x13444
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH13_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started. For example starting a flash in anticipation of an upcoming frame, or setup up a focusing process. This is done setting a tripline count which flags a notification once the number of lines has been reached. The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

The line timer events are not filtered by the cropped region, they are solely based on the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x4d12
Byte Offset: 0x13448
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH13_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x4d13
Byte Offset: 0x1344c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH13_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use. If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x4d14
Byte Offset: 0x13450
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH13_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x4d15
Byte Offset: 0x13454
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH13_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x4d16
Byte Offset: 0x13458
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH13_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) - SKIP_Y_LINES

Offset: 0x4d17
Byte Offset: 0x1345c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH13_OUT_Y_0

Offset: 0x4d18
Byte Offset: 0x13460
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH13_NOTIFY_MASK_0

Offset: 0x4d19
Byte Offset: 0x13464
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped

Bit	Reset	Description
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH13_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED.

The expected use of NOMATCH is for debug purposes.

If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification. To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x4d1a
Byte Offset: 0x13468
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).

Bit	Reset	Description
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH13_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x4d1b
Byte Offset: 0x1346c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH13_FRAME_COUNT_0

FRAME COUNTS:

FS: a count of all valid frame start seen for channel saturates at 4095

FE: a count of all valid frame end seen for channel (incudling errored frames), saturates at 4095

FAULT: a count of all faulty frame end seen for channel, saturates at 255

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame.

If in frame then FS=FE+1. FE can be perceived as a checksum check for FS. Write any value to register to clear all counts.

Offset: 0x4d1c
Byte Offset: 0x13470
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH13_LINE_TIMER_FIRST_0

Offset: 0x4d1d
Byte Offset: 0x13474
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH13_FLUSH_FIRST_0

Offset: 0x4d1e
Byte Offset: 0x13478
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH13_DOL_CTRL_0

Offset: 0x4d1f
Byte Offset: 0x1347c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH13_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x4d20
Byte Offset: 0x13480
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH13_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x4d21
Byte Offset: 0x13484
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH13_PIXFMT_FORMAT_0

Offset: 0x4d22
Byte Offset: 0x13488
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0

Bit	Reset	Description
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH13_PIXFMT_WIDE_0

Offset: 0x4d23
Byte Offset: 0x1348c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH13_PIXFMT_PDAF_CROP_X_0

Offset: 0x4d24
Byte Offset: 0x13490
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH13_PIXFMT_PDAF_CROP_Y_0

Offset: 0x4d25
Byte Offset: 0x13494
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends

Bit	Reset	Description
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH13_PIXFMT_PDAF_CONFIG0_0

Offset: 0x4d26
Byte Offset: 0x13498
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH13_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x4d27
Byte Offset: 0x1349c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH13_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x4d28
Byte Offset: 0x134a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH13_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.

Offset: 0x4d29
Byte Offset: 0x134a4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH13_PIXFMT_PDAF_CONFIG1_0

Offset: 0x4d2a
Byte Offset: 0x134a8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH13_DPCM_STRIP_0

Offset: 0x4d2f
Byte Offset: 0x134bc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH13_DPCM_CHUNK_FIRST_0

Offset: 0x4d30
Byte Offset: 0x134c0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH13_DPCM_CHUNK_BODY_0

Offset: 0x4d31
Byte Offset: 0x134c4

Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH13_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x4d32
Byte Offset: 0x134c8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH13_DPCM_CHUNK_LAST_0

Offset: 0x4d33
Byte Offset: 0x134cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH13_DPCM_STATISTICS_0_0

Offset: 0x4d34
Byte Offset: 0x134d0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH13_DPCM_STATISTICS_1_0

Offset: 0x4d35
Byte Offset: 0x134d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH13_DPCM_MODE_0

Offset: 0x4d36
Byte Offset: 0x134d8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH13_DPCM_CLAMP_HIGH_0

Offset: 0x4d37
Byte Offset: 0x134dc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH13_DPCM_CLAMP_LOW_0

Offset: 0x4d38
Byte Offset: 0x134e0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH13_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x4d3b
Byte Offset: 0x134ec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH13_ATOMP_SURFACE_OFFSET0_0

Offset: 0x4d3c
Byte Offset: 0x134f0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH13_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x4d3d
Byte Offset: 0x134f4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH13_ATOMP_SURFACE_STRIDE0_0

Offset: 0x4d3e
Byte Offset: 0x134f8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH13_ATOMP_DPCM_CHUNK_0

Offset: 0x4d3f
Byte Offset: 0x134fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH13_ATOMP_SURFACE_OFFSET1_0

Offset: 0x4d40
Byte Offset: 0x13500
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH13_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x4d41
Byte Offset: 0x13504
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH13_ATOMP_SURFACE_STRIDE1_0

Offset: 0x4d42
Byte Offset: 0x13508
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH13_ATOMP_SURFACE_OFFSET2_0

Offset: 0x4d43
Byte Offset: 0x1350c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH13_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x4d44
Byte Offset: 0x13510
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH13_ATOMP_SURFACE_STRIDE2_0

Offset: 0x4d45
Byte Offset: 0x13514
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH13_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x4d46
Byte Offset: 0x13518
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH13_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x4d47
Byte Offset: 0x1351c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH13_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x4d48
Byte Offset: 0x13520
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH13_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x4d49
Byte Offset: 0x13524
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH13_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x4d4a
Byte Offset: 0x13528
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH13_ATOMP_HIGH_PRI_REQ_0

Offset: 0x4d4b
Byte Offset: 0x1352c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH13_ATOMP_RESERVE_0

Offset: 0x4d4c
Byte Offset: 0x13530
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 14 Specific Registers

VI_CH14_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS(non-shadowed)

Offset: 0x4e00
Byte Offset: 0x13800
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH14_CHANSEL_0

Offset: 0x4e04
Byte Offset: 0x13810
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH14_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x4e05
Byte Offset: 0x13814
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH14_CONTROL_0
Channel prefix gets added to register name (VI_CH*)
Offset: 0x4e06
Byte Offset: 0x13818
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.

Bit	Parity Protection	Reset	Description
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH14_MATCH_0

SHADOWED CHANSEL REGISTERS

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x4e07
Byte Offset: 0x1381c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH14_MATCH_DOL_0
Offset: 0x4e08
Byte Offset: 0x13820
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH14_MATCH_VC_HI_0
Offset: 0x4e09
Byte Offset: 0x13824
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH14_MATCH_DATATYPE_0
Offset: 0x4e0a
Byte Offset: 0x13828
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH14_MATCH_FRAMEID_0
Offset: 0x4e0b
Byte Offset: 0x1382c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH14_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT.

The one exception is embedded data type which does not get remapped through this register.

If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on.

You are also likely not to have a wildcard but an exact match for data type.

It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}).

Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x4e0c
Byte Offset: 0x13830
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.

Bit	Reset	Description
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH14_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x4e0d
Byte Offset: 0x13834
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH14_FRAME_Y_0

Expected frame y dimension

Offset: 0x4e0e
Byte Offset: 0x13838
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH14_EMBED_X_0
Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x4e0f
Byte Offset: 0x1383c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH14_EMBED_Y_0
Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x4e10
Byte Offset: 0x13840
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.

Bit	Reset	Description
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH14_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x4e11
Byte Offset: 0x13844
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH14_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started. For example starting a flash in anticipation of an upcoming frame, or setup up a

focusing process. This is done setting a tripline count which flags a notification once the number of lines has been reached. The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

The line timer events are not filtered by the cropped region, they are solely based on the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x4e12
Byte Offset: 0x13848
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH14_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x4e13
Byte Offset: 0x1384c
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH14_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use. If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x4e14
Byte Offset: 0x13850
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH14_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x4e15
Byte Offset: 0x13854
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH14_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x4e16
Byte Offset: 0x13858
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH14_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) - SKIP_Y_LINES

Offset: 0x4e17
Byte Offset: 0x1385c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH14_OUT_Y_0

Offset: 0x4e18
Byte Offset: 0x13860
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH14_NOTIFY_MASK_0

Offset: 0x4e19
Byte Offset: 0x13864
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH14_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED.

The expected use of NOMATCH is for debug purposes.

If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification. To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x4e1a
Byte Offset: 0x13868
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels

Bit	Reset	Description
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH14_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x4e1b
Byte Offset: 0x1386c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH14_FRAME_COUNT_0

FRAME COUNTS:

FS: a count of all valid frame start seen for channel saturates at 4095

FE: a count of all valid frame end seen for channel (incudling errored frames), saturates at 4095

FAULT: a count of all faulty frame end seen for channel, saturates at 255

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame. If in frame then FS=FE+1. FE can be perceived as a checksum check for FS. Write any value to register to clear all counts.

Offset: 0x4e1c
Byte Offset: 0x13870
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH14_LINE_TIMER_FIRST_0

Offset: 0x4e1d
Byte Offset: 0x13874
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH14_FLUSH_FIRST_0

Offset: 0x4e1e
Byte Offset: 0x13878
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH14_DOL_CTRL_0

Offset: 0x4e1f
Byte Offset: 0x1387c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH14_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x4e20
Byte Offset: 0x13880
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;

Bit	R/W	Reset	Description
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH14_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x4e21
Byte Offset: 0x13884
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH14_PIXFMT_FORMAT_0

Offset: 0x4e22
Byte Offset: 0x13888
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/ RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH14_PIXFMT_WIDE_0

Offset: 0x4e23
Byte Offset: 0x1388c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH14_PIXFMT_PDAF_CROP_X_0

Offset: 0x4e24
Byte Offset: 0x13890
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH14_PIXFMT_PDAF_CROP_Y_0

Offset: 0x4e25
Byte Offset: 0x13894
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH14_PIXFMT_PDAF_CONFIG0_0

Offset: 0x4e26
Byte Offset: 0x13898
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH14_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x4e27
Byte Offset: 0x1389c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH14_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x4e28
Byte Offset: 0x138a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH14_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.
Offset: 0x4e29
Byte Offset: 0x138a4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH14_PIXFMT_PDAF_CONFIG1_0

Offset: 0x4e2a
Byte Offset: 0x138a8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH14_DPCM_STRIP_0

Offset: 0x4e2f
Byte Offset: 0x138bc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH14_DPCM_CHUNK_FIRST_0

Offset: 0x4e30
Byte Offset: 0x138c0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH14_DPCM_CHUNK_BODY_0

Offset: 0x4e31
Byte Offset: 0x138c4
Read/Write: R/W
Parity Protection: Y

Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH14_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x4e32
Byte Offset: 0x138c8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH14_DPCM_CHUNK_LAST_0

Offset: 0x4e33
Byte Offset: 0x138cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH14_DPCM_STATISTICS_0_0

Offset: 0x4e34
Byte Offset: 0x138d0
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH14_DPCM_STATISTICS_1_0

Offset: 0x4e35
Byte Offset: 0x138d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH14_DPCM_MODE_0

Offset: 0x4e36
Byte Offset: 0x138d8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH14_DPCM_CLAMP_HIGH_0

Offset: 0x4e37
Byte Offset: 0x138dc

Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH14_DPCM_CLAMP_LOW_0

Offset: 0x4e38
Byte Offset: 0x138e0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH14_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x4e3b
Byte Offset: 0x138ec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH14_ATOMP_SURFACE_OFFSET0_0

Offset: 0x4e3c
Byte Offset: 0x138f0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH14_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x4e3d
Byte Offset: 0x138f4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH14_ATOMP_SURFACE_STRIDE0_0

Offset: 0x4e3e
Byte Offset: 0x138f8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH14_ATOMP_DPCM_CHUNK_0

Offset: 0x4e3f
Byte Offset: 0x138fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH14_ATOMP_SURFACE_OFFSET1_0

Offset: 0x4e40
Byte Offset: 0x13900
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH14_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x4e41
Byte Offset: 0x13904
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH14_ATOMP_SURFACE_STRIDE1_0

Offset: 0x4e42
Byte Offset: 0x13908

Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH14_ATOMP_SURFACE_OFFSET2_0

Offset: 0x4e43
Byte Offset: 0x1390c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH14_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x4e44
Byte Offset: 0x13910
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH14_ATOMP_SURFACE_STRIDE2_0

Offset: 0x4e45
Byte Offset: 0x13914

Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH14_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x4e46
Byte Offset: 0x13918
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH14_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x4e47
Byte Offset: 0x1391c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH14_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x4e48
Byte Offset: 0x13920
Read/Write: R/W
Parity Protection: Y

Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH14_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x4e49
Byte Offset: 0x13924
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH14_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x4e4a
Byte Offset: 0x13928
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH14_ATOMP_HIGH_PRI_REQ_0

Offset: 0x4e4b
Byte Offset: 0x1392c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH14_ATOMP_RESERVE_0

Offset: 0x4e4c
Byte Offset: 0x13930
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 15 Specific Registers

VI_CH15_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x4f00
Byte Offset: 0x13c00
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH15_CHANSEL_0

Offset: 0x4f04
Byte Offset: 0x13c10
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH15_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x4f05
Byte Offset: 0x13c14
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH15_CONTROL_0

Channel prefix gets added to register name (VI_CH*)

Offset: 0x4f06
Byte Offset: 0x13c18
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.

Bit	Parity Protection	Reset	Description
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH15_MATCH_0

SHADOWED CHANSEL REGISTERS

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x4f07
Byte Offset: 0x13c1c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH15_MATCH_DOL_0

Offset: 0x4f08
Byte Offset: 0x13c20
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH15_MATCH_VC_HI_0

Offset: 0x4f09
Byte Offset: 0x13c24
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH15_MATCH_DATATYPE_0

Offset: 0x4f0a
Byte Offset: 0x13c28
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH15_MATCH_FRAMEID_0

Offset: 0x4f0b
Byte Offset: 0x13c2c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH15_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT. The one exception is embedded data type which does not get remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on. You are also likely not to have a wildcard but an exact match for data type. It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x4f0c
Byte Offset: 0x13c30
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH15_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x4f0d
Byte Offset: 0x13c34
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH15_FRAME_Y_0

Expected frame y dimension

Offset: 0x4f0e
Byte Offset: 0x13c38
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH15_EMBED_X_0

Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x4f0f
Byte Offset: 0x13c3c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH15_EMBED_Y_0
Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x4f10
Byte Offset: 0x13c40
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH15_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x4f11
Byte Offset: 0x13c44
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH15_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started. For example starting a flash in anticipation of an upcoming frame, or setup up a focusing process. This is done setting a tripline count which flags a notification once the number of lines has been reached. The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. The line timer events are not filtered by the cropped region, they are solely based on the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x4f12
Byte Offset: 0x13c48
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.

Bit	Reset	Description
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH15_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x4f13
Byte Offset: 0x13c4c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH15_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use. If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x4f14
Byte Offset: 0x13c50
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH15_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x4f15
Byte Offset: 0x13c54
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH15_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x4f16
Byte Offset: 0x13c58
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH15_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) - SKIP_Y_LINES

Offset: 0x4f17
Byte Offset: 0x13c5c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH15_OUT_Y_0

Offset: 0x4f18
Byte Offset: 0x13c60
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH15_NOTIFY_MASK_0

Offset: 0x4f19
Byte Offset: 0x13c64
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame

Bit	Reset	Description
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH15_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED.The expected use of NOMATCH is for debug purposes.

If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification. To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x4f1a
Byte Offset: 0x13c68
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH15_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x4f1b
Byte Offset: 0x13c6c
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH15_FRAME_COUNT_0

FRAME COUNTS:

FS: a count of all valid frame start seen for channel saturates at 4095

FE: a count of all valid frame end seen for channel (incudling errored frames), saturates at 4095

FAULT: a count of all faulty frame end seen for channel, saturates at 255

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame. If in frame then FS=FE+1. FE can be perceived as a checksum check for FS.

Write any value to register to clear all counts.

Offset: 0x4f1c
Byte Offset: 0x13c70
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095

Bit	Reset	Description
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH15_LINE_TIMER_FIRST_0

Offset: 0x4f1d
Byte Offset: 0x13c74
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH15_FLUSH_FIRST_0

Offset: 0x4f1e
Byte Offset: 0x13c78
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH15_DOL_CTRL_0

Offset: 0x4f1f
Byte Offset: 0x13c7c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH15_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x4f20
Byte Offset: 0x13c80
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH15_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x4f21
Byte Offset: 0x13c84
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH15_PIXFMT_FORMAT_0

Offset: 0x4f22
Byte Offset: 0x13c88
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0

Bit	Reset	Description
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH15_PIXFMT_WIDE_0

Offset: 0x4f23
Byte Offset: 0x13c8c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH15_PIXFMT_PDAF_CROP_X_0

Offset: 0x4f24
Byte Offset: 0x13c90
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH15_PIXFMT_PDAF_CROP_Y_0

Offset: 0x4f25
Byte Offset: 0x13c94
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends

Bit	Reset	Description
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH15_PIXFMT_PDAF_CONFIG0_0

Offset: 0x4f26
Byte Offset: 0x13c98
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH15_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x4f27
Byte Offset: 0x13c9c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH15_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x4f28
Byte Offset: 0x13ca0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH15_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.

Offset: 0x4f29
Byte Offset: 0x13ca4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH15_PIXFMT_PDAF_CONFIG1_0

Offset: 0x4f2a
Byte Offset: 0x13ca8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH15_DPCM_STRIP_0

Offset: 0x4f2f
Byte Offset: 0x13cbc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH15_DPCM_CHUNK_FIRST_0

Offset: 0x4f30
Byte Offset: 0x13cc0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH15_DPCM_CHUNK_BODY_0

Offset: 0x4f31
Byte Offset: 0x13cc4

Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH15_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x4f32
Byte Offset: 0x13cc8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH15_DPCM_CHUNK_LAST_0

Offset: 0x4f33
Byte Offset: 0x13ccc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH15_DPCM_STATISTICS_0_0

Offset: 0x4f34
Byte Offset: 0x13cd0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH15_DPCM_STATISTICS_1_0

Offset: 0x4f35
Byte Offset: 0x13cd4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH15_DPCM_MODE_0

Offset: 0x4f36
Byte Offset: 0x13cd8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH15_DPCM_CLAMP_HIGH_0

Offset: 0x4f37
Byte Offset: 0x13cdc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH15_DPCM_CLAMP_LOW_0

Offset: 0x4f38
Byte Offset: 0x13ce0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH15_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x4f3b
Byte Offset: 0x13cec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH15_ATOMP_SURFACE_OFFSET0_0

Offset: 0x4f3c
Byte Offset: 0x13cf0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH15_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x4f3d
Byte Offset: 0x13cf4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH15_ATOMP_SURFACE_STRIDE0_0

Offset: 0x4f3e
Byte Offset: 0x13cf8
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH15_ATOMP_DPCM_CHUNK_0

Offset: 0x4f3f
Byte Offset: 0x13cfc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH15_ATOMP_SURFACE_OFFSET1_0

Offset: 0x4f40
Byte Offset: 0x13d00
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH15_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x4f41
Byte Offset: 0x13d04
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH15_ATOMP_SURFACE_STRIDE1_0

Offset: 0x4f42
Byte Offset: 0x13d08
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH15_ATOMP_SURFACE_OFFSET2_0

Offset: 0x4f43
Byte Offset: 0x13d0c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH15_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x4f44
Byte Offset: 0x13d10
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH15_ATOMP_SURFACE_STRIDE2_0

Offset: 0x4f45
Byte Offset: 0x13d14
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH15_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x4f46
Byte Offset: 0x13d18
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH15_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x4f47
Byte Offset: 0x13d1c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH15_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x4f48
Byte Offset: 0x13d20
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH15_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x4f49
Byte Offset: 0x13d24
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH15_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x4f4a
Byte Offset: 0x13d28
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH15_ATOMP_HIGH_PRI_REQ_0

Offset: 0x4f4b
Byte Offset: 0x13d2c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH15_ATOMP_RESERVE_0

Offset: 0x4f4c
Byte Offset: 0x13d30
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 16 Specific Registers

VI_CH16_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x5000
Byte Offset: 0x14000
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH16_CHANSEL_0

Offset: 0x5004
Byte Offset: 0x14010
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH16_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x5005
Byte Offset: 0x14014
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH16_CONTROL_0

Channel prefix gets added to register name (VI_CH*)

Offset: 0x5006
Byte Offset: 0x14018
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH16_MATCH_0

SHADOWED CHANSEL REGISTERS

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x5007
Byte Offset: 0x1401c
Read/Write: R/W
Parity Protection: Y

Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH16_MATCH_DOL_0

Offset: 0x5008
Byte Offset: 0x14020
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH16_MATCH_VC_HI_0

Offset: 0x5009
Byte Offset: 0x14024
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH16_MATCH_DATATYPE_0

Offset: 0x500a
Byte Offset: 0x14028
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH16_MATCH_FRAMEID_0

Offset: 0x500b
Byte Offset: 0x1402c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH16_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT. The one exception is embedded data type which does not get remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on. You are also likely not to have a wildcard but an exact match for data type. It is illegal to program

OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x500c
Byte Offset: 0x14030
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH16_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x500d
Byte Offset: 0x14034
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH16_FRAME_Y_0

Expected frame y dimension

Offset: 0x500e
Byte Offset: 0x14038
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH16_EMBED_X_0
Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x500f
Byte Offset: 0x1403c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH16_EMBED_Y_0
Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x5010
Byte Offset: 0x14040
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.

Bit	Reset	Description
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH16_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x5011
Byte Offset: 0x14044
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH16_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started.

For example starting a flash in anticipation of an upcoming frame, or setup up a focusing process. This is done setting a tripline count which flags a notification once the number of lines has been reached. The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. The line timer events are not filtered by the cropped region, they are solely based on the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x5012
Byte Offset: 0x14048
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH16_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data). Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x5013
Byte Offset: 0x1404c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH16_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use. If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x5014
Byte Offset: 0x14050
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH16_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x5015
Byte Offset: 0x14054
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH16_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x5016
Byte Offset: 0x14058
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH16_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) -   SKIP_Y_LINES

Offset: 0x5017
Byte Offset: 0x1405c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH16_OUT_Y_0

Offset: 0x5018
Byte Offset: 0x14060
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH16_NOTIFY_MASK_0

Offset: 0x5019
Byte Offset: 0x14064
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH16_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED.The expected use of NOMATCH is for debug purposes. If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification.

To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x501a
Byte Offset: 0x14068
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH16_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x501b
Byte Offset: 0x1406c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH16_FRAME_COUNT_0

FRAME COUNTS:

FS: a count of all valid frame start seen for channel saturates at 4095

FE: a count of all valid frame end seen for channel (incudling errored frames), saturates at 4095

FAULT: a count of all faulty frame end seen for channel, saturates at 255

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame.

If in frame then FS=FE+1. FE can be perceived as a checksum check for FS.

Write any value to register to clear all counts.

Offset: 0x501c
Byte Offset: 0x14070
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH16_LINE_TIMER_FIRST_0

Offset: 0x501d
Byte Offset: 0x14074
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH16_FLUSH_FIRST_0

Offset: 0x501e
Byte Offset: 0x14078
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH16_DOL_CTRL_0

Offset: 0x501f
Byte Offset: 0x1407c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH16_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x5020
Byte Offset: 0x14080
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;

Bit	R/W	Reset	Description
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH16_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x5021
Byte Offset: 0x14084
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH16_PIXFMT_FORMAT_0

Offset: 0x5022
Byte Offset: 0x14088
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/ RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH16_PIXFMT_WIDE_0

Offset: 0x5023
Byte Offset: 0x1408c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH16_PIXFMT_PDAF_CROP_X_0

Offset: 0x5024
Byte Offset: 0x14090
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH16_PIXFMT_PDAF_CROP_Y_0

Offset: 0x5025
Byte Offset: 0x14094
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH16_PIXFMT_PDAF_CONFIG0_0

Offset: 0x5026
Byte Offset: 0x14098
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH16_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x5027
Byte Offset: 0x1409c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends

Bit	Reset	Description
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH16_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x5028
Byte Offset: 0x140a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH16_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.

Offset: 0x5029
Byte Offset: 0x140a4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH16_PIXFMT_PDAF_CONFIG1_0

Offset: 0x502a
Byte Offset: 0x140a8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH16_DPCM_STRIP_0

Offset: 0x502f
Byte Offset: 0x140bc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH16_DPCM_CHUNK_FIRST_0

Offset: 0x5030
Byte Offset: 0x140c0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH16_DPCM_CHUNK_BODY_0

Offset: 0x5031
Byte Offset: 0x140c4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH16_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x5032
Byte Offset: 0x140c8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH16_DPCM_CHUNK_LAST_0

Offset: 0x5033
Byte Offset: 0x140cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH16_DPCM_STATISTICS_0_0

Offset: 0x5034
Byte Offset: 0x140d0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH16_DPCM_STATISTICS_1_0

Offset: 0x5035
Byte Offset: 0x140d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH16_DPCM_MODE_0

Offset: 0x5036
Byte Offset: 0x140d8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH16_DPCM_CLAMP_HIGH_0

Offset: 0x5037
Byte Offset: 0x140dc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH16_DPCM_CLAMP_LOW_0

Offset: 0x5038
Byte Offset: 0x140e0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH16_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x503b
Byte Offset: 0x140ec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH16_ATOMP_SURFACE_OFFSET0_0

Offset: 0x503c
Byte Offset: 0x140f0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH16_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x503d
Byte Offset: 0x140f4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH16_ATOMP_SURFACE_STRIDE0_0 Offset: 0x503e
Byte Offset: 0x140f8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH16_ATOMP_DPCM_CHUNK_0

Offset: 0x503f
Byte Offset: 0x140fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH16_ATOMP_SURFACE_OFFSET1_0

Offset: 0x5040
Byte Offset: 0x14100
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH16_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x5041
Byte Offset: 0x14104
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH16_ATOMP_SURFACE_STRIDE1_0

Offset: 0x5042
Byte Offset: 0x14108
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH16_ATOMP_SURFACE_OFFSET2_0

Offset: 0x5043
Byte Offset: 0x1410c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH16_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x5044
Byte Offset: 0x14110
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH16_ATOMP_SURFACE_STRIDE2_0

Offset: 0x5045
Byte Offset: 0x14114
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH16_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x5046
Byte Offset: 0x14118
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH16_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x5047
Byte Offset: 0x1411c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH16_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x5048
Byte Offset: 0x14120
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH16_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x5049
Byte Offset: 0x14124
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH16_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x504a
Byte Offset: 0x14128
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH16_ATOMP_HIGH_PRI_REQ_0

Offset: 0x504b
Byte Offset: 0x1412c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH16_ATOMP_RESERVE_0

Offset: 0x504c
Byte Offset: 0x14130
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 17 Specific Registers

VI_CH17_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x5100
Byte Offset: 0x14400
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH17_CHANSEL_0

Offset: 0x5104
Byte Offset: 0x14410
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH17_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. put channel enable to a separate register field to avoid SW race condition.

Offset: 0x5105
Byte Offset: 0x14414
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH17_CONTROL_0

Channel prefix gets added to register name (VI_CH*)

Offset: 0x5106
Byte Offset: 0x14418
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH17_MATCH_0

SHADOWED CHANSEL REGISTERS

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x5107
Byte Offset: 0x1441c
Read/Write: R/W
Parity Protection: Y

Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH17_MATCH_DOL_0

Offset: 0x5108
Byte Offset: 0x14420
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH17_MATCH_VC_HI_0

Offset: 0x5109
Byte Offset: 0x14424
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH17_MATCH_DATATYPE_0

Offset: 0x510a
Byte Offset: 0x14428
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH17_MATCH_FRAMEID_0

Offset: 0x510b
Byte Offset: 0x1442c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH17_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT. The one exception is embedded data type which does not get remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on. You are also likely not to have a wildcard but an exact match for data type. It is illegal to program

OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x510c
Byte Offset: 0x14430
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH17_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x510d
Byte Offset: 0x14434
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH17_FRAME_Y_0

Expected frame y dimension

Offset: 0x510e
Byte Offset: 0x14438
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH17_EMBED_X_0

Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x510f
Byte Offset: 0x1443c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH17_EMBED_Y_0

Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x5110
Byte Offset: 0x14440
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.

Bit	Reset	Description
24	0x0	**EXPECT**: Embedded data expected within channel.
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH17_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x5111
Byte Offset: 0x14444
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH17_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started. For example starting a flash in anticipation of an upcoming frame, or setup up a

focusing process. This is done setting a tripline count which flags a notification once the number of lines has been reached. The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. The line timer events are not filtered by the cropped region, they are solely based on the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x5112
Byte Offset: 0x14448
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH17_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x5113
Byte Offset: 0x1444c
Read/Write: R/W
Parity Protection: Y

Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH17_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use. If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x5114
Byte Offset: 0x14450
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH17_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x5115
Byte Offset: 0x14454
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH17_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x5116
Byte Offset: 0x14458
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH17_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) -   SKIP_Y_LINES

Offset: 0x5117
Byte Offset: 0x1445c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH17_OUT_Y_0

Offset: 0x5118
Byte Offset: 0x14460
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH17_NOTIFY_MASK_0

Offset: 0x5119
Byte Offset: 0x14464
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH17_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED. The expected use of NOMATCH is for debug purposes.

If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification. To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x511a
Byte Offset: 0x14468
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH17_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x511b
Byte Offset: 0x1446c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH17_FRAME_COUNT_0

FRAME COUNTS:

FS: a count of all valid frame start seen for channel saturates at 4095

FE: a count of all valid frame end seen for channel (incudling errored frames), saturates at 4095

FAULT: a count of all faulty frame end seen for channel, saturates at 255

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame. If in frame then FS=FE+1. FE can be perceived as a checksum check for FS.

Write any value to register to clear all counts.

Offset: 0x511c
Byte Offset: 0x14470
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH17_LINE_TIMER_FIRST_0

Offset: 0x511d
Byte Offset: 0x14474
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH17_FLUSH_FIRST_0

Offset: 0x511e
Byte Offset: 0x14478
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH17_DOL_CTRL_0

Offset: 0x511f
Byte Offset: 0x1447c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH17_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x5120
Byte Offset: 0x14480
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;

Bit	R/W	Reset	Description
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH17_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x5121
Byte Offset: 0x14484
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH17_PIXFMT_FORMAT_0

Offset: 0x5122
Byte Offset: 0x14488
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/ RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH17_PIXFMT_WIDE_0

Offset: 0x5123
Byte Offset: 0x1448c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH17_PIXFMT_PDAF_CROP_X_0

Offset: 0x5124
Byte Offset: 0x14490
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH17_PIXFMT_PDAF_CROP_Y_0

Offset: 0x5125
Byte Offset: 0x14494
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH17_PIXFMT_PDAF_CONFIG0_0

Offset: 0x5126
Byte Offset: 0x14498
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH17_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x5127
Byte Offset: 0x1449c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends

Bit	Reset	Description
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH17_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x5128
Byte Offset: 0x144a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH17_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.

Offset: 0x5129
Byte Offset: 0x144a4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH17_PIXFMT_PDAF_CONFIG1_0

Offset: 0x512a
Byte Offset: 0x144a8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH17_DPCM_STRIP_0

Offset: 0x512f
Byte Offset: 0x144bc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH17_DPCM_CHUNK_FIRST_0

Offset: 0x5130
Byte Offset: 0x144c0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH17_DPCM_CHUNK_BODY_0

Offset: 0x5131
Byte Offset: 0x144c4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH17_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x5132
Byte Offset: 0x144c8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH17_DPCM_CHUNK_LAST_0

Offset: 0x5133
Byte Offset: 0x144cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH17_DPCM_STATISTICS_0_0

Offset: 0x5134
Byte Offset: 0x144d0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH17_DPCM_STATISTICS_1_0

Offset: 0x5135
Byte Offset: 0x144d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH17_DPCM_MODE_0

Offset: 0x5136
Byte Offset: 0x144d8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH17_DPCM_CLAMP_HIGH_0

Offset: 0x5137
Byte Offset: 0x144dc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH17_DPCM_CLAMP_LOW_0

Offset: 0x5138
Byte Offset: 0x144e0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH17_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x513b
Byte Offset: 0x144ec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH17_ATOMP_SURFACE_OFFSET0_0

Offset: 0x513c
Byte Offset: 0x144f0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH17_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x513d
Byte Offset: 0x144f4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH17_ATOMP_SURFACE_STRIDE0_0

Offset: 0x513e
Byte Offset: 0x144f8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH17_ATOMP_DPCM_CHUNK_0

Offset: 0x513f
Byte Offset: 0x144fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH17_ATOMP_SURFACE_OFFSET1_0

Offset: 0x5140
Byte Offset: 0x14500
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH17_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x5141
Byte Offset: 0x14504
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH17_ATOMP_SURFACE_STRIDE1_0

Offset: 0x5142
Byte Offset: 0x14508
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH17_ATOMP_SURFACE_OFFSET2_0

Offset: 0x5143
Byte Offset: 0x1450c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH17_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x5144
Byte Offset: 0x14510
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH17_ATOMP_SURFACE_STRIDE2_0

Offset: 0x5145
Byte Offset: 0x14514
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH17_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x5146
Byte Offset: 0x14518
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH17_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x5147
Byte Offset: 0x1451c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH17_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x5148
Byte Offset: 0x14520
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH17_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x5149
Byte Offset: 0x14524
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH17_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x514a
Byte Offset: 0x14528
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH17_ATOMP_HIGH_PRI_REQ_0

Offset: 0x514b
Byte Offset: 0x1452c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH17_ATOMP_RESERVE_0

Offset: 0x514c
Byte Offset: 0x14530
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 18 Specific Registers

VI_CH18_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x5200
Byte Offset: 0x14800
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH18_CHANSEL_0

Offset: 0x5204
Byte Offset: 0x14810
Read/Write: R/W
Parity Protection: N
Shadow: Y

SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH18_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. put channel enable to a separate register field to avoid SW race condition.

Offset: 0x5205
Byte Offset: 0x14814
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH18_CONTROL_0

Channel prefix gets added to register name (VI_CH*)

Offset: 0x5206
Byte Offset: 0x14818
Read/Write: R/W
Parity Protection: See table below
Shadow: N

SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH18_MATCH_0

SHADOWED CHANSEL REGISTERS

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x5207
Byte Offset: 0x1481c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH18_MATCH_DOL_0

Offset: 0x5208
Byte Offset: 0x14820
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH18_MATCH_VC_HI_0

Offset: 0x5209
Byte Offset: 0x14824
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH18_MATCH_DATATYPE_0

Offset: 0x520a
Byte Offset: 0x14828
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH18_MATCH_FRAMEID_0

Offset: 0x520b
Byte Offset: 0x1482c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH18_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT. The one exception is embedded data type which does not get

remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on. You are also likely not to have a wildcard but an exact match for data type. It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x520c
Byte Offset: 0x14830
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH18_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x520d
Byte Offset: 0x14834
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH18_FRAME_Y_0

Expected frame y dimension

Offset: 0x520e
Byte Offset: 0x14838
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH18_EMBED_X_0

Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x520f
Byte Offset: 0x1483c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH18_EMBED_Y_0

Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x5210
Byte Offset: 0x14840
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH18_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x5211
Byte Offset: 0x14844
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH18_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started.

For example starting a flash in anticipation of an upcoming frame, or setup up a focusing process. This is done setting a tripline count which flags a notification once the number of lines has been reached. The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. The line timer events are not filtered by the cropped region, they are solely based on the input frame.The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x5212
Byte Offset: 0x14848
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH18_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x5213
Byte Offset: 0x1484c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH18_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use. If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x5214
Byte Offset: 0x14850
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH18_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x5215
Byte Offset: 0x14854
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH18_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x5216
Byte Offset: 0x14858
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH18_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) - SKIP_Y_LINES

Offset: 0x5217
Byte Offset: 0x1485c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH18_OUT_Y_0

Offset: 0x5218
Byte Offset: 0x14860
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH18_NOTIFY_MASK_0

Offset: 0x5219
Byte Offset: 0x14864
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped

Bit	Reset	Description
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH18_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED.

The expected use of NOMATCH is for debug purposes.

If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification. To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x521a
Byte Offset: 0x14868
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).

Bit	Reset	Description
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH18_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x521b
Byte Offset: 0x1486c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded

Bit	Reset	Description
15:0	X	**FRAMEID**: Frame ID

VI_CH18_FRAME_COUNT_0

FRAME COUNTS:

FS: a count of all valid frame start seen for channel saturates at 4095

FE: a count of all valid frame end seen for channel (incudling errored frames), saturates at 4095

FAULT: a count of all faulty frame end seen for channel, saturates at 255

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame. If in frame then FS=FE+1. FE can be perceived as a checksum check for FS. Write any value to register to clear all counts.

Offset: 0x521c
Byte Offset: 0x14870
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH18_LINE_TIMER_FIRST_0

Offset: 0x521d
Byte Offset: 0x14874
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH18_FLUSH_FIRST_0

Offset: 0x521e
Byte Offset: 0x14878
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH18_DOL_CTRL_0

Offset: 0x521f
Byte Offset: 0x1487c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH18_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x5220
Byte Offset: 0x14880
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH18_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x5221
Byte Offset: 0x14884
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel

Bit	Reset	Description
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH18_PIXFMT_FORMAT_0 Offset: 0x5222
Byte Offset: 0x14888
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0

Bit	Reset	Description
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH18_PIXFMT_WIDE_0

Offset: 0x5223
Byte Offset: 0x1488c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH18_PIXFMT_PDAF_CROP_X_0

Offset: 0x5224
Byte Offset: 0x14890
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH18_PIXFMT_PDAF_CROP_Y_0

Offset: 0x5225
Byte Offset: 0x14894
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends

Bit	Reset	Description
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH18_PIXFMT_PDAF_CONFIG0_0

Offset: 0x5226
Byte Offset: 0x14898
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH18_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x5227
Byte Offset: 0x1489c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH18_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x5228
Byte Offset: 0x148a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH18_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.

Offset: 0x5229
Byte Offset: 0x148a4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH18_PIXFMT_PDAF_CONFIG1_0

Offset: 0x522a
Byte Offset: 0x148a8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH18_DPCM_STRIP_0

Offset: 0x522f
Byte Offset: 0x148bc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH18_DPCM_CHUNK_FIRST_0

Offset: 0x5230
Byte Offset: 0x148c0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH18_DPCM_CHUNK_BODY_0

Offset: 0x5231
Byte Offset: 0x148c4

Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH18_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x5232
Byte Offset: 0x148c8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH18_DPCM_CHUNK_LAST_0

Offset: 0x5233
Byte Offset: 0x148cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH18_DPCM_STATISTICS_0_0

Offset: 0x5234
Byte Offset: 0x148d0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH18_DPCM_STATISTICS_1_0

Offset: 0x5235
Byte Offset: 0x148d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH18_DPCM_MODE_0

Offset: 0x5236
Byte Offset: 0x148d8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH18_DPCM_CLAMP_HIGH_0

Offset: 0x5237
Byte Offset: 0x148dc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH18_DPCM_CLAMP_LOW_0

Offset: 0x5238
Byte Offset: 0x148e0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH18_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x523b
Byte Offset: 0x148ec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH18_ATOMP_SURFACE_OFFSET0_0

Offset: 0x523c
Byte Offset: 0x148f0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH18_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x523d
Byte Offset: 0x148f4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH18_ATOMP_SURFACE_STRIDE0_0

Offset: 0x523e
Byte Offset: 0x148f8
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH18_ATOMP_DPCM_CHUNK_0

Offset: 0x523f
Byte Offset: 0x148fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH18_ATOMP_SURFACE_OFFSET1_0

Offset: 0x5240
Byte Offset: 0x14900
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH18_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x5241
Byte Offset: 0x14904
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH18_ATOMP_SURFACE_STRIDE1_0

Offset: 0x5242
Byte Offset: 0x14908
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH18_ATOMP_SURFACE_OFFSET2_0

Offset: 0x5243
Byte Offset: 0x1490c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH18_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x5244
Byte Offset: 0x14910
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH18_ATOMP_SURFACE_STRIDE2_0

Offset: 0x5245
Byte Offset: 0x14914
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH18_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x5246
Byte Offset: 0x14918
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH18_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x5247
Byte Offset: 0x1491c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH18_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x5248
Byte Offset: 0x14920
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH18_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x5249
Byte Offset: 0x14924
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH18_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x524a
Byte Offset: 0x14928
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH18_ATOMP_HIGH_PRI_REQ_0

Offset: 0x524b
Byte Offset: 0x1492c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH18_ATOMP_RESERVE_0

Offset: 0x524c
Byte Offset: 0x14930
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 19 Specific Registers

VI_CH19_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x5300
Byte Offset: 0x14c00
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH19_CHANSEL_0

Offset: 0x5304
Byte Offset: 0x14c10
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH19_ENABLE_0
Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x5305
Byte Offset: 0x14c14
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH19_CONTROL_0
Channel prefix gets added to register name (VI_CH*)
Offset: 0x5306
Byte Offset: 0x14c18
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.

Bit	Parity Protection	Reset	Description
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH19_MATCH_0

SHADOWED CHANSEL REGISTERS

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x5307
Byte Offset: 0x14c1c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH19_MATCH_DOL_0

Offset: 0x5308
Byte Offset: 0x14c20
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH19_MATCH_VC_HI_0

Offset: 0x5309
Byte Offset: 0x14c24
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH19_MATCH_DATATYPE_0

Offset: 0x530a
Byte Offset: 0x14c28
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH19_MATCH_FRAMEID_0

Offset: 0x530b
Byte Offset: 0x14c2c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH19_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT. The one exception is embedded data type which does not get remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on.

You are also likely not to have a wildcard but an exact match for data type. It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x530c
Byte Offset: 0x14c30
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.

Bit	Reset	Description
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH19_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x530d
Byte Offset: 0x14c34
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH19_FRAME_Y_0

Expected frame y dimension

Offset: 0x530e
Byte Offset: 0x14c38
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH19_EMBED_X_0
Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x530f
Byte Offset: 0x14c3c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH19_EMBED_Y_0
Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x5310
Byte Offset: 0x14c40
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.

Bit	Reset	Description
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH19_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x5311
Byte Offset: 0x14c44
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH19_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started.

For example starting a flash in anticipation of an upcoming frame, or setup up a focusing process. This is done setting a tripline count which flags a notification once the number of lines has been reached.

The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

The line timer events are not filtered by the cropped region, they are solely based on the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x5312
Byte Offset: 0x14c48
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH19_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x5313
Byte Offset: 0x14c4c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH19_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use. If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x5314
Byte Offset: 0x14c50
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH19_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x5315
Byte Offset: 0x14c54
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH19_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x5316
Byte Offset: 0x14c58
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH19_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) - SKIP_Y_LINES

Offset: 0x5317
Byte Offset: 0x14c5c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH19_OUT_Y_0

Offset: 0x5318
Byte Offset: 0x14c60
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH19_NOTIFY_MASK_0

Offset: 0x5319
Byte Offset: 0x14c64
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH19_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED.

The expected use of NOMATCH is for debug purposes.

If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification.

To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x531a
Byte Offset: 0x14c68
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data

Bit	Reset	Description
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH19_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x531b
Byte Offset: 0x14c6c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH19_FRAME_COUNT_0

FRAME COUNTS:

FS: a count of all valid frame start seen for channel saturates at 4095

FE: a count of all valid frame end seen for channel (incudling errored frames), saturates at 4095

FAULT: a count of all faulty frame end seen for channel, saturates at 255

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame.

If in frame then FS=FE+1. FE can be perceived as a checksum check for FS. Write any value to register to clear all counts.

Offset: 0x531c
Byte Offset: 0x14c70
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH19_LINE_TIMER_FIRST_0

Offset: 0x531d
Byte Offset: 0x14c74
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH19_FLUSH_FIRST_0

Offset: 0x531e
Byte Offset: 0x14c78
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH19_DOL_CTRL_0

Offset: 0x531f
Byte Offset: 0x14c7c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH19_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x5320
Byte Offset: 0x14c80
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH19_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x5321
Byte Offset: 0x14c84
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH19_PIXFMT_FORMAT_0

Offset: 0x5322
Byte Offset: 0x14c88
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0

Bit	Reset	Description
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH19_PIXFMT_WIDE_0

Offset: 0x5323
Byte Offset: 0x14c8c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH19_PIXFMT_PDAF_CROP_X_0

Offset: 0x5324
Byte Offset: 0x14c90
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH19_PIXFMT_PDAF_CROP_Y_0

Offset: 0x5325
Byte Offset: 0x14c94
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends

Bit	Reset	Description
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH19_PIXFMT_PDAF_CONFIG0_0

Offset: 0x5326
Byte Offset: 0x14c98
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH19_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x5327
Byte Offset: 0x14c9c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH19_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x5328
Byte Offset: 0x14ca0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH19_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.

Offset: 0x5329
Byte Offset: 0x14ca4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH19_PIXFMT_PDAF_CONFIG1_0

Offset: 0x532a
Byte Offset: 0x14ca8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH19_DPCM_STRIP_0

Offset: 0x532f
Byte Offset: 0x14cbc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH19_DPCM_CHUNK_FIRST_0

Offset: 0x5330
Byte Offset: 0x14cc0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH19_DPCM_CHUNK_BODY_0

Offset: 0x5331
Byte Offset: 0x14cc4
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH19_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x5332
Byte Offset: 0x14cc8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH19_DPCM_CHUNK_LAST_0

Offset: 0x5333
Byte Offset: 0x14ccc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH19_DPCM_STATISTICS_0_0Offset: 0x5334
Byte Offset: 0x14cd0
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH19_DPCM_STATISTICS_1_0

Offset: 0x5335
Byte Offset: 0x14cd4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH19_DPCM_MODE_0

Offset: 0x5336
Byte Offset: 0x14cd8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH19_DPCM_CLAMP_HIGH_0

Offset: 0x5337
Byte Offset: 0x14cdc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH19_DPCM_CLAMP_LOW_0

Offset: 0x5338
Byte Offset: 0x14ce0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH19_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x533b
Byte Offset: 0x14cec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH19_ATOMP_SURFACE_OFFSET0_0

Offset: 0x533c
Byte Offset: 0x14cf0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH19_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x533d
Byte Offset: 0x14cf4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH19_ATOMP_SURFACE_STRIDE0_0

Offset: 0x533e
Byte Offset: 0x14cf8
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH19_ATOMP_DPCM_CHUNK_0

Offset: 0x533f
Byte Offset: 0x14cfc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH19_ATOMP_SURFACE_OFFSET1_0

Offset: 0x5340
Byte Offset: 0x14d00
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH19_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x5341
Byte Offset: 0x14d04
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH19_ATOMP_SURFACE_STRIDE1_0

Offset: 0x5342
Byte Offset: 0x14d08
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH19_ATOMP_SURFACE_OFFSET2_0

Offset: 0x5343
Byte Offset: 0x14d0c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH19_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x5344
Byte Offset: 0x14d10
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH19_ATOMP_SURFACE_STRIDE2_0

Offset: 0x5345
Byte Offset: 0x14d14
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH19_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x5346
Byte Offset: 0x14d18
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH19_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x5347
Byte Offset: 0x14d1c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH19_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x5348
Byte Offset: 0x14d20
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH19_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x5349
Byte Offset: 0x14d24
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH19_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x534a
Byte Offset: 0x14d28
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH19_ATOMP_HIGH_PRI_REQ_0

Offset: 0x534b
Byte Offset: 0x14d2c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH19_ATOMP_RESERVE_0

Offset: 0x534c
Byte Offset: 0x14d30
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 20 Specific Registers

VI_CH20_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x5400
Byte Offset: 0x15000
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH20_CHANSEL_0

Offset: 0x5404
Byte Offset: 0x15010
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH20_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x5405
Byte Offset: 0x15014
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH20_CONTROL_0

Channel prefix gets added to register name (VI_CH*)

Offset: 0x5406
Byte Offset: 0x15018
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.

Bit	Parity Protection	Reset	Description
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH20_MATCH_0

SHADOWED CHANSEL REGISTERS

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x5407
Byte Offset: 0x1501c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH20_MATCH_DOL_0

Offset: 0x5408
Byte Offset: 0x15020
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH20_MATCH_VC_HI_0

Offset: 0x5409
Byte Offset: 0x15024
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH20_MATCH_DATATYPE_0

Offset: 0x540a
Byte Offset: 0x15028
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH20_MATCH_FRAMEID_0

Offset: 0x540b
Byte Offset: 0x1502c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH20_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT. The one exception is embedded data type which does not get remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on.

You are also likely not to have a wildcard but an exact match for data type. It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x540c
Byte Offset: 0x15030
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.

Bit	Reset	Description
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH20_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x540d
Byte Offset: 0x15034
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH20_FRAME_Y_0

Expected frame y dimension

Offset: 0x540e
Byte Offset: 0x15038
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH20_EMBED_X_0
Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x540f
Byte Offset: 0x1503c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH20_EMBED_Y_0
Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x5410
Byte Offset: 0x15040
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.

Bit	Reset	Description
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH20_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x5411
Byte Offset: 0x15044
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH20_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started. For example starting a flash in anticipation of an upcoming frame, or setup up a

focusing process. This is done setting a tripline count which flags a notification once the number of lines has been reached. The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

The line timer events are not filtered by the cropped region, they are solely based on the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x5412
Byte Offset: 0x15048
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH20_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x5413
Byte Offset: 0x1504c
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH20_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use. If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x5414
Byte Offset: 0x15050
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH20_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x5415
Byte Offset: 0x15054
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH20_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x5416
Byte Offset: 0x15058
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH20_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) - SKIP_Y_LINES

Offset: 0x5417
Byte Offset: 0x1505c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH20_OUT_Y_0

Offset: 0x5418
Byte Offset: 0x15060
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH20_NOTIFY_MASK_0

Offset: 0x5419
Byte Offset: 0x15064
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH20_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED.

The expected use of NOMATCH is for debug purposes. If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification. To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x541a
Byte Offset: 0x15068
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels

Bit	Reset	Description
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH20_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x541b
Byte Offset: 0x1506c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH20_FRAME_COUNT_0

FRAME COUNTS:

FS: a count of all valid frame start seen for channel saturates at 4095

FE: a count of all valid frame end seen for channel (incudling errored frames), saturates at 4095

FAULT: a count of all faulty frame end seen for channel, saturates at 255

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame.

If in frame then FS=FE+1. FE can be perceived as a checksum check for FS. Write any value to register to clear all counts.

Offset: 0x541c
Byte Offset: 0x15070
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH20_LINE_TIMER_FIRST_0

Offset: 0x541d
Byte Offset: 0x15074
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH20_FLUSH_FIRST_0

Offset: 0x541e
Byte Offset: 0x15078
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH20_DOL_CTRL_0

Offset: 0x541f
Byte Offset: 0x1507c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH20_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x5420
Byte Offset: 0x15080
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;

Bit	R/W	Reset	Description
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH20_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x5421
Byte Offset: 0x15084
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH20_PIXFMT_FORMAT_0

Offset: 0x5422
Byte Offset: 0x15088
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/ RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH20_PIXFMT_WIDE_0

Offset: 0x5423
Byte Offset: 0x1508c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH20_PIXFMT_PDAF_CROP_X_0

Offset: 0x5424
Byte Offset: 0x15090
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH20_PIXFMT_PDAF_CROP_Y_0

Offset: 0x5425
Byte Offset: 0x15094
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH20_PIXFMT_PDAF_CONFIG0_0

Offset: 0x5426
Byte Offset: 0x15098
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH20_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x5427
Byte Offset: 0x1509c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends

Bit	Reset	Description
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH20_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x5428
Byte Offset: 0x150a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH20_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.

Offset: 0x5429
Byte Offset: 0x150a4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH20_PIXFMT_PDAF_CONFIG1_0

Offset: 0x542a
Byte Offset: 0x150a8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH20_DPCM_STRIP_0

Offset: 0x542f
Byte Offset: 0x150bc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH20_DPCM_CHUNK_FIRST_0

Offset: 0x5430
Byte Offset: 0x150c0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH20_DPCM_CHUNK_BODY_0

Offset: 0x5431
Byte Offset: 0x150c4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH20_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x5432
Byte Offset: 0x150c8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH20_DPCM_CHUNK_LAST_0

Offset: 0x5433
Byte Offset: 0x150cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH20_DPCM_STATISTICS_0_0

Offset: 0x5434
Byte Offset: 0x150d0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH20_DPCM_STATISTICS_1_0

Offset: 0x5435
Byte Offset: 0x150d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH20_DPCM_MODE_0

Offset: 0x5436
Byte Offset: 0x150d8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH20_DPCM_CLAMP_HIGH_0

Offset: 0x5437
Byte Offset: 0x150dc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH20_DPCM_CLAMP_LOW_0

Offset: 0x5438
Byte Offset: 0x150e0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH20_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x543b
Byte Offset: 0x150ec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH20_ATOMP_SURFACE_OFFSET0_0

Offset: 0x543c
Byte Offset: 0x150f0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH20_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x543d
Byte Offset: 0x150f4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH20_ATOMP_SURFACE_STRIDE0_0

Offset: 0x543e
Byte Offset: 0x150f8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH20_ATOMP_DPCM_CHUNK_0

Offset: 0x543f
Byte Offset: 0x150fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH20_ATOMP_SURFACE_OFFSET1_0

Offset: 0x5440
Byte Offset: 0x15100
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH20_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x5441
Byte Offset: 0x15104
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH20_ATOMP_SURFACE_STRIDE1_0

Offset: 0x5442
Byte Offset: 0x15108
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH20_ATOMP_SURFACE_OFFSET2_0

Offset: 0x5443
Byte Offset: 0x1510c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH20_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x5444
Byte Offset: 0x15110
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH20_ATOMP_SURFACE_STRIDE2_0

Offset: 0x5445
Byte Offset: 0x15114
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH20_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x5446
Byte Offset: 0x15118
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH20_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x5447
Byte Offset: 0x1511c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH20_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x5448
Byte Offset: 0x15120
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH20_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x5449
Byte Offset: 0x15124
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH20_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x544a
Byte Offset: 0x15128
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH20_ATOMP_HIGH_PRI_REQ_0

Offset: 0x544b
Byte Offset: 0x1512c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH20_ATOMP_RESERVE_0

Offset: 0x544c
Byte Offset: 0x15130
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

7.2.2.3.4 Video Input Channels 21-35 Registers

Channel 21 Specific Registers

VI_CH21_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x5500
Byte Offset: 0x15400
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH21_CHANSEL_0

Offset: 0x5504
Byte Offset: 0x15410
Read/Write: R/W

Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH21_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x5505
Byte Offset: 0x15414
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH21_CONTROL_0

Channel prefix gets added to register name (VI_CH*)

Offset: 0x5506
Byte Offset: 0x15418
Read/Write: R/W
Parity Protection: See table below
Shadow: N

SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH21_MATCH_0

SHADOWED CHANSEL REGISTERS

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x5507
Byte Offset: 0x1541c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH21_MATCH_DOL_0

Offset: 0x5508
Byte Offset: 0x15420
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH21_MATCH_VC_HI_0

Offset: 0x5509
Byte Offset: 0x15424
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH21_MATCH_DATATYPE_0

Offset: 0x550a
Byte Offset: 0x15428
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH21_MATCH_FRAMEID_0

Offset: 0x550b
Byte Offset: 0x1542c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH21_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT. The one exception is embedded data type which does not get

remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on.

You are also likely not to have a wildcard but an exact match for data type. It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x550c
Byte Offset: 0x15430
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH21_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x550d
Byte Offset: 0x15434
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH21_FRAME_Y_0

Expected frame y dimension

Offset: 0x550e
Byte Offset: 0x15438
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH21_EMBED_X_0
Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x550f
Byte Offset: 0x1543c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH21_EMBED_Y_0
Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x5510
Byte Offset: 0x15440
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH21_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x5511
Byte Offset: 0x15444
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH21_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started. For example starting a flash in anticipation of an upcoming frame, or setup up a focusing process. This is done setting a tripline count which flags a notification once the number of lines has been reached. The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

The line timer events are not filtered by the cropped region, they are solely based on the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x5512
Byte Offset: 0x15448
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH21_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x5513
Byte Offset: 0x1544c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH21_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use. If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x5514
Byte Offset: 0x15450
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH21_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x5515
Byte Offset: 0x15454
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH21_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x5516
Byte Offset: 0x15458
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH21_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) - SKIP_Y_LINES

Offset: 0x5517
Byte Offset: 0x1545c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH21_OUT_Y_0

Offset: 0x5518
Byte Offset: 0x15460
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH21_NOTIFY_MASK_0

Offset: 0x5519
Byte Offset: 0x15464
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped

Bit	Reset	Description
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH21_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED.

The expected use of NOMATCH is for debug purposes.

If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification. To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x551a
Byte Offset: 0x15468
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).

Bit	Reset	Description
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH21_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x551b
Byte Offset: 0x1546c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded

Bit	Reset	Description
15:0	X	**FRAMEID**: Frame ID

VI_CH21_FRAME_COUNT_0

FRAME COUNTS:

FS: a count of all valid frame start seen for channel saturates at 4095

FE: a count of all valid frame end seen for channel (incudling errored frames), saturates at 4095

FAULT: a count of all faulty frame end seen for channel, saturates at 255

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame.

If in frame then FS=FE+1. FE can be perceived as a checksum check for FS. Write any value to register to clear all counts.

Offset: 0x551c
Byte Offset: 0x15470
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH21_LINE_TIMER_FIRST_0

Offset: 0x551d
Byte Offset: 0x15474
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH21_FLUSH_FIRST_0

Offset: 0x551e
Byte Offset: 0x15478
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH21_DOL_CTRL_0

Offset: 0x551f
Byte Offset: 0x1547c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH21_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x5520
Byte Offset: 0x15480
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH21_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x5521
Byte Offset: 0x15484
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel

Bit	Reset	Description
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH21_PIXFMT_FORMAT_0

Offset: 0x5522
Byte Offset: 0x15488
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/ RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0

Bit	Reset	Description
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH21_PIXFMT_WIDE_0

Offset: 0x5523
Byte Offset: 0x1548c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH21_PIXFMT_PDAF_CROP_X_0

Offset: 0x5524
Byte Offset: 0x15490
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH21_PIXFMT_PDAF_CROP_Y_0

Offset: 0x5525
Byte Offset: 0x15494
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends

Bit	Reset	Description
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH21_PIXFMT_PDAF_CONFIG0_0

Offset: 0x5526
Byte Offset: 0x15498
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH21_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x5527
Byte Offset: 0x1549c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH21_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x5528
Byte Offset: 0x154a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH21_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.

Offset: 0x5529
Byte Offset: 0x154a4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH21_PIXFMT_PDAF_CONFIG1_0

Offset: 0x552a
Byte Offset: 0x154a8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH21_DPCM_STRIP_0

Offset: 0x552f
Byte Offset: 0x154bc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH21_DPCM_CHUNK_FIRST_0

Offset: 0x5530
Byte Offset: 0x154c0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH21_DPCM_CHUNK_BODY_0

Offset: 0x5531
Byte Offset: 0x154c4

Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH21_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x5532
Byte Offset: 0x154c8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH21_DPCM_CHUNK_LAST_0

Offset: 0x5533
Byte Offset: 0x154cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH21_DPCM_STATISTICS_0_0

Offset: 0x5534
Byte Offset: 0x154d0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH21_DPCM_STATISTICS_1_0

Offset: 0x5535
Byte Offset: 0x154d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH21_DPCM_MODE_0

Offset: 0x5536
Byte Offset: 0x154d8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH21_DPCM_CLAMP_HIGH_0

Offset: 0x5537
Byte Offset: 0x154dc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH21_DPCM_CLAMP_LOW_0

Offset: 0x5538
Byte Offset: 0x154e0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH21_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x553b
Byte Offset: 0x154ec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH21_ATOMP_SURFACE_OFFSET0_0

Offset: 0x553c
Byte Offset: 0x154f0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH21_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x553d
Byte Offset: 0x154f4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH21_ATOMP_SURFACE_STRIDE0_0

Offset: 0x553e
Byte Offset: 0x154f8
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH21_ATOMP_DPCM_CHUNK_0

Offset: 0x553f
Byte Offset: 0x154fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH21_ATOMP_SURFACE_OFFSET1_0

Offset: 0x5540
Byte Offset: 0x15500
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH21_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x5541
Byte Offset: 0x15504
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH21_ATOMP_SURFACE_STRIDE1_0

Offset: 0x5542
Byte Offset: 0x15508
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH21_ATOMP_SURFACE_OFFSET2_0

Offset: 0x5543
Byte Offset: 0x1550c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH21_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x5544
Byte Offset: 0x15510
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH21_ATOMP_SURFACE_STRIDE2_0

Offset: 0x5545
Byte Offset: 0x15514
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH21_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x5546
Byte Offset: 0x15518
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH21_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x5547
Byte Offset: 0x1551c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH21_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x5548
Byte Offset: 0x15520
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH21_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x5549
Byte Offset: 0x15524
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH21_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x554a
Byte Offset: 0x15528
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH21_ATOMP_HIGH_PRI_REQ_0

Offset: 0x554b
Byte Offset: 0x1552c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH21_ATOMP_RESERVE_0

Offset: 0x554c
Byte Offset: 0x15530
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 22 Specific Registers

VI_CH22_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x5600
Byte Offset: 0x15800
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH22_CHANSEL_0

Offset: 0x5604
Byte Offset: 0x15810
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH22_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x5605
Byte Offset: 0x15814
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH22_CONTROL_0

Channel prefix gets added to register name (VI_CH*)

Offset: 0x5606
Byte Offset: 0x15818
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.

Bit	Parity Protection	Reset	Description
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH22_MATCH_0

SHADOWED CHANSEL REGISTERS

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x5607
Byte Offset: 0x1581c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH22_MATCH_DOL_0

Offset: 0x5608
Byte Offset: 0x15820
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH22_MATCH_VC_HI_0

Offset: 0x5609
Byte Offset: 0x15824
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH22_MATCH_DATATYPE_0

Offset: 0x560a
Byte Offset: 0x15828
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH22_MATCH_FRAMEID_0

Offset: 0x560b
Byte Offset: 0x1582c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH22_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT. The one exception is embedded data type which does not get remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on.

You are also likely not to have a wildcard but an exact match for data type. It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x560c
Byte Offset: 0x15830
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.

Bit	Reset	Description
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH22_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x560d
Byte Offset: 0x15834
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH22_FRAME_Y_0

Expected frame y dimension

Offset: 0x560e
Byte Offset: 0x15838
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH22_EMBED_X_0
Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x560f
Byte Offset: 0x1583c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH22_EMBED_Y_0
Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x5610
Byte Offset: 0x15840
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.

Bit	Reset	Description
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH22_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x5611
Byte Offset: 0x15844
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH22_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started. For example starting a flash in anticipation of an upcoming frame, or setup up a

focusing process. This is done setting a tripline count which flags a notification once the number of lines has been reached. The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

The line timer events are not filtered by the cropped region, they are solely based on the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x5612
Byte Offset: 0x15848
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH22_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x5613
Byte Offset: 0x1584c
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH22_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use. If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x5614
Byte Offset: 0x15850
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH22_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x5615
Byte Offset: 0x15854
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH22_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x5616
Byte Offset: 0x15858
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH22_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) - SKIP_Y_LINES

Offset: 0x5617
Byte Offset: 0x1585c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH22_OUT_Y_0

Offset: 0x5618
Byte Offset: 0x15860
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH22_NOTIFY_MASK_0

Offset: 0x5619
Byte Offset: 0x15864
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH22_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED.The expected use of NOMATCH is for debug purposes.

If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification.

To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x561a
Byte Offset: 0x15868
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels

Bit	Reset	Description
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH22_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x561b
Byte Offset: 0x1586c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH22_FRAME_COUNT_0

FRAME COUNTS:

FS: a count of all valid frame start seen for channel saturates at 4095

FE: a count of all valid frame end seen for channel (incudling errored frames), saturates at 4095

FAULT: a count of all faulty frame end seen for channel, saturates at 255

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame. If in frame then FS=FE+1. FE can be perceived as a checksum check for FS.

Write any value to register to clear all counts.

Offset: 0x561c
Byte Offset: 0x15870
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH22_LINE_TIMER_FIRST_0

Offset: 0x561d
Byte Offset: 0x15874
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH22_FLUSH_FIRST_0

Offset: 0x561e
Byte Offset: 0x15878
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH22_DOL_CTRL_0

Offset: 0x561f
Byte Offset: 0x1587c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH22_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x5620
Byte Offset: 0x15880
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;

Bit	R/W	Reset	Description
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH22_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x5621
Byte Offset: 0x15884
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH22_PIXFMT_FORMAT_0

Offset: 0x5622
Byte Offset: 0x15888
Read/Write: R/W
Parity Protection: Y

Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH22_PIXFMT_WIDE_0

Offset: 0x5623
Byte Offset: 0x1588c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH22_PIXFMT_PDAF_CROP_X_0

Offset: 0x5624
Byte Offset: 0x15890
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH22_PIXFMT_PDAF_CROP_Y_0

Offset: 0x5625
Byte Offset: 0x15894
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH22_PIXFMT_PDAF_CONFIG0_0

Offset: 0x5626
Byte Offset: 0x15898
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH22_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x5627
Byte Offset: 0x1589c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends

Bit	Reset	Description
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH22_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x5628
Byte Offset: 0x158a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH22_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.
Offset: 0x5629
Byte Offset: 0x158a4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH22_PIXFMT_PDAF_CONFIG1_0

Offset: 0x562a
Byte Offset: 0x158a8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH22_DPCM_STRIP_0

Offset: 0x562f
Byte Offset: 0x158bc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH22_DPCM_CHUNK_FIRST_0

Offset: 0x5630
Byte Offset: 0x158c0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH22_DPCM_CHUNK_BODY_0

Offset: 0x5631
Byte Offset: 0x158c4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH22_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x5632
Byte Offset: 0x158c8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH22_DPCM_CHUNK_LAST_0

Offset: 0x5633
Byte Offset: 0x158cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH22_DPCM_STATISTICS_0_0

Offset: 0x5634
Byte Offset: 0x158d0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH22_DPCM_STATISTICS_1_0

Offset: 0x5635
Byte Offset: 0x158d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH22_DPCM_MODE_0

Offset: 0x5636
Byte Offset: 0x158d8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH22_DPCM_CLAMP_HIGH_0

Offset: 0x5637
Byte Offset: 0x158dc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH22_DPCM_CLAMP_LOW_0

Offset: 0x5638
Byte Offset: 0x158e0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH22_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x563b
Byte Offset: 0x158ec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH22_ATOMP_SURFACE_OFFSET0_0

Offset: 0x563c
Byte Offset: 0x158f0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH22_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x563d
Byte Offset: 0x158f4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH22_ATOMP_SURFACE_STRIDE0_0

Offset: 0x563e
Byte Offset: 0x158f8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH22_ATOMP_DPCM_CHUNK_0

Offset: 0x563f
Byte Offset: 0x158fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH22_ATOMP_SURFACE_OFFSET1_0

Offset: 0x5640
Byte Offset: 0x15900
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH22_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x5641
Byte Offset: 0x15904
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH22_ATOMP_SURFACE_STRIDE1_0

Offset: 0x5642
Byte Offset: 0x15908
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH22_ATOMP_SURFACE_OFFSET2_0

Offset: 0x5643
Byte Offset: 0x1590c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH22_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x5644
Byte Offset: 0x15910
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH22_ATOMP_SURFACE_STRIDE2_0

Offset: 0x5645
Byte Offset: 0x15914
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH22_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x5646
Byte Offset: 0x15918
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH22_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x5647
Byte Offset: 0x1591c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH22_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x5648
Byte Offset: 0x15920
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH22_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x5649
Byte Offset: 0x15924
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH22_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x564a
Byte Offset: 0x15928
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH22_ATOMP_HIGH_PRI_REQ_0

Offset: 0x564b
Byte Offset: 0x1592c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH22_ATOMP_RESERVE_0

Offset: 0x564c
Byte Offset: 0x15930
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 23 Specific Registers

VI_CH23_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x5700
Byte Offset: 0x15c00
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH23_CHANSEL_0

Offset: 0x5704
Byte Offset: 0x15c10
Read/Write: R/W
Parity Protection: N
Shadow: Y

SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH23_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x5705
Byte Offset: 0x15c14
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH23_CONTROL_0

Channel prefix gets added to register name (VI_CH*)

Offset: 0x5706
Byte Offset: 0x15c18
Read/Write: R/W
Parity Protection: See table below
Shadow: N

SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH23_MATCH_0

SHADOWED CHANSEL REGISTERS

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x5707
Byte Offset: 0x15c1c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH23_MATCH_DOL_0

Offset: 0x5708
Byte Offset: 0x15c20
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH23_MATCH_VC_HI_0

Offset: 0x5709
Byte Offset: 0x15c24
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH23_MATCH_DATATYPE_0

Offset: 0x570a
Byte Offset: 0x15c28
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH23_MATCH_FRAMEID_0

Offset: 0x570b
Byte Offset: 0x15c2c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH23_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT. The one exception is embedded data type which does not get

remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on.

You are also likely not to have a wildcard but an exact match for data type. It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x570c
Byte Offset: 0x15c30
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH23_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x570d
Byte Offset: 0x15c34
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH23_FRAME_Y_0

Expected frame y dimension

Offset: 0x570e
Byte Offset: 0x15c38
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH23_EMBED_X_0
Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x570f
Byte Offset: 0x15c3c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH23_EMBED_Y_0
Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x5710
Byte Offset: 0x15c40
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH23_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x5711
Byte Offset: 0x15c44
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH23_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started. For example starting a flash in anticipation of an upcoming frame, or setup up a focusing process. This is done setting a tripline count which flags a notification once the number of lines has been reached. The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

The line timer events are not filtered by the cropped region, they are solely based on the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x5712
Byte Offset: 0x15c48
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH23_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x5713
Byte Offset: 0x15c4c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH23_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use. If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x5714
Byte Offset: 0x15c50
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH23_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x5715
Byte Offset: 0x15c54
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH23_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x5716
Byte Offset: 0x15c58
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH23_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) - SKIP_Y_LINES

Offset: 0x5717
Byte Offset: 0x15c5c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH23_OUT_Y_0

Offset: 0x5718
Byte Offset: 0x15c60
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH23_NOTIFY_MASK_0

Offset: 0x5719
Byte Offset: 0x15c64
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped

Bit	Reset	Description
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH23_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED.

The expected use of NOMATCH is for debug purposes.

If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification.

To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x571a
Byte Offset: 0x15c68
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).

Bit	Reset	Description
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH23_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x571b
Byte Offset: 0x15c6c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded

Bit	Reset	Description
15:0	X	**FRAMEID**: Frame ID

VI_CH23_FRAME_COUNT_0

FRAME COUNTS:

FS: a count of all valid frame start seen for channel saturates at 4095

FE: a count of all valid frame end seen for channel (incudling errored frames), saturates at 4095

FAULT: a count of all faulty frame end seen for channel, saturates at 255

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame.

If in frame then FS=FE+1. FE can be perceived as a checksum check for FS. Write any value to register to clear all counts.

Offset: 0x571c
Byte Offset: 0x15c70
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH23_LINE_TIMER_FIRST_0

Offset: 0x571d
Byte Offset: 0x15c74
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH23_FLUSH_FIRST_0

Offset: 0x571e
Byte Offset: 0x15c78
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH23_DOL_CTRL_0

Offset: 0x571f
Byte Offset: 0x15c7c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH23_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x5720
Byte Offset: 0x15c80
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH23_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x5721
Byte Offset: 0x15c84
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel

Bit	Reset	Description
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH23_PIXFMT_FORMAT_0

Offset: 0x5722
Byte Offset: 0x15c88
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/ RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0

Bit	Reset	Description
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH23_PIXFMT_WIDE_0

Offset: 0x5723
Byte Offset: 0x15c8c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH23_PIXFMT_PDAF_CROP_X_0

Offset: 0x5724
Byte Offset: 0x15c90
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH23_PIXFMT_PDAF_CROP_Y_0

Offset: 0x5725
Byte Offset: 0x15c94
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends

Bit	Reset	Description
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH23_PIXFMT_PDAF_CONFIG0_0

Offset: 0x5726
Byte Offset: 0x15c98
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH23_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x5727
Byte Offset: 0x15c9c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH23_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x5728
Byte Offset: 0x15ca0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH23_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.

Offset: 0x5729
Byte Offset: 0x15ca4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH23_PIXFMT_PDAF_CONFIG1_0

Offset: 0x572a
Byte Offset: 0x15ca8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH23_DPCM_STRIP_0

Offset: 0x572f
Byte Offset: 0x15cbc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH23_DPCM_CHUNK_FIRST_0

Offset: 0x5730
Byte Offset: 0x15cc0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH23_DPCM_CHUNK_BODY_0

Offset: 0x5731
Byte Offset: 0x15cc4

Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH23_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x5732
Byte Offset: 0x15cc8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH23_DPCM_CHUNK_LAST_0

Offset: 0x5733
Byte Offset: 0x15ccc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH23_DPCM_STATISTICS_0_0

Offset: 0x5734
Byte Offset: 0x15cd0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH23_DPCM_STATISTICS_1_0

Offset: 0x5735
Byte Offset: 0x15cd4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH23_DPCM_MODE_0

Offset: 0x5736
Byte Offset: 0x15cd8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH23_DPCM_CLAMP_HIGH_0

Offset: 0x5737
Byte Offset: 0x15cdc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH23_DPCM_CLAMP_LOW_0

Offset: 0x5738
Byte Offset: 0x15ce0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH23_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x573b
Byte Offset: 0x15cec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH23_ATOMP_SURFACE_OFFSET0_0

Offset: 0x573c
Byte Offset: 0x15cf0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH23_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x573d
Byte Offset: 0x15cf4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH23_ATOMP_SURFACE_STRIDE0_0

Offset: 0x573e
Byte Offset: 0x15cf8
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH23_ATOMP_DPCM_CHUNK_0

Offset: 0x573f
Byte Offset: 0x15cfc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH23_ATOMP_SURFACE_OFFSET1_0

Offset: 0x5740
Byte Offset: 0x15d00
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH23_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x5741
Byte Offset: 0x15d04
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH23_ATOMP_SURFACE_STRIDE1_0

Offset: 0x5742
Byte Offset: 0x15d08
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH23_ATOMP_SURFACE_OFFSET2_0

Offset: 0x5743
Byte Offset: 0x15d0c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH23_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x5744
Byte Offset: 0x15d10
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH23_ATOMP_SURFACE_STRIDE2_0

Offset: 0x5745
Byte Offset: 0x15d14
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH23_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x5746
Byte Offset: 0x15d18
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH23_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x5747
Byte Offset: 0x15d1c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH23_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x5748
Byte Offset: 0x15d20
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH23_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x5749
Byte Offset: 0x15d24
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH23_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x574a
Byte Offset: 0x15d28
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH23_ATOMP_HIGH_PRI_REQ_0

Offset: 0x574b
Byte Offset: 0x15d2c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH23_ATOMP_RESERVE_0

Offset: 0x574c
Byte Offset: 0x15d30
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 24 Specific Registers

VI_CH24_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x5800
Byte Offset: 0x16000
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH24_CHANSEL_0

Offset: 0x5804
Byte Offset: 0x16010
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH24_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x5805
Byte Offset: 0x16014
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH24_CONTROL_0

Channel prefix gets added to register name (VI_CH*)
Offset: 0x5806
Byte Offset: 0x16018
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.

Bit	Parity Protection	Reset	Description
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH24_MATCH_0

SHADOWED CHANSEL REGISTERS

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x5807
Byte Offset: 0x1601c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH24_MATCH_DOL_0

Offset: 0x5808
Byte Offset: 0x16020
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH24_MATCH_VC_HI_0

Offset: 0x5809
Byte Offset: 0x16024
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH24_MATCH_DATATYPE_0

Offset: 0x580a
Byte Offset: 0x16028
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH24_MATCH_FRAMEID_0

Offset: 0x580b
Byte Offset: 0x1602c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH24_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT. The one exception is embedded data type which does not get remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on.

You are also likely not to have a wildcard but an exact match for data type. It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x580c
Byte Offset: 0x16030
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.

Bit	Reset	Description
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH24_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x580d
Byte Offset: 0x16034
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH24_FRAME_Y_0

Expected frame y dimension

Offset: 0x580e
Byte Offset: 0x16038
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH24_EMBED_X_0
Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x580f
Byte Offset: 0x1603c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH24_EMBED_Y_0
Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x5810
Byte Offset: 0x16040
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.

Bit	Reset	Description
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH24_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data.

This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x5811
Byte Offset: 0x16044
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH24_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started.

For example starting a flash in anticipation of an upcoming frame, or setup up a focusing process.

This is done setting a tripline count which flags a notification once the number of lines has been reached.

The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

The line timer events are not filtered by the cropped region, they are solely based on the input frame.

The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x5812
Byte Offset: 0x16048
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH24_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x5813
Byte Offset: 0x1604c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH24_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use.

If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x5814
Byte Offset: 0x16050
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH24_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x5815
Byte Offset: 0x16054
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH24_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x5816
Byte Offset: 0x16058
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH24_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) - SKIP_Y_LINES

Offset: 0x5817
Byte Offset: 0x1605c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH24_OUT_Y_0

Offset: 0x5818
Byte Offset: 0x16060
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH24_NOTIFY_MASK_0

Offset: 0x5819
Byte Offset: 0x16064
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped

Bit	Reset	Description
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH24_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED.

The expected use of NOMATCH is for debug purposes.

If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification. To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x581a
Byte Offset: 0x16068
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).

Bit	Reset	Description
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH24_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x581b
Byte Offset: 0x1606c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded

Bit	Reset	Description
15:0	X	**FRAMEID**: Frame ID

VI_CH24_FRAME_COUNT_0

FRAME COUNTS:

FS: a count of all valid frame start seen for channel saturates at 4095

FE: a count of all valid frame end seen for channel (incudling errored frames), saturates at 4095

FAULT: a count of all faulty frame end seen for channel, saturates at 255

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame. If in frame then FS=FE+1. FE can be perceived as a checksum check for FS.

Write any value to register to clear all counts.

Offset: 0x581c
Byte Offset: 0x16070
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH24_LINE_TIMER_FIRST_0

Offset: 0x581d
Byte Offset: 0x16074
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH24_FLUSH_FIRST_0

Offset: 0x581e
Byte Offset: 0x16078
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH24_DOL_CTRL_0

Offset: 0x581f
Byte Offset: 0x1607c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH24_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x5820
Byte Offset: 0x16080
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH24_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x5821
Byte Offset: 0x16084
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel

Bit	Reset	Description
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH24_PIXFMT_FORMAT_0

Offset: 0x5822
Byte Offset: 0x16088
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/ RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0

Bit	Reset	Description
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH24_PIXFMT_WIDE_0

Offset: 0x5823
Byte Offset: 0x1608c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH24_PIXFMT_PDAF_CROP_X_0

Offset: 0x5824
Byte Offset: 0x16090
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH24_PIXFMT_PDAF_CROP_Y_0

Offset: 0x5825
Byte Offset: 0x16094
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends

Bit	Reset	Description
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH24_PIXFMT_PDAF_CONFIG0_0

Offset: 0x5826
Byte Offset: 0x16098
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH24_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x5827
Byte Offset: 0x1609c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH24_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x5828
Byte Offset: 0x160a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH24_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.

Offset: 0x5829
Byte Offset: 0x160a4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH24_PIXFMT_PDAF_CONFIG1_0

Offset: 0x582a
Byte Offset: 0x160a8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH24_DPCM_STRIP_0

Offset: 0x582f
Byte Offset: 0x160bc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH24_DPCM_CHUNK_FIRST_0

Offset: 0x5830
Byte Offset: 0x160c0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH24_DPCM_CHUNK_BODY_0

Offset: 0x5831
Byte Offset: 0x160c4

Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH24_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x5832
Byte Offset: 0x160c8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH24_DPCM_CHUNK_LAST_0

Offset: 0x5833
Byte Offset: 0x160cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH24_DPCM_STATISTICS_0_0

Offset: 0x5834
Byte Offset: 0x160d0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH24_DPCM_STATISTICS_1_0

Offset: 0x5835
Byte Offset: 0x160d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH24_DPCM_MODE_0

Offset: 0x5836
Byte Offset: 0x160d8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH24_DPCM_CLAMP_HIGH_0

Offset: 0x5837
Byte Offset: 0x160dc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH24_DPCM_CLAMP_LOW_0

Offset: 0x5838
Byte Offset: 0x160e0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH24_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x583b
Byte Offset: 0x160ec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH24_ATOMP_SURFACE_OFFSET0_0

Offset: 0x583c
Byte Offset: 0x160f0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH24_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x583d
Byte Offset: 0x160f4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH24_ATOMP_SURFACE_STRIDE0_0

Offset: 0x583e
Byte Offset: 0x160f8
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH24_ATOMP_DPCM_CHUNK_0

Offset: 0x583f
Byte Offset: 0x160fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH24_ATOMP_SURFACE_OFFSET1_0

Offset: 0x5840
Byte Offset: 0x16100
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH24_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x5841
Byte Offset: 0x16104
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH24_ATOMP_SURFACE_STRIDE1_0

Offset: 0x5842
Byte Offset: 0x16108
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH24_ATOMP_SURFACE_OFFSET2_0

Offset: 0x5843
Byte Offset: 0x1610c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH24_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x5844
Byte Offset: 0x16110
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH24_ATOMP_SURFACE_STRIDE2_0

Offset: 0x5845
Byte Offset: 0x16114
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH24_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x5846
Byte Offset: 0x16118
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH24_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x5847
Byte Offset: 0x1611c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH24_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x5848
Byte Offset: 0x16120
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH24_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x5849
Byte Offset: 0x16124
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH24_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x584a
Byte Offset: 0x16128
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH24_ATOMP_HIGH_PRI_REQ_0

Offset: 0x584b
Byte Offset: 0x1612c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH24_ATOMP_RESERVE_0

Offset: 0x584c
Byte Offset: 0x16130
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 25 Specific Registers

VI_CH25_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x5900
Byte Offset: 0x16400
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH25_CHANSEL_0

Offset: 0x5904
Byte Offset: 0x16410
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH25_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x5905
Byte Offset: 0x16414
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH25_CONTROL_0

Channel prefix gets added to register name (VI_CH*)

Offset: 0x5906
Byte Offset: 0x16418
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.

Bit	Parity Protection	Reset	Description
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH25_MATCH_0

SHADOWED CHANSEL REGISTERS

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x5907
Byte Offset: 0x1641c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH25_MATCH_DOL_0

Offset: 0x5908
Byte Offset: 0x16420
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH25_MATCH_VC_HI_0

Offset: 0x5909
Byte Offset: 0x16424
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH25_MATCH_DATATYPE_0

Offset: 0x590a
Byte Offset: 0x16428
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH25_MATCH_FRAMEID_0

Offset: 0x590b
Byte Offset: 0x1642c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH25_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT. The one exception is embedded data type which does not get remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on. You are also likely not to have a wildcard but an exact match for data type. It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x590c
Byte Offset: 0x16430
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.

0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH25_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x590d
Byte Offset: 0x16434
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH25_FRAME_Y_0

Expected frame y dimension

Offset: 0x590e
Byte Offset: 0x16438
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH25_EMBED_X_0

Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x590f
Byte Offset: 0x1643c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH25_EMBED_Y_0
Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x5910
Byte Offset: 0x16440
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH25_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x5911
Byte Offset: 0x16444
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH25_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started. For example starting a flash in anticipation of an upcoming frame, or setup up a focusing process. This is done setting a tripline count which flags a notification once the number of lines has been reached. The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

The line timer events are not filtered by the cropped region, they are solely based on the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x5912
Byte Offset: 0x16448
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH25_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x5913
Byte Offset: 0x1644c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH25_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use. If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x5914
Byte Offset: 0x16450
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH25_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x5915
Byte Offset: 0x16454
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH25_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x5916
Byte Offset: 0x16458
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH25_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) - SKIP_Y_LINES

Offset: 0x5917
Byte Offset: 0x1645c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH25_OUT_Y_0

Offset: 0x5918
Byte Offset: 0x16460
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH25_NOTIFY_MASK_0

Offset: 0x5919
Byte Offset: 0x16464
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH25_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED. The expected use of NOMATCH is for debug purposes.

If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification.

To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x591a
Byte Offset: 0x16468
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH25_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x591b
Byte Offset: 0x1646c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH25_FRAME_COUNT_0

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame.

If in frame then FS=FE+1. FE can be perceived as a checksum check for FS.

Write any value to register to clear all counts.

Offset: 0x591c
Byte Offset: 0x16470
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH25_LINE_TIMER_FIRST_0

Offset: 0x591d
Byte Offset: 0x16474
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH25_FLUSH_FIRST_0

Offset: 0x591e
Byte Offset: 0x16478
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.

Bit	Reset	Description
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH25_DOL_CTRL_0

Offset: 0x591f
Byte Offset: 0x1647c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH25_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x5920
Byte Offset: 0x16480
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

Bit	R/W	Reset	Description
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH25_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x5921
Byte Offset: 0x16484
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH25_PIXFMT_FORMAT_0

Offset: 0x5922
Byte Offset: 0x16488
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/ RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH25_PIXFMT_WIDE_0

Offset: 0x5923
Byte Offset: 0x1648c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH25_PIXFMT_PDAF_CROP_X_0

Offset: 0x5924
Byte Offset: 0x16490
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH25_PIXFMT_PDAF_CROP_Y_0

Offset: 0x5925
Byte Offset: 0x16494
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH25_PIXFMT_PDAF_CONFIG0_0

Offset: 0x5926
Byte Offset: 0x16498
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH25_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x5927
Byte Offset: 0x1649c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH25_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x5928
Byte Offset: 0x164a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH25_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.

Offset: 0x5929
Byte Offset: 0x164a4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH25_PIXFMT_PDAF_CONFIG1_0

Offset: 0x592a
Byte Offset: 0x164a8
Read/Write: R/W
Parity Protection: Y

Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH25_DPCM_STRIP_0

Offset: 0x592f
Byte Offset: 0x164bc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH25_DPCM_CHUNK_FIRST_0

Offset: 0x5930
Byte Offset: 0x164c0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH25_DPCM_CHUNK_BODY_0

Offset: 0x5931
Byte Offset: 0x164c4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH25_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x5932
Byte Offset: 0x164c8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH25_DPCM_CHUNK_LAST_0

Offset: 0x5933
Byte Offset: 0x164cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH25_DPCM_STATISTICS_0_0

Offset: 0x5934
Byte Offset: 0x164d0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH25_DPCM_STATISTICS_1_0

Offset: 0x5935
Byte Offset: 0x164d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH25_DPCM_MODE_0

Offset: 0x5936
Byte Offset: 0x164d8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH25_DPCM_CLAMP_HIGH_0

Offset: 0x5937
Byte Offset: 0x164dc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH25_DPCM_CLAMP_LOW_0

Offset: 0x5938
Byte Offset: 0x164e0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH25_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x593b
Byte Offset: 0x164ec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH25_ATOMP_SURFACE_OFFSET0_0

Offset: 0x593c
Byte Offset: 0x164f0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH25_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x593d
Byte Offset: 0x164f4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH25_ATOMP_SURFACE_STRIDE0_0

Offset: 0x593e
Byte Offset: 0x164f8
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH25_ATOMP_DPCM_CHUNK_0

Offset: 0x593f
Byte Offset: 0x164fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH25_ATOMP_SURFACE_OFFSET1_0

Offset: 0x5940
Byte Offset: 0x16500
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH25_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x5941
Byte Offset: 0x16504
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH25_ATOMP_SURFACE_STRIDE1_0 Offset: 0x5942
Byte Offset: 0x16508
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH25_ATOMP_SURFACE_OFFSET2_0

Offset: 0x5943
Byte Offset: 0x1650c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH25_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x5944
Byte Offset: 0x16510
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH25_ATOMP_SURFACE_STRIDE2_0

Offset: 0x5945
Byte Offset: 0x16514
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH25_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x5946
Byte Offset: 0x16518
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH25_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x5947
Byte Offset: 0x1651c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH25_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x5948
Byte Offset: 0x16520
Read/Write: R/W
Parity Protection: Y
Shadow: YSCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH25_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x5949
Byte Offset: 0x16524
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH25_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x594a
Byte Offset: 0x16528
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH25_ATOMP_HIGH_PRI_REQ_0

Offset: 0x594b
Byte Offset: 0x1652c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH25_ATOMP_RESERVE_0

Offset: 0x594c
Byte Offset: 0x16530
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 26 Specific Registers

VI_CH26_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x5a00
Byte Offset: 0x16800
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH26_CHANSEL_0

Offset: 0x5a04
Byte Offset: 0x16810
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH26_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x5a05
Byte Offset: 0x16814
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH26_CONTROL_0

Channel prefix gets added to register name (VI_CH*)

Offset: 0x5a06
Byte Offset: 0x16818
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.

4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH26_MATCH_0

SHADOWED CHANSEL REGISTERS

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x5a07
Byte Offset: 0x1681c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH26_MATCH_DOL_0

Offset: 0x5a08
Byte Offset: 0x16820
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH26_MATCH_VC_HI_0

Offset: 0x5a09
Byte Offset: 0x16824
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH26_MATCH_DATATYPE_0

Offset: 0x5a0a
Byte Offset: 0x16828
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH26_MATCH_FRAMEID_0

Offset: 0x5a0b
Byte Offset: 0x1682c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH26_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT. The one exception is embedded data type which does not get remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on.

You are also likely not to have a wildcard but an exact match for data type. It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x5a0c
Byte Offset: 0x16830
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.

Bit	Reset	Description
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH26_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x5a0d
Byte Offset: 0x16834
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH26_FRAME_Y_0

Expected frame y dimension

Offset: 0x5a0e
Byte Offset: 0x16838
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH26_EMBED_X_0
Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x5a0f
Byte Offset: 0x1683c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH26_EMBED_Y_0
Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x5a10
Byte Offset: 0x16840
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.

Bit	Reset	Description
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH26_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x5a11
Byte Offset: 0x16844
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH26_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started. For example starting a flash in anticipation of an upcoming frame, or setup up a focusing process. This is done setting a tripline count which flags a notification once the number

of lines has been reached. The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. The line timer events are not filtered by the cropped region, they are solely based on the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x5a12
Byte Offset: 0x16848
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH26_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x5a13
Byte Offset: 0x1684c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH26_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use. If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x5a14
Byte Offset: 0x16850
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH26_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x5a15
Byte Offset: 0x16854
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH26_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x5a16
Byte Offset: 0x16858
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH26_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) -   SKIP_Y_LINES

Offset: 0x5a17
Byte Offset: 0x1685c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH26_OUT_Y_0

Offset: 0x5a18
Byte Offset: 0x16860
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH26_NOTIFY_MASK_0

Offset: 0x5a19
Byte Offset: 0x16864
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH26_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED.

The expected use of NOMATCH is for debug purposes.

If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification. To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x5a1a
Byte Offset: 0x16868
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels

Bit	Reset	Description
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH26_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x5a1b
Byte Offset: 0x1686c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH26_FRAME_COUNT_0

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame.

If in frame then FS=FE+1. FE can be perceived as a checksum check for FS.

Write any value to register to clear all counts.

Offset: 0x5a1c
Byte Offset: 0x16870
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH26_LINE_TIMER_FIRST_0

Offset: 0x5a1d
Byte Offset: 0x16874
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH26_FLUSH_FIRST_0

Offset: 0x5a1e
Byte Offset: 0x16878
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH26_DOL_CTRL_0

Offset: 0x5a1f
Byte Offset: 0x1687c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH26_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x5a20
Byte Offset: 0x16880
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;

Bit	R/W	Reset	Description
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH26_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x5a21
Byte Offset: 0x16884
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH26_PIXFMT_FORMAT_0

Offset: 0x5a22
Byte Offset: 0x16888
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/ RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH26_PIXFMT_WIDE_0

Offset: 0x5a23
Byte Offset: 0x1688c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH26_PIXFMT_PDAF_CROP_X_0

Offset: 0x5a24
Byte Offset: 0x16890
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH26_PIXFMT_PDAF_CROP_Y_0

Offset: 0x5a25
Byte Offset: 0x16894
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH26_PIXFMT_PDAF_CONFIG0_0

Offset: 0x5a26
Byte Offset: 0x16898
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH26_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x5a27
Byte Offset: 0x1689c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends

Bit	Reset	Description
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH26_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x5a28
Byte Offset: 0x168a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH26_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.

Offset: 0x5a29
Byte Offset: 0x168a4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH26_PIXFMT_PDAF_CONFIG1_0

Offset: 0x5a2a
Byte Offset: 0x168a8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH26_DPCM_STRIP_0

Offset: 0x5a2f
Byte Offset: 0x168bc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH26_DPCM_CHUNK_FIRST_0

Offset: 0x5a30
Byte Offset: 0x168c0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH26_DPCM_CHUNK_BODY_0

Offset: 0x5a31
Byte Offset: 0x168c4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH26_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x5a32
Byte Offset: 0x168c8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH26_DPCM_CHUNK_LAST_0

Offset: 0x5a33
Byte Offset: 0x168cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH26_DPCM_STATISTICS_0_0

Offset: 0x5a34
Byte Offset: 0x168d0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH26_DPCM_STATISTICS_1_0

Offset: 0x5a35
Byte Offset: 0x168d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH26_DPCM_MODE_0

Offset: 0x5a36
Byte Offset: 0x168d8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH26_DPCM_CLAMP_HIGH_0

Offset: 0x5a37
Byte Offset: 0x168dc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH26_DPCM_CLAMP_LOW_0

Offset: 0x5a38
Byte Offset: 0x168e0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH26_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x5a3b
Byte Offset: 0x168ec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH26_ATOMP_SURFACE_OFFSET0_0

Offset: 0x5a3c
Byte Offset: 0x168f0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH26_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x5a3d
Byte Offset: 0x168f4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH26_ATOMP_SURFACE_STRIDE0_0

Offset: 0x5a3e
Byte Offset: 0x168f8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH26_ATOMP_DPCM_CHUNK_0

Offset: 0x5a3f
Byte Offset: 0x168fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH26_ATOMP_SURFACE_OFFSET1_0

Offset: 0x5a40
Byte Offset: 0x16900
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH26_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x5a41
Byte Offset: 0x16904
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH26_ATOMP_SURFACE_STRIDE1_0

Offset: 0x5a42
Byte Offset: 0x16908
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH26_ATOMP_SURFACE_OFFSET2_0

Offset: 0x5a43
Byte Offset: 0x1690c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH26_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x5a44
Byte Offset: 0x16910
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH26_ATOMP_SURFACE_STRIDE2_0

Offset: 0x5a45
Byte Offset: 0x16914
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH26_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x5a46
Byte Offset: 0x16918
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH26_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x5a47
Byte Offset: 0x1691c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH26_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x5a48
Byte Offset: 0x16920
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH26_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x5a49
Byte Offset: 0x16924
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH26_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x5a4a
Byte Offset: 0x16928
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH26_ATOMP_HIGH_PRI_REQ_0

Offset: 0x5a4b
Byte Offset: 0x1692c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH26_ATOMP_RESERVE_0

Offset: 0x5a4c
Byte Offset: 0x16930
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 27 Specific Registers

VI_CH27_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x5b00
Byte Offset: 0x16c00
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH27_CHANSEL_0

Offset: 0x5b04
Byte Offset: 0x16c10
Read/Write: R/W
Parity Protection: N
Shadow: Y

SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH27_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x5b05
Byte Offset: 0x16c14
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH27_CONTROL_0

Channel prefix gets added to register name (VI_CH*)

Offset: 0x5b06
Byte Offset: 0x16c18
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH27_MATCH_0

SHADOWED CHANSEL REGISTERS

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x5b07
Byte Offset: 0x16c1c
Read/Write: R/W
Parity Protection: Y

Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH27_MATCH_DOL_0

Offset: 0x5b08
Byte Offset: 0x16c20
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH27_MATCH_VC_HI_0

Offset: 0x5b09
Byte Offset: 0x16c24
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH27_MATCH_DATATYPE_0

Offset: 0x5b0a
Byte Offset: 0x16c28
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH27_MATCH_FRAMEID_0

Offset: 0x5b0b
Byte Offset: 0x16c2c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH27_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT. The one exception is embedded data type which does not get remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on.

You are also likely not to have a wildcard but an exact match for data type. It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x5b0c
Byte Offset: 0x16c30
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH27_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x5b0d
Byte Offset: 0x16c34
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH27_FRAME_Y_0

Expected frame y dimension

Offset: 0x5b0e
Byte Offset: 0x16c38
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH27_EMBED_X_0
Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x5b0f
Byte Offset: 0x16c3c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH27_EMBED_Y_0
Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x5b10
Byte Offset: 0x16c40
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.

Bit	Reset	Description
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH27_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x5b11
Byte Offset: 0x16c44
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH27_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started. For example starting a flash in anticipation of an upcoming frame, or setup up a focusing process. This is done setting a tripline count which flags a notification once the number

of lines has been reached. The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. The line timer events are not filtered by the cropped region, they are solely based on the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x5b12
Byte Offset: 0x16c48
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH27_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x5b13
Byte Offset: 0x16c4c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH27_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use.

If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x5b14
Byte Offset: 0x16c50
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH27_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x5b15
Byte Offset: 0x16c54
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH27_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x5b16
Byte Offset: 0x16c58
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH27_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) -   SKIP_Y_LINES

Offset: 0x5b17
Byte Offset: 0x16c5c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH27_OUT_Y_0

Offset: 0x5b18
Byte Offset: 0x16c60
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH27_NOTIFY_MASK_0

Offset: 0x5b19
Byte Offset: 0x16c64
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH27_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED.

The expected use of NOMATCH is for debug purposes.

If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification.

To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x5b1a
Byte Offset: 0x16c68
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data

Bit	Reset	Description
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH27_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x5b1b
Byte Offset: 0x16c6c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH27_FRAME_COUNT_0

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame.

If in frame then FS=FE+1. FE can be perceived as a checksum check for FS.

Write any value to register to clear all counts.

Offset: 0x5b1c
Byte Offset: 0x16c70
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH27_LINE_TIMER_FIRST_0

Offset: 0x5b1d
Byte Offset: 0x16c74
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH27_FLUSH_FIRST_0

Offset: 0x5b1e
Byte Offset: 0x16c78
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH27_DOL_CTRL_0

Offset: 0x5b1f
Byte Offset: 0x16c7c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH27_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x5b20
Byte Offset: 0x16c80
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;

Bit	R/W	Reset	Description
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH27_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x5b21
Byte Offset: 0x16c84
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH27_PIXFMT_FORMAT_0

Offset: 0x5b22
Byte Offset: 0x16c88
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/ RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH27_PIXFMT_WIDE_0

Offset: 0x5b23
Byte Offset: 0x16c8c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH27_PIXFMT_PDAF_CROP_X_0Offset: 0x5b24

Byte Offset: 0x16c90
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH27_PIXFMT_PDAF_CROP_Y_0

Offset: 0x5b25
Byte Offset: 0x16c94
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH27_PIXFMT_PDAF_CONFIG0_0

Offset: 0x5b26
Byte Offset: 0x16c98
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH27_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x5b27
Byte Offset: 0x16c9c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH27_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x5b28
Byte Offset: 0x16ca0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH27_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.

Offset: 0x5b29
Byte Offset: 0x16ca4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH27_PIXFMT_PDAF_CONFIG1_0

Offset: 0x5b2a
Byte Offset: 0x16ca8
Read/Write: R/W
Parity Protection: Y

Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH27_DPCM_STRIP_0

Offset: 0x5b2f
Byte Offset: 0x16cbc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH27_DPCM_CHUNK_FIRST_0

Offset: 0x5b30
Byte Offset: 0x16cc0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH27_DPCM_CHUNK_BODY_0

Offset: 0x5b31
Byte Offset: 0x16cc4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH27_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x5b32
Byte Offset: 0x16cc8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH27_DPCM_CHUNK_LAST_0

Offset: 0x5b33
Byte Offset: 0x16ccc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH27_DPCM_STATISTICS_0_0

Offset: 0x5b34
Byte Offset: 0x16cd0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH27_DPCM_STATISTICS_1_0

Offset: 0x5b35
Byte Offset: 0x16cd4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH27_DPCM_MODE_0

Offset: 0x5b36
Byte Offset: 0x16cd8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH27_DPCM_CLAMP_HIGH_0

Offset: 0x5b37
Byte Offset: 0x16cdc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH27_DPCM_CLAMP_LOW_0

Offset: 0x5b38
Byte Offset: 0x16ce0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH27_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x5b3b
Byte Offset: 0x16cec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH27_ATOMP_SURFACE_OFFSET0_0

Offset: 0x5b3c
Byte Offset: 0x16cf0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH27_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x5b3d
Byte Offset: 0x16cf4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH27_ATOMP_SURFACE_STRIDE0_0

Offset: 0x5b3e
Byte Offset: 0x16cf8
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH27_ATOMP_DPCM_CHUNK_0

Offset: 0x5b3f
Byte Offset: 0x16cfc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH27_ATOMP_SURFACE_OFFSET1_0

Offset: 0x5b40
Byte Offset: 0x16d00
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH27_ATOMP_SURFACE_OFFSET1_H_0Offset: 0x5b41

Byte Offset: 0x16d04
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH27_ATOMP_SURFACE_STRIDE1_0

Offset: 0x5b42
Byte Offset: 0x16d08
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH27_ATOMP_SURFACE_OFFSET2_0

Offset: 0x5b43
Byte Offset: 0x16d0c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH27_ATOMP_SURFACE_OFFSET2_H_0Offset: 0x5b44
Byte Offset: 0x16d10
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH27_ATOMP_SURFACE_STRIDE2_0

Offset: 0x5b45
Byte Offset: 0x16d14
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH27_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x5b46
Byte Offset: 0x16d18
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH27_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x5b47
Byte Offset: 0x16d1c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH27_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x5b48
Byte Offset: 0x16d20
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH27_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x5b49
Byte Offset: 0x16d24
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH27_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x5b4a
Byte Offset: 0x16d28
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH27_ATOMP_HIGH_PRI_REQ_0

Offset: 0x5b4b
Byte Offset: 0x16d2c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH27_ATOMP_RESERVE_0

Offset: 0x5b4c
Byte Offset: 0x16d30
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 28 Specific Registers

VI_CH28_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x5c00
Byte Offset: 0x17000
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH28_CHANSEL_0

Offset: 0x5c04
Byte Offset: 0x17010
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH28_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x5c05
Byte Offset: 0x17014
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH28_CONTROL_0

Channel prefix gets added to register name (VI_CH*)

Offset: 0x5c06
Byte Offset: 0x17018
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.

Bit	Parity Protection	Reset	Description
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH28_MATCH_0

SHADOWED CHANSEL REGISTERSSTREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x5c07
Byte Offset: 0x1701c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH28_MATCH_DOL_0

Offset: 0x5c08
Byte Offset: 0x17020
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH28_MATCH_VC_HI_0

Offset: 0x5c09
Byte Offset: 0x17024
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH28_MATCH_DATATYPE_0

Offset: 0x5c0a
Byte Offset: 0x17028
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH28_MATCH_FRAMEID_0

Offset: 0x5c0b
Byte Offset: 0x1702c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH28_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT. The one exception is embedded data type which does not get remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on.

You are also likely not to have a wildcard but an exact match for data type It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x5c0c
Byte Offset: 0x17030
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.

Bit	Reset	Description
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH28_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x5c0d
Byte Offset: 0x17034
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH28_FRAME_Y_0

Expected frame y dimension

Offset: 0x5c0e
Byte Offset: 0x17038
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH28_EMBED_X_0
Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x5c0f
Byte Offset: 0x1703c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH28_EMBED_Y_0
Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x5c10
Byte Offset: 0x17040
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.

Bit	Reset	Description
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH28_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x5c11
Byte Offset: 0x17044
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH28_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started. For example starting a flash in anticipation of an upcoming frame, or setup up a focusing process. This is done setting a tripline count which flags a notification once the number

of lines has been reached. The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. The line timer events are not filtered by the cropped region, they are solely based on the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x5c12
Byte Offset: 0x17048
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH28_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x5c13
Byte Offset: 0x1704c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH28_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use. If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x5c14
Byte Offset: 0x17050
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH28_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x5c15
Byte Offset: 0x17054
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH28_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x5c16
Byte Offset: 0x17058
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH28_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) -   SKIP_Y_LINES

Offset: 0x5c17
Byte Offset: 0x1705c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH28_OUT_Y_0

Offset: 0x5c18
Byte Offset: 0x17060
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH28_NOTIFY_MASK_0

Offset: 0x5c19
Byte Offset: 0x17064
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH28_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED.

The expected use of NOMATCH is for debug purposes. If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification.

To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x5c1a
Byte Offset: 0x17068
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels

Bit	Reset	Description
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH28_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x5c1b
Byte Offset: 0x1706c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH28_FRAME_COUNT_0

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame.

If in frame then FS=FE+1. FE can be perceived as a checksum check for FS.

Write any value to register to clear all counts.

Offset: 0x5c1c
Byte Offset: 0x17070
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH28_LINE_TIMER_FIRST_0

Offset: 0x5c1d
Byte Offset: 0x17074
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH28_FLUSH_FIRST_0

Offset: 0x5c1e
Byte Offset: 0x17078
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH28_DOL_CTRL_0

Offset: 0x5c1f
Byte Offset: 0x1707c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH28_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x5c20
Byte Offset: 0x17080
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;

Bit	R/W	Reset	Description
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH28_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x5c21
Byte Offset: 0x17084
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH28_PIXFMT_FORMAT_0

Offset: 0x5c22
Byte Offset: 0x17088
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/ RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH28_PIXFMT_WIDE_0

Offset: 0x5c23
Byte Offset: 0x1708c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH28_PIXFMT_PDAF_CROP_X_0

Offset: 0x5c24
Byte Offset: 0x17090
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH28_PIXFMT_PDAF_CROP_Y_0

Offset: 0x5c25
Byte Offset: 0x17094
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH28_PIXFMT_PDAF_CONFIG0_0

Offset: 0x5c26
Byte Offset: 0x17098
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH28_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x5c27
Byte Offset: 0x1709c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends

Bit	Reset	Description
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH28_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x5c28
Byte Offset: 0x170a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH28_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.

Offset: 0x5c29
Byte Offset: 0x170a4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH28_PIXFMT_PDAF_CONFIG1_0

Offset: 0x5c2a
Byte Offset: 0x170a8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH28_DPCM_STRIP_0

Offset: 0x5c2f
Byte Offset: 0x170bc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH28_DPCM_CHUNK_FIRST_0

Offset: 0x5c30
Byte Offset: 0x170c0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH28_DPCM_CHUNK_BODY_0

Offset: 0x5c31
Byte Offset: 0x170c4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH28_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x5c32
Byte Offset: 0x170c8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH28_DPCM_CHUNK_LAST_0

Offset: 0x5c33
Byte Offset: 0x170cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH28_DPCM_STATISTICS_0_0

Offset: 0x5c34
Byte Offset: 0x170d0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH28_DPCM_STATISTICS_1_0

Offset: 0x5c35
Byte Offset: 0x170d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH28_DPCM_MODE_0

Offset: 0x5c36
Byte Offset: 0x170d8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH28_DPCM_CLAMP_HIGH_0 Offset: 0x5c37
Byte Offset: 0x170dc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH28_DPCM_CLAMP_LOW_0

Offset: 0x5c38
Byte Offset: 0x170e0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH28_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x5c3b
Byte Offset: 0x170ec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH28_ATOMP_SURFACE_OFFSET0_0

Offset: 0x5c3c
Byte Offset: 0x170f0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH28_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x5c3d
Byte Offset: 0x170f4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH28_ATOMP_SURFACE_STRIDE0_0

Offset: 0x5c3e
Byte Offset: 0x170f8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH28_ATOMP_DPCM_CHUNK_0

Offset: 0x5c3f
Byte Offset: 0x170fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH28_ATOMP_SURFACE_OFFSET1_0

Offset: 0x5c40
Byte Offset: 0x17100
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH28_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x5c41
Byte Offset: 0x17104
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH28_ATOMP_SURFACE_STRIDE1_0

Offset: 0x5c42
Byte Offset: 0x17108
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH28_ATOMP_SURFACE_OFFSET2_0

Offset: 0x5c43
Byte Offset: 0x1710c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH28_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x5c44
Byte Offset: 0x17110
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH28_ATOMP_SURFACE_STRIDE2_0

Offset: 0x5c45
Byte Offset: 0x17114
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH28_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x5c46
Byte Offset: 0x17118
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH28_ATOMP_EMB_SURFACE_OFFSET0_H_0 Offset: 0x5c47
Byte Offset: 0x1711c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH28_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x5c48
Byte Offset: 0x17120
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH28_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x5c49
Byte Offset: 0x17124
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH28_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x5c4a
Byte Offset: 0x17128
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH28_ATOMP_HIGH_PRI_REQ_0

Offset: 0x5c4b
Byte Offset: 0x1712c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH28_ATOMP_RESERVE_0

Offset: 0x5c4c
Byte Offset: 0x17130
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 29 Specific Registers

VI_CH29_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x5d00
Byte Offset: 0x17400
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH29_CHANSEL_0

Offset: 0x5d04
Byte Offset: 0x17410
Read/Write: R/W
Parity Protection: N
Shadow: Y

SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH29_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x5d05
Byte Offset: 0x17414
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH29_CONTROL_0

Channel prefix gets added to register name (VI_CH*)

Offset: 0x5d06
Byte Offset: 0x17418
Read/Write: R/W
Parity Protection: See table below
Shadow: N

SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH29_MATCH_0

SHADOWED CHANSEL REGISTERS

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x5d07
Byte Offset: 0x1741c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH29_MATCH_DOL_0

Offset: 0x5d08
Byte Offset: 0x17420
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH29_MATCH_VC_HI_0

Offset: 0x5d09
Byte Offset: 0x17424
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH29_MATCH_DATATYPE_0

Offset: 0x5d0a
Byte Offset: 0x17428
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH29_MATCH_FRAMEID_0

Offset: 0x5d0b
Byte Offset: 0x1742c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH29_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT. The one exception is embedded data type which does not get

remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on. You are also likely not to have a wildcard but an exact match for data type. It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error.The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x5d0c
Byte Offset: 0x17430
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH29_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x5d0d
Byte Offset: 0x17434
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH29_FRAME_Y_0

Expected frame y dimension

Offset: 0x5d0e
Byte Offset: 0x17438
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH29_EMBED_X_0
Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x5d0f
Byte Offset: 0x1743c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH29_EMBED_Y_0
Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x5d10
Byte Offset: 0x17440
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH29_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x5d11
Byte Offset: 0x17444
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH29_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started. For example starting a flash in anticipation of an upcoming frame, or setup up a focusing process. This is done setting a tripline count which flags a notification once the number of lines has been reached. The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

The line timer events are not filtered by the cropped region, they are solely based on the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x5d12
Byte Offset: 0x17448
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH29_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x5d13
Byte Offset: 0x1744c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH29_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use. If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x5d14
Byte Offset: 0x17450
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH29_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x5d15
Byte Offset: 0x17454
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH29_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x5d16
Byte Offset: 0x17458
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH29_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) - SKIP_Y_LINES

Offset: 0x5d17
Byte Offset: 0x1745c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH29_OUT_Y_0

Offset: 0x5d18
Byte Offset: 0x17460
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH29_NOTIFY_MASK_0

Offset: 0x5d19
Byte Offset: 0x17464
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped

Bit	Reset	Description
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH29_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED. The expected use of NOMATCH is for debug purposes. If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification.

To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x5d1a
Byte Offset: 0x17468
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)

Bit	Reset	Description
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH29_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x5d1b
Byte Offset: 0x1746c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH29_FRAME_COUNT_0

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame.

If in frame then FS=FE+1. FE can be perceived as a checksum check for FS.

Write any value to register to clear all counts.

Offset: 0x5d1c
Byte Offset: 0x17470
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH29_LINE_TIMER_FIRST_0

Offset: 0x5d1d
Byte Offset: 0x17474
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH29_FLUSH_FIRST_0

Offset: 0x5d1e
Byte Offset: 0x17478
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH29_DOL_CTRL_0

Offset: 0x5d1f
Byte Offset: 0x1747c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH29_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x5d20
Byte Offset: 0x17480
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH29_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x5d21
Byte Offset: 0x17484
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH29_PIXFMT_FORMAT_0

Offset: 0x5d22
Byte Offset: 0x17488
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0

Bit	Reset	Description
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH29_PIXFMT_WIDE_0

Offset: 0x5d23
Byte Offset: 0x1748c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH29_PIXFMT_PDAF_CROP_X_0

Offset: 0x5d24
Byte Offset: 0x17490
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH29_PIXFMT_PDAF_CROP_Y_0

Offset: 0x5d25
Byte Offset: 0x17494
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends

Bit	Reset	Description
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH29_PIXFMT_PDAF_CONFIG0_0

Offset: 0x5d26
Byte Offset: 0x17498
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH29_PIXFMT_PDAF_REPLACE_CROP_X_0Offset: 0x5d27

Byte Offset: 0x1749c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH29_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x5d28
Byte Offset: 0x174a0
Read/Write: R/W

Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH29_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.

Offset: 0x5d29
Byte Offset: 0x174a4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH29_PIXFMT_PDAF_CONFIG1_0

Offset: 0x5d2a
Byte Offset: 0x174a8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH29_DPCM_STRIP_0

Offset: 0x5d2f
Byte Offset: 0x174bc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH29_DPCM_CHUNK_FIRST_0

Offset: 0x5d30
Byte Offset: 0x174c0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH29_DPCM_CHUNK_BODY_0

Offset: 0x5d31
Byte Offset: 0x174c4

Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH29_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x5d32
Byte Offset: 0x174c8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH29_DPCM_CHUNK_LAST_0

Offset: 0x5d33
Byte Offset: 0x174cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH29_DPCM_STATISTICS_0_0

Offset: 0x5d34
Byte Offset: 0x174d0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH29_DPCM_STATISTICS_1_0

Offset: 0x5d35
Byte Offset: 0x174d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH29_DPCM_MODE_0

Offset: 0x5d36
Byte Offset: 0x174d8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH29_DPCM_CLAMP_HIGH_0

Offset: 0x5d37
Byte Offset: 0x174dc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH29_DPCM_CLAMP_LOW_0

Offset: 0x5d38
Byte Offset: 0x174e0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH29_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x5d3b
Byte Offset: 0x174ec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH29_ATOMP_SURFACE_OFFSET0_0

Offset: 0x5d3c
Byte Offset: 0x174f0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH29_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x5d3d
Byte Offset: 0x174f4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH29_ATOMP_SURFACE_STRIDE0_0

Offset: 0x5d3e
Byte Offset: 0x174f8
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH29_ATOMP_DPCM_CHUNK_0

Offset: 0x5d3f
Byte Offset: 0x174fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH29_ATOMP_SURFACE_OFFSET1_0

Offset: 0x5d40
Byte Offset: 0x17500
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH29_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x5d41
Byte Offset: 0x17504
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH29_ATOMP_SURFACE_STRIDE1_0

Offset: 0x5d42
Byte Offset: 0x17508
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH29_ATOMP_SURFACE_OFFSET2_0

Offset: 0x5d43
Byte Offset: 0x1750c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH29_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x5d44
Byte Offset: 0x17510
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH29_ATOMP_SURFACE_STRIDE2_0

Offset: 0x5d45
Byte Offset: 0x17514
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH29_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x5d46
Byte Offset: 0x17518
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH29_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x5d47
Byte Offset: 0x1751c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH29_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x5d48
Byte Offset: 0x17520
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH29_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x5d49
Byte Offset: 0x17524
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH29_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x5d4a
Byte Offset: 0x17528
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH29_ATOMP_HIGH_PRI_REQ_0

Offset: 0x5d4b
Byte Offset: 0x1752c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH29_ATOMP_RESERVE_0

Offset: 0x5d4c
Byte Offset: 0x17530
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 30 Specific Registers

VI_CH30_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x5e00
Byte Offset: 0x17800
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH30_CHANSEL_0

Offset: 0x5e04
Byte Offset: 0x17810
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH30_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x5e05
Byte Offset: 0x17814
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH30_CONTROL_0

Channel prefix gets added to register name (VI_CH*)

Offset: 0x5e06
Byte Offset: 0x17818
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.

Bit	Parity Protection	Reset	Description
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH30_MATCH_0

SHADOWED CHANSEL REGISTERS

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)Offset: 0x5e07
Byte Offset: 0x1781c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH30_MATCH_DOL_0

Offset: 0x5e08
Byte Offset: 0x17820
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH30_MATCH_VC_HI_0

Offset: 0x5e09
Byte Offset: 0x17824
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH30_MATCH_DATATYPE_0

Offset: 0x5e0a
Byte Offset: 0x17828
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH30_MATCH_FRAMEID_0

Offset: 0x5e0b
Byte Offset: 0x1782c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH30_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT. The one exception is embedded data type which does not get remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on.

You are also likely not to have a wildcard but an exact match for data type. It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x5e0c
Byte Offset: 0x17830
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.

Bit	Reset	Description
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH30_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x5e0d
Byte Offset: 0x17834
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH30_FRAME_Y_0

Expected frame y dimension

Offset: 0x5e0e
Byte Offset: 0x17838
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH30_EMBED_X_0
Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x5e0f
Byte Offset: 0x1783c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH30_EMBED_Y_0
Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x5e10
Byte Offset: 0x17840
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.

Bit	Reset	Description
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH30_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x5e11
Byte Offset: 0x17844
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH30_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started. For example starting a flash in anticipation of an upcoming frame, or setup up a focusing process. This is done setting a tripline count which flags a notification once the number

of lines has been reached. The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. The line timer events are not filtered by the cropped region, they are solely based on the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x5e12
Byte Offset: 0x17848
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH30_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x5e13
Byte Offset: 0x1784c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH30_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use. If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x5e14
Byte Offset: 0x17850
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH30_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x5e15
Byte Offset: 0x17854
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH30_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x5e16
Byte Offset: 0x17858
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH30_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) -   SKIP_Y_LINES

Offset: 0x5e17
Byte Offset: 0x1785c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH30_OUT_Y_0

Offset: 0x5e18
Byte Offset: 0x17860
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH30_NOTIFY_MASK_0

Offset: 0x5e19
Byte Offset: 0x17864
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH30_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED.The expected use of NOMATCH is for debug purposes.

If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification.To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x5e1a
Byte Offset: 0x17868
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH30_FRAME_SOURCE_0

Observation Registers

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x5e1b
Byte Offset: 0x1786c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH30_FRAME_COUNT_0

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame.

If in frame then FS=FE+1. FE can be perceived as a checksum check for FS. Write any value to register to clear all counts.

Offset: 0x5e1c
Byte Offset: 0x17870
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH30_LINE_TIMER_FIRST_0

Offset: 0x5e1d
Byte Offset: 0x17874
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH30_FLUSH_FIRST_0

Offset: 0x5e1e
Byte Offset: 0x17878
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH30_DOL_CTRL_0

Offset: 0x5e1f
Byte Offset: 0x1787c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH30_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x5e20
Byte Offset: 0x17880
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH30_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x5e21
Byte Offset: 0x17884
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH30_PIXFMT_FORMAT_0

Offset: 0x5e22
Byte Offset: 0x17888
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/ RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0

Bit	Reset	Description
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH30_PIXFMT_WIDE_0

Offset: 0x5e23
Byte Offset: 0x1788c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH30_PIXFMT_PDAF_CROP_X_0

Offset: 0x5e24
Byte Offset: 0x17890
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH30_PIXFMT_PDAF_CROP_Y_0

Offset: 0x5e25
Byte Offset: 0x17894
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends

Bit	Reset	Description
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH30_PIXFMT_PDAF_CONFIG0_0

Offset: 0x5e26
Byte Offset: 0x17898
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH30_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x5e27
Byte Offset: 0x1789c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH30_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x5e28
Byte Offset: 0x178a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH30_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.

Offset: 0x5e29
Byte Offset: 0x178a4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH30_PIXFMT_PDAF_CONFIG1_0

Offset: 0x5e2a
Byte Offset: 0x178a8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH30_DPCM_STRIP_0

Offset: 0x5e2f
Byte Offset: 0x178bc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH30_DPCM_CHUNK_FIRST_0

Offset: 0x5e30
Byte Offset: 0x178c0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH30_DPCM_CHUNK_BODY_0

Offset: 0x5e31
Byte Offset: 0x178c4

Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH30_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x5e32
Byte Offset: 0x178c8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH30_DPCM_CHUNK_LAST_0

Offset: 0x5e33
Byte Offset: 0x178cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH30_DPCM_STATISTICS_0_0

Offset: 0x5e34
Byte Offset: 0x178d0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH30_DPCM_STATISTICS_1_0

Offset: 0x5e35
Byte Offset: 0x178d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH30_DPCM_MODE_0

Offset: 0x5e36
Byte Offset: 0x178d8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH30_DPCM_CLAMP_HIGH_0

Offset: 0x5e37
Byte Offset: 0x178dc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH30_DPCM_CLAMP_LOW_0

Offset: 0x5e38
Byte Offset: 0x178e0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH30_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x5e3b
Byte Offset: 0x178ec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH30_ATOMP_SURFACE_OFFSET0_0

Offset: 0x5e3c
Byte Offset: 0x178f0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH30_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x5e3d
Byte Offset: 0x178f4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH30_ATOMP_SURFACE_STRIDE0_0

Offset: 0x5e3e
Byte Offset: 0x178f8
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH30_ATOMP_DPCM_CHUNK_0

Offset: 0x5e3f
Byte Offset: 0x178fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH30_ATOMP_SURFACE_OFFSET1_0

Offset: 0x5e40
Byte Offset: 0x17900
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH30_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x5e41
Byte Offset: 0x17904
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH30_ATOMP_SURFACE_STRIDE1_0

Offset: 0x5e42
Byte Offset: 0x17908
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH30_ATOMP_SURFACE_OFFSET2_0

Offset: 0x5e43
Byte Offset: 0x1790c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH30_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x5e44
Byte Offset: 0x17910
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH30_ATOMP_SURFACE_STRIDE2_0

Offset: 0x5e45
Byte Offset: 0x17914
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH30_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x5e46
Byte Offset: 0x17918
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH30_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x5e47
Byte Offset: 0x1791c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH30_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x5e48
Byte Offset: 0x17920
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH30_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x5e49
Byte Offset: 0x17924
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH30_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x5e4a
Byte Offset: 0x17928
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH30_ATOMP_HIGH_PRI_REQ_0

Offset: 0x5e4b
Byte Offset: 0x1792c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH30_ATOMP_RESERVE_0

Offset: 0x5e4c
Byte Offset: 0x17930
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 31 Specific Registers

VI_CH31_CHANNEL_COMMAND_0

Offset: 0x5f00
Byte Offset: 0x17c00
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH31_CHANSEL_0

Offset: 0x5f04
Byte Offset: 0x17c10
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH31_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x5f05
Byte Offset: 0x17c14
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH31_CONTROL_0

Channel prefix gets added to register name (VI_CH*)

Offset: 0x5f06
Byte Offset: 0x17c18
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.

Bit	Parity Protection	Reset	Description
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH31_MATCH_0

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x5f07
Byte Offset: 0x17c1c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**

Bit	Reset	Description
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH31_MATCH_DOL_0

Offset: 0x5f08
Byte Offset: 0x17c20
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH31_MATCH_VC_HI_0

Offset: 0x5f09
Byte Offset: 0x17c24
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH31_MATCH_DATATYPE_0

Offset: 0x5f0a
Byte Offset: 0x17c28
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH31_MATCH_FRAMEID_0

Offset: 0x5f0b
Byte Offset: 0x17c2c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH31_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT. The one exception is embedded data type which does not get remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on. You are also likely not to have a wildcard but an exact match for data type. It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x5f0c
Byte Offset: 0x17c30
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.

Bit	Reset	Description
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH31_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x5f0d
Byte Offset: 0x17c34
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH31_FRAME_Y_0

Expected frame y dimension

Offset: 0x5f0e
Byte Offset: 0x17c38
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH31_EMBED_X_0

Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x5f0f
Byte Offset: 0x17c3c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH31_EMBED_Y_0
Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x5f10
Byte Offset: 0x17c40
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH31_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached.The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x5f11
Byte Offset: 0x17c44
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH31_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started. For example starting a flash in anticipation of an upcoming frame, or setup up a focusing process. This is done setting a tripline count which flags a notification once the number of lines has been reached. The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. The line timer events are not filtered by the cropped region, they are solely based on

the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x5f12
Byte Offset: 0x17c48
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH31_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data) Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x5f13
Byte Offset: 0x17c4c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH31_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1. Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use. If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x5f14
Byte Offset: 0x17c50
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH31_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x5f15
Byte Offset: 0x17c54
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH31_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x5f16
Byte Offset: 0x17c58
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH31_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) - SKIP_Y_LINES

Offset: 0x5f17
Byte Offset: 0x17c5c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH31_OUT_Y_0

Offset: 0x5f18
Byte Offset: 0x17c60
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH31_NOTIFY_MASK_0

Offset: 0x5f19
Byte Offset: 0x17c64
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events

Bit	Reset	Description
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH31_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED.The expected use of NOMATCH is for debug purposes.

If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification. To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x5f1a
Byte Offset: 0x17c68
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH31_FRAME_SOURCE_0

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x5f1b
Byte Offset: 0x17c6c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH31_FRAME_COUNT_0

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame.

If in frame then FS=FE+1. FE can be perceived as a checksum check for FS.

Write any value to register to clear all counts.

Offset: 0x5f1c
Byte Offset: 0x17c70
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH31_LINE_TIMER_FIRST_0

Offset: 0x5f1d
Byte Offset: 0x17c74
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH31_FLUSH_FIRST_0

Offset: 0x5f1e
Byte Offset: 0x17c78
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH31_DOL_CTRL_0

Offset: 0x5f1f
Byte Offset: 0x17c7c
Read/Write: R/W
Parity Protection: Y

Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH31_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x5f20
Byte Offset: 0x17c80
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH31_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x5f21
Byte Offset: 0x17c84
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH31_PIXFMT_FORMAT_0

Offset: 0x5f22
Byte Offset: 0x17c88
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/ RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0

Bit	Reset	Description
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH31_PIXFMT_WIDE_0

Offset: 0x5f23
Byte Offset: 0x17c8c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH31_PIXFMT_PDAF_CROP_X_0

Offset: 0x5f24
Byte Offset: 0x17c90
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH31_PIXFMT_PDAF_CROP_Y_0

Offset: 0x5f25
Byte Offset: 0x17c94
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends

Bit	Reset	Description
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH31_PIXFMT_PDAF_CONFIG0_0

Offset: 0x5f26
Byte Offset: 0x17c98
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH31_PIXFMT_PDAF_REPLACE_CROP_X_0Offset: 0x5f27
Byte Offset: 0x17c9c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH31_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x5f28
Byte Offset: 0x17ca0
Read/Write: R/W

Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH31_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.

Offset: 0x5f29
Byte Offset: 0x17ca4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH31_PIXFMT_PDAF_CONFIG1_0

Offset: 0x5f2a
Byte Offset: 0x17ca8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH31_DPCM_STRIP_0

Offset: 0x5f2f
Byte Offset: 0x17cbc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH31_DPCM_CHUNK_FIRST_0

Offset: 0x5f30
Byte Offset: 0x17cc0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH31_DPCM_CHUNK_BODY_0

Offset: 0x5f31
Byte Offset: 0x17cc4

Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH31_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x5f32
Byte Offset: 0x17cc8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH31_DPCM_CHUNK_LAST_0

Offset: 0x5f33
Byte Offset: 0x17ccc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH31_DPCM_STATISTICS_0_0

Offset: 0x5f34
Byte Offset: 0x17cd0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH31_DPCM_STATISTICS_1_0

Offset: 0x5f35
Byte Offset: 0x17cd4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH31_DPCM_MODE_0

Offset: 0x5f36
Byte Offset: 0x17cd8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH31_DPCM_CLAMP_HIGH_0

Offset: 0x5f37
Byte Offset: 0x17cdc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH31_DPCM_CLAMP_LOW_0

Offset: 0x5f38
Byte Offset: 0x17ce0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH31_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x5f3b
Byte Offset: 0x17cec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH31_ATOMP_SURFACE_OFFSET0_0

Offset: 0x5f3c
Byte Offset: 0x17cf0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH31_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x5f3d
Byte Offset: 0x17cf4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH31_ATOMP_SURFACE_STRIDE0_0

Offset: 0x5f3e
Byte Offset: 0x17cf8
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH31_ATOMP_DPCM_CHUNK_0

Offset: 0x5f3f
Byte Offset: 0x17cfc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH31_ATOMP_SURFACE_OFFSET1_0

Offset: 0x5f40
Byte Offset: 0x17d00
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH31_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x5f41
Byte Offset: 0x17d04
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH31_ATOMP_SURFACE_STRIDE1_0

Offset: 0x5f42
Byte Offset: 0x17d08
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH31_ATOMP_SURFACE_OFFSET2_0

Offset: 0x5f43
Byte Offset: 0x17d0c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH31_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x5f44
Byte Offset: 0x17d10
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH31_ATOMP_SURFACE_STRIDE2_0

Offset: 0x5f45
Byte Offset: 0x17d14
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH31_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x5f46
Byte Offset: 0x17d18
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH31_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x5f47
Byte Offset: 0x17d1c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH31_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x5f48
Byte Offset: 0x17d20
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH31_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x5f49
Byte Offset: 0x17d24
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH31_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x5f4a
Byte Offset: 0x17d28
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH31_ATOMP_HIGH_PRI_REQ_0

Offset: 0x5f4b
Byte Offset: 0x17d2c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH31_ATOMP_RESERVE_0

Offset: 0x5f4c
Byte Offset: 0x17d30
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 32 Specific Registers

VI_CH32_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x6000
Byte Offset: 0x18000
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH32_CHANSEL_0

Offset: 0x6004
Byte Offset: 0x18010
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH32_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x6005
Byte Offset: 0x18014
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH32_CONTROL_0

Channel prefix gets added to register name (VI_CH*)

Offset: 0x6006
Byte Offset: 0x18018
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.

Bit	Parity Protection	Reset	Description
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH32_MATCH_0

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x6007
Byte Offset: 0x1801c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**

Bit	Reset	Description
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH32_MATCH_DOL_0

Offset: 0x6008
Byte Offset: 0x18020
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH32_MATCH_VC_HI_0

Offset: 0x6009
Byte Offset: 0x18024
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH32_MATCH_DATATYPE_0

Offset: 0x600a
Byte Offset: 0x18028
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH32_MATCH_FRAMEID_0

Offset: 0x600b
Byte Offset: 0x1802c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH32_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT. The one exception is embedded data type which does not get remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on. You are also likely not to have a wildcard but an exact match for data type. It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x600c
Byte Offset: 0x18030
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.

Bit	Reset	Description
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH32_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x600d
Byte Offset: 0x18034
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH32_FRAME_Y_0

Expected frame y dimension

Offset: 0x600e
Byte Offset: 0x18038
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH32_EMBED_X_0
Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x600f
Byte Offset: 0x1803c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH32_EMBED_Y_0
Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x6010
Byte Offset: 0x18040
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.

Bit	Reset	Description
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH32_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached.

The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x6011
Byte Offset: 0x18044
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH32_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started. For example starting a flash in anticipation of an upcoming frame, or setup up a

focusing process. This is done setting a tripline count which flags a notification once the number of lines has been reached. The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. The line timer events are not filtered by the cropped region, they are solely based on the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x6012
Byte Offset: 0x18048
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH32_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x6013
Byte Offset: 0x1804c
Read/Write: R/W
Parity Protection: Y

Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH32_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use. If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x6014
Byte Offset: 0x18050
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH32_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x6015
Byte Offset: 0x18054
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH32_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x6016
Byte Offset: 0x18058
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH32_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) -   SKIP_Y_LINES

Offset: 0x6017
Byte Offset: 0x1805c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH32_OUT_Y_0

Offset: 0x6018
Byte Offset: 0x18060
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH32_NOTIFY_MASK_0

Offset: 0x6019
Byte Offset: 0x18064
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH32_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED.The expected use of NOMATCH is for debug purposes.

If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification. To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x601a
Byte Offset: 0x18068
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH32_FRAME_SOURCE_0

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x601b
Byte Offset: 0x1806c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH32_FRAME_COUNT_0

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame.

If in frame then FS=FE+1. FE can be perceived as a checksum check for FS.

Write any value to register to clear all counts.

Offset: 0x601c
Byte Offset: 0x18070
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255

Bit	Reset	Description
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH32_LINE_TIMER_FIRST_0

Offset: 0x601d
Byte Offset: 0x18074
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH32_FLUSH_FIRST_0

Offset: 0x601e
Byte Offset: 0x18078
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH32_DOL_CTRL_0

Offset: 0x601f
Byte Offset: 0x1807c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH32_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x6020
Byte Offset: 0x18080
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH32_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x6021
Byte Offset: 0x18084
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH32_PIXFMT_FORMAT_0

Offset: 0x6022
Byte Offset: 0x18088
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/ RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0

Bit	Reset	Description
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH32_PIXFMT_WIDE_0

Offset: 0x6023
Byte Offset: 0x1808c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH32_PIXFMT_PDAF_CROP_X_0

Offset: 0x6024
Byte Offset: 0x18090
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH32_PIXFMT_PDAF_CROP_Y_0

Offset: 0x6025
Byte Offset: 0x18094
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends

Bit	Reset	Description
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH32_PIXFMT_PDAF_CONFIG0_0

Offset: 0x6026
Byte Offset: 0x18098
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH32_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x6027
Byte Offset: 0x1809c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH32_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x6028
Byte Offset: 0x180a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH32_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.

Offset: 0x6029
Byte Offset: 0x180a4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH32_PIXFMT_PDAF_CONFIG1_0

Offset: 0x602a
Byte Offset: 0x180a8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH32_DPCM_STRIP_0

Offset: 0x602f
Byte Offset: 0x180bc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH32_DPCM_CHUNK_FIRST_0

Offset: 0x6030
Byte Offset: 0x180c0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH32_DPCM_CHUNK_BODY_0

Offset: 0x6031
Byte Offset: 0x180c4

Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH32_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x6032
Byte Offset: 0x180c8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH32_DPCM_CHUNK_LAST_0

Offset: 0x6033
Byte Offset: 0x180cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH32_DPCM_STATISTICS_0_0

Offset: 0x6034
Byte Offset: 0x180d0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH32_DPCM_STATISTICS_1_0

Offset: 0x6035
Byte Offset: 0x180d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH32_DPCM_MODE_0

Offset: 0x6036
Byte Offset: 0x180d8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH32_DPCM_CLAMP_HIGH_0

Offset: 0x6037
Byte Offset: 0x180dc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH32_DPCM_CLAMP_LOW_0

Offset: 0x6038
Byte Offset: 0x180e0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH32_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x603b
Byte Offset: 0x180ec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH32_ATOMP_SURFACE_OFFSET0_0

Offset: 0x603c
Byte Offset: 0x180f0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH32_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x603d
Byte Offset: 0x180f4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH32_ATOMP_SURFACE_STRIDE0_0

Offset: 0x603e
Byte Offset: 0x180f8
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH32_ATOMP_DPCM_CHUNK_0

Offset: 0x603f
Byte Offset: 0x180fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH32_ATOMP_SURFACE_OFFSET1_0

Offset: 0x6040
Byte Offset: 0x18100
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH32_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x6041
Byte Offset: 0x18104
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH32_ATOMP_SURFACE_STRIDE1_0

Offset: 0x6042
Byte Offset: 0x18108
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH32_ATOMP_SURFACE_OFFSET2_0

Offset: 0x6043
Byte Offset: 0x1810c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH32_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x6044
Byte Offset: 0x18110
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH32_ATOMP_SURFACE_STRIDE2_0

Offset: 0x6045
Byte Offset: 0x18114
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH32_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x6046
Byte Offset: 0x18118
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH32_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x6047
Byte Offset: 0x1811c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH32_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x6048
Byte Offset: 0x18120
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH32_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x6049
Byte Offset: 0x18124
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH32_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x604a
Byte Offset: 0x18128
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH32_ATOMP_HIGH_PRI_REQ_0

Offset: 0x604b
Byte Offset: 0x1812c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH32_ATOMP_RESERVE_0

Offset: 0x604c
Byte Offset: 0x18130
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 33 Specific Registers

VI_CH33_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x6100
Byte Offset: 0x18400
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH33_CHANSEL_0

Offset: 0x6104
Byte Offset: 0x18410
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH33_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x6105
Byte Offset: 0x18414
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH33_CONTROL_0

Channel prefix gets added to register name (VI_CH*)

Offset: 0x6106
Byte Offset: 0x18418
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.

Bit	Parity Protection	Reset	Description
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH33_MATCH_0

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x6107
Byte Offset: 0x1841c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**

Bit	Reset	Description
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH33_MATCH_DOL_0

Offset: 0x6108
Byte Offset: 0x18420
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH33_MATCH_VC_HI_0

Offset: 0x6109
Byte Offset: 0x18424
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH33_MATCH_DATATYPE_0

Offset: 0x610a
Byte Offset: 0x18428
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH33_MATCH_FRAMEID_0

Offset: 0x610b
Byte Offset: 0x1842c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH33_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT. The one exception is embedded data type which does not get remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on. You are also likely not to have a wildcard but an exact match for data type. It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x610c
Byte Offset: 0x18430
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.

Bit	Reset	Description
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH33_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x610d
Byte Offset: 0x18434
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH33_FRAME_Y_0

Expected frame y dimension

Offset: 0x610e
Byte Offset: 0x18438
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH33_EMBED_X_0
Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x610f
Byte Offset: 0x1843c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH33_EMBED_Y_0
Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x6110
Byte Offset: 0x18440
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.

Bit	Reset	Description
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH33_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x6111
Byte Offset: 0x18444
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH33_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started. For example starting a flash in anticipation of an upcoming frame, or setup up a focusing process. This is done setting a tripline count which flags a notification once the number

of lines has been reached The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines.

The line timer events are not filtered by the cropped region, they are solely based on the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x6112
Byte Offset: 0x18448
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH33_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x6113
Byte Offset: 0x1844c
Read/Write: R/W
Parity Protection: Y

Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH33_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use. If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x6114
Byte Offset: 0x18450
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH33_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x6115
Byte Offset: 0x18454
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH33_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x6116
Byte Offset: 0x18458
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH33_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) -   SKIP_Y_LINES

Offset: 0x6117
Byte Offset: 0x1845c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH33_OUT_Y_0

Offset: 0x6118
Byte Offset: 0x18460
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH33_NOTIFY_MASK_0

Offset: 0x6119
Byte Offset: 0x18464
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH33_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED.The expected use of NOMATCH is for debug purposes.

If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification.

To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x611a
Byte Offset: 0x18468
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH33_FRAME_SOURCE_0

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x611b
Byte Offset: 0x1846c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH33_FRAME_COUNT_0

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame.

If in frame then FS=FE+1. FE can be perceived as a checksum check for FS.

Write any value to register to clear all counts.

Offset: 0x611c
Byte Offset: 0x18470
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH33_LINE_TIMER_FIRST_0

Offset: 0x611d
Byte Offset: 0x18474
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH33_FLUSH_FIRST_0

Offset: 0x611e
Byte Offset: 0x18478
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH33_DOL_CTRL_0

Offset: 0x611f
Byte Offset: 0x1847c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH33_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x6120
Byte Offset: 0x18480
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH33_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x6121
Byte Offset: 0x18484
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH33_PIXFMT_FORMAT_0

Offset: 0x6122
Byte Offset: 0x18488
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/ RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0

Bit	Reset	Description
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH33_PIXFMT_WIDE_0

Offset: 0x6123
Byte Offset: 0x1848c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH33_PIXFMT_PDAF_CROP_X_0

Offset: 0x6124
Byte Offset: 0x18490
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH33_PIXFMT_PDAF_CROP_Y_0

Offset: 0x6125
Byte Offset: 0x18494
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends

Bit	Reset	Description
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH33_PIXFMT_PDAF_CONFIG0_0

Offset: 0x6126
Byte Offset: 0x18498
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH33_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x6127
Byte Offset: 0x1849c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH33_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x6128
Byte Offset: 0x184a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH33_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.

Offset: 0x6129
Byte Offset: 0x184a4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH33_PIXFMT_PDAF_CONFIG1_0

Offset: 0x612a
Byte Offset: 0x184a8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH33_DPCM_STRIP_0

Offset: 0x612f
Byte Offset: 0x184bc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH33_DPCM_CHUNK_FIRST_0

Offset: 0x6130
Byte Offset: 0x184c0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH33_DPCM_CHUNK_BODY_0

Offset: 0x6131
Byte Offset: 0x184c4

Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH33_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x6132
Byte Offset: 0x184c8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH33_DPCM_CHUNK_LAST_0

Offset: 0x6133
Byte Offset: 0x184cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH33_DPCM_STATISTICS_0_0

Offset: 0x6134
Byte Offset: 0x184d0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH33_DPCM_STATISTICS_1_0

Offset: 0x6135
Byte Offset: 0x184d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH33_DPCM_MODE_0

Offset: 0x6136
Byte Offset: 0x184d8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH33_DPCM_CLAMP_HIGH_0

Offset: 0x6137
Byte Offset: 0x184dc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH33_DPCM_CLAMP_LOW_0

Offset: 0x6138
Byte Offset: 0x184e0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH33_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x613b
Byte Offset: 0x184ec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH33_ATOMP_SURFACE_OFFSET0_0

Offset: 0x613c
Byte Offset: 0x184f0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH33_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x613d
Byte Offset: 0x184f4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH33_ATOMP_SURFACE_STRIDE0_0

Offset: 0x613e
Byte Offset: 0x184f8
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH33_ATOMP_DPCM_CHUNK_0

Offset: 0x613f
Byte Offset: 0x184fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH33_ATOMP_SURFACE_OFFSET1_0

Offset: 0x6140
Byte Offset: 0x18500
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH33_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x6141
Byte Offset: 0x18504
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH33_ATOMP_SURFACE_STRIDE1_0

Offset: 0x6142
Byte Offset: 0x18508
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH33_ATOMP_SURFACE_OFFSET2_0

Offset: 0x6143
Byte Offset: 0x1850c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH33_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x6144
Byte Offset: 0x18510
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH33_ATOMP_SURFACE_STRIDE2_0

Offset: 0x6145
Byte Offset: 0x18514
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH33_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x6146
Byte Offset: 0x18518
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH33_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x6147
Byte Offset: 0x1851c
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH33_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x6148
Byte Offset: 0x18520
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH33_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x6149
Byte Offset: 0x18524
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH33_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x614a
Byte Offset: 0x18528
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH33_ATOMP_HIGH_PRI_REQ_0

Offset: 0x614b
Byte Offset: 0x1852c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH33_ATOMP_RESERVE_0

Offset: 0x614c
Byte Offset: 0x18530
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 34 Specific Registers

VI_CH34_CHANNEL_COMMAND_0

Offset: 0x6200
Byte Offset: 0x18800
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH34_CHANSEL_0

Offset: 0x6204
Byte Offset: 0x18810
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH34_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x6205
Byte Offset: 0x18814
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH34_CONTROL_0

Channel prefix gets added to register name (VI_CH*)

Offset: 0x6206
Byte Offset: 0x18818
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.

Bit	Parity Protection	Reset	Description
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH34_MATCH_0

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x6207
Byte Offset: 0x1881c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**

Bit	Reset	Description
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH34_MATCH_DOL_0

Offset: 0x6208
Byte Offset: 0x18820
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH34_MATCH_VC_HI_0

Offset: 0x6209
Byte Offset: 0x18824
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH34_MATCH_DATATYPE_0

Offset: 0x620a
Byte Offset: 0x18828
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH34_MATCH_FRAMEID_0

Offset: 0x620b
Byte Offset: 0x1882c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH34_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT. The one exception is embedded data type which does not get remapped through this register.

If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on. You are also likely not to have a wildcard but an exact match for data type. It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x620c
Byte Offset: 0x18830
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.

Bit	Reset	Description
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH34_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x620d
Byte Offset: 0x18834
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH34_FRAME_Y_0

Expected frame y dimension

Offset: 0x620e
Byte Offset: 0x18838
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH34_EMBED_X_0

Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x620f
Byte Offset: 0x1883c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH34_EMBED_Y_0

Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x6210
Byte Offset: 0x18840
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.

Bit	Reset	Description
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH34_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x6211
Byte Offset: 0x18844
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH34_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started. For example starting a flash in anticipation of an upcoming frame, or setup up a focusing process.

This is done setting a tripline count which flags a notification once the number of lines has been reached. The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. The line timer events are not filtered by the cropped region, they are solely based on the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x6212
Byte Offset: 0x18848
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH34_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x6213
Byte Offset: 0x1884c
Read/Write: R/W
Parity Protection: Y

Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH34_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop.

Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use. If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x6214
Byte Offset: 0x18850
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH34_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x6215
Byte Offset: 0x18854
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH34_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x6216
Byte Offset: 0x18858
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH34_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) -   SKIP_Y_LINES

Offset: 0x6217
Byte Offset: 0x1885c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH34_OUT_Y_0

Offset: 0x6218
Byte Offset: 0x18860
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH34_NOTIFY_MASK_0

Offset: 0x6219
Byte Offset: 0x18864
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH34_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED.

The expected use of NOMATCH is for debug purposes.

If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification.

To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x621a
Byte Offset: 0x18868
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data

Bit	Reset	Description
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH34_FRAME_SOURCE_0

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x621b
Byte Offset: 0x1886c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH34_FRAME_COUNT_0

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame.

If in frame then FS=FE+1. FE can be perceived as a checksum check for FS. Write any value to register to clear all counts.

Offset: 0x621c
Byte Offset: 0x18870
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH34_LINE_TIMER_FIRST_0

Offset: 0x621d
Byte Offset: 0x18874
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH34_FLUSH_FIRST_0

Offset: 0x621e
Byte Offset: 0x18878
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH34_DOL_CTRL_0

Offset: 0x621f
Byte Offset: 0x1887c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH34_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x6220
Byte Offset: 0x18880
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;

30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH34_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x6221
Byte Offset: 0x18884
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH34_PIXFMT_FORMAT_0

Offset: 0x6222
Byte Offset: 0x18888
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/ RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH34_PIXFMT_WIDE_0

Offset: 0x6223
Byte Offset: 0x1888c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH34_PIXFMT_PDAF_CROP_X_0

Offset: 0x6224
Byte Offset: 0x18890
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH34_PIXFMT_PDAF_CROP_Y_0

Offset: 0x6225
Byte Offset: 0x18894
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH34_PIXFMT_PDAF_CONFIG0_0

Offset: 0x6226
Byte Offset: 0x18898
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH34_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x6227
Byte Offset: 0x1889c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends

Bit	Reset	Description
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH34_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x6228
Byte Offset: 0x188a0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH34_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.

Offset: 0x6229
Byte Offset: 0x188a4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH34_PIXFMT_PDAF_CONFIG1_0

Offset: 0x622a
Byte Offset: 0x188a8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH34_DPCM_STRIP_0

Offset: 0x622f
Byte Offset: 0x188bc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH34_DPCM_CHUNK_FIRST_0

Offset: 0x6230
Byte Offset: 0x188c0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH34_DPCM_CHUNK_BODY_0

Offset: 0x6231
Byte Offset: 0x188c4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH34_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x6232
Byte Offset: 0x188c8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH34_DPCM_CHUNK_LAST_0

Offset: 0x6233
Byte Offset: 0x188cc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH34_DPCM_STATISTICS_0_0

Offset: 0x6234
Byte Offset: 0x188d0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH34_DPCM_STATISTICS_1_0

Offset: 0x6235
Byte Offset: 0x188d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH34_DPCM_MODE_0

Offset: 0x6236
Byte Offset: 0x188d8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH34_DPCM_CLAMP_HIGH_0

Offset: 0x6237
Byte Offset: 0x188dc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH34_DPCM_CLAMP_LOW_0

Offset: 0x6238
Byte Offset: 0x188e0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH34_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x623b
Byte Offset: 0x188ec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH34_ATOMP_SURFACE_OFFSET0_0

Offset: 0x623c
Byte Offset: 0x188f0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH34_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x623d
Byte Offset: 0x188f4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH34_ATOMP_SURFACE_STRIDE0_0

Offset: 0x623e
Byte Offset: 0x188f8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH34_ATOMP_DPCM_CHUNK_0

Offset: 0x623f
Byte Offset: 0x188fc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH34_ATOMP_SURFACE_OFFSET1_0

Offset: 0x6240
Byte Offset: 0x18900
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH34_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x6241
Byte Offset: 0x18904
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH34_ATOMP_SURFACE_STRIDE1_0

Offset: 0x6242
Byte Offset: 0x18908
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH34_ATOMP_SURFACE_OFFSET2_0

Offset: 0x6243
Byte Offset: 0x1890c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH34_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x6244
Byte Offset: 0x18910
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH34_ATOMP_SURFACE_STRIDE2_0

Offset: 0x6245
Byte Offset: 0x18914
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH34_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x6246
Byte Offset: 0x18918
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH34_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x6247
Byte Offset: 0x1891c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH34_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x6248
Byte Offset: 0x18920
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH34_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x6249
Byte Offset: 0x18924
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH34_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x624a
Byte Offset: 0x18928
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH34_ATOMP_HIGH_PRI_REQ_0

Offset: 0x624b
Byte Offset: 0x1892c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH34_ATOMP_RESERVE_0

Offset: 0x624c
Byte Offset: 0x18930
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

Channel 35 Specific Registers

VI_CH35_CHANNEL_COMMAND_0

HOST CHANNEL REGISTERS (non-shadowed)

Offset: 0x6300
Byte Offset: 0x18c00
Read/Write: R/W
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00)

Bit	Parity Protection	Reset	Description
5	Y	0x0	**WR_ACT_SEL**: Host write operations operate on both shadow and active register (debug)
4	Y	0x0	**RD_MUX_SEL**: Host read operations are on non-shadowed register (debug)
1	Y	0x0	**AUTOLOAD**: Set to 1, if this channel should automatically load a new configuration (as if LOAD were set) at the time that an EOF is received at the last pipeline stage that reads from configuration registers.
0	N	0x0	**LOAD**: Atomically loads all channel state from shadow register to active registers. Write 1 to load, Always read as 0.

VI_CH35_CHANSEL_0

Offset: 0x6304
Byte Offset: 0x18c10
Read/Write: R/W
Parity Protection: N
Shadow: Y

SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffff	**SPARE**: Per CHANNEL shadowed spare register for chansel

VI_CH35_ENABLE_0

Channel prefix gets added to register name (VI_CH*)

NON-SHADOWED CHANSEL REGISTER

When turning channel ENABLE off (1 -> 0) it is always important to issue a follow up LOAD command from the CHANNEL_COMMAND register in channel hostif. This guarantees that any ongoing frame on the channel is properly terminated. The two step process ensures no races in hardware. Put channel enable to a separate register field to avoid SW race condition.

Offset: 0x6305
Byte Offset: 0x18c14
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Enables or disables this channel's match state in CHANSEL. Does not necessarily disable the channel in all stages of the pipeline. If there are atoms remaining to be written in ATOMP, they will still be written even if the channel is disabled.

VI_CH35_CONTROL_0

Channel prefix gets added to register name (VI_CH*)

Offset: 0x6306
Byte Offset: 0x18c18
Read/Write: R/W
Parity Protection: See table below
Shadow: N

SCR Protection: 0
Reset: 0x00ff0030 (0b0000,0000,1111,1111,xxxx,xxxx,xx11,000x)

Bit	Parity Protection	Reset	Description
31:16	N	0xff	**SPARE**: Spare control bits that are non-shadowed
5	Y	0x1	**POST_RUNAWAY_EMBED**: Same as previous field, except for embedded data.
4	Y	0x1	**POST_RUNAWAY_PIXEL**: Always flag a runaway error even a previous type of error has occurred during frame. A RUNAWAY error occurs when too many lines are detected in the frame. Therefore RUNAWAY errors necessarily occur after EOF and therefore will not be flagged if another type of error occurred during the course of frame. This field is for non-embeded data runaways. Value of 1 means always post, 0 only if it is first frame error. The disadvantage of turning this feature on is that it places more pressure on the notification FIFO from CHANSEL.
3	Y	0x0	**EARLY_ABORT**: If LOAD command was received during an on going frame of an ENABLEd channel then frame is immediately aborted with an FE at the time of the LOAD arrival. Otherwise frame is left to finish normally and shadow operation occurs at FE packet. If a channel is disabled when LOAD arrives from host then this bit is ignored and any ongoing frame would always be aborted.
2	Y	0x0	**SINGLESHOT_AUTO**: 1 if this channel should automatically set SINGLESHOT to 1 when an autoload event occurs (either AUTOLOAD or deferred-LOAD.)
1	Y	0x0	**SINGLESHOT**: Set to 1, if this channel should automatically set ENABLE to 0 when an EOF that was preceded by an SOF arrives at CHANSEL.

VI_CH35_MATCH_0

STREAM and VIRTUAL CHANNELS are encoded on a hot bit per stream scheme.

Example: only STREAM 2 is desired then STREAM =6''b000100 (0x4 or (1U << 2) needs to be programmed with mask STREAM_MASK=6''b111111 (0x3F or (1U << VI_STREAMS)-1)

Offset: 0x6307
Byte Offset: 0x18c1c
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00003f0f (0bxxxx,xxxx,xxxx,0000,0011,1111,0000,1111)

Bit	Reset	Description
19:14	0x0	**STREAM**
13:8	0x3f	**STREAM_MASK**
7:4	0x0	**VIRTUAL_CHANNEL**
3:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH35_MATCH_DOL_0

Offset: 0x6308
Byte Offset: 0x18c20
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xx00,0000,0000,0000)

Bit	Reset	Description
29:16	0x0	**DOL**
13:0	0x0	**DOL_MASK**

VI_CH35_MATCH_VC_HI_0

Offset: 0x6309
Byte Offset: 0x18c24
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
23:12	0x0	**VIRTUAL_CHANNEL**
11:0	0xf	**VIRTUAL_CHANNEL_MASK**

VI_CH35_MATCH_DATATYPE_0

Offset: 0x630a
Byte Offset: 0x18c28
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0011,1111)

Bit	Reset	Description
11:6	0x0	**DATATYPE**
5:0	0x3f	**DATATYPE_MASK**

VI_CH35_MATCH_FRAMEID_0

Offset: 0x630b
Byte Offset: 0x18c2c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	Reset	Description
31:16	0x0	**FRAMEID**
15:0	0xffff	**FRAMEID_MASK**

VI_CH35_DT_OVERRIDE_0

When DT_OVRD_EN=DISABLE all incoming pixels types, except embedded (CSI_DT_GED_ED), will be mapped to the type of OVRD_DT. The one exception is embedded data type which does not get remapped through this register. If USER data types are expected on the channel then programming must enable this feature and map to a valid type.

Note that if a channel uses data type interleaving it is most likely not to have this feature on. You are also likely not to have a wildcard but an exact match for data type.

It is illegal to program OVRD_DT to a user data type (CSI_DT_UED_U{1,2,3,4} or CSI_DT_UED_R{1,2,3,4}). Improper programming of this register can result in an DTYPE_MISMATCH error.

The following explains how CHANSEL treats incoming data type in conjunction with the override register:

a) LOCK FILTER: Prior to override CHANSEL searches for an LS pixel (line start). Once a LS has been found the pixel

data type is saved and locked: call this itype LOCK_DT. The lock-in persists until an LE is determined.

After LE occurs, the search for a new LS and new LOCK_DT starts. The LOCK_DT data type is determined

prior to any override mapping. Should any incoming pixel after LS not match the lock in type then

a DTYPE_MISMATCH is flagged and the non-matching pixel is forced to the LOCK_DT type.

This applies any incoming type, included embedded. The LOCK filter protects against CSIMUX FIFO overflow.

b) OVERRIDE filter: if DT_OVRD_EN=ENABLE then incoming pixels are converted to OVRD_DT unless they are of the embedded type.

If DT_OVRD_EN=DISABLE and pixel is found to have a user data type then all pixels for the line will be converted

to CSI_DT_RAW_16 (8PPC) and a DTYPE_MISMATCH fault will be thrown.

c) Once a data type is finalized it will be determined whether the type is 8PPC or 4PPC. If pixel enables

are not consistent with the 8PPC or 4PPC expectation then a correction to the pixel enables will be made and

a DTYPE_MISMATCH fault will be thrown.

d) Unused pixels will be cleared out using the pixel enables.

Offset: 0x630c
Byte Offset: 0x18c30
Read/Write: R/W
Parity Protection: Y
Shadow: Y

SCR Protection: 0
Reset: 0x0000005c (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x101,1100)

Bit	Reset	Description
6:1	0x2e	**OVRD_DT**: Datatype to override input format with.
0	0x0	**DT_OVRD_EN**: Enable input format override **0** = DISABLE **1** = ENABLE

VI_CH35_FRAME_X_0

Expected frame x dimension prior to any cropping from CHANSEL (see SKIP_X and CROP_X registers below)

Offset: 0x630d
Byte Offset: 0x18c34
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of frame, if 0x0 then 2^16

VI_CH35_FRAME_Y_0

Expected frame y dimension

Offset: 0x630e
Byte Offset: 0x18c38
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of frame, if 0x0 then 2^16

VI_CH35_EMBED_X_0

Applies when DTYPE==CSI_DT_GED_ED

Upper bound of the number of bytes on an embedded line, if 0x0 then 2^17

Offset: 0x630f
Byte Offset: 0x18c3c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x0	**MAX_BYTES**: Maximum number of embedded data bytes on a line

VI_CH35_EMBED_Y_0

Applies when DTYPE==CSI_DT_GED_ED

Embedded lines configuration -- the EXPECTED field will not block embedded data.

it is used for reporting an error when embedded data is present but not expected.

if there is no embedded data on particular frame and the expect field is true then no error is reported.

Offset: 0x6310
Byte Offset: 0x18c40
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether embedded data should be processed (i.e., if set, embedded data is transmitted downstream). Note that EXPECT and ENABLE are handled independently.
24	0x0	**EXPECT**: Embedded data expected within channel.
15:0	0x0	**LINES**: If EXPECTED=1: Number of embedded lines in frame, where 0x0 implies 2^16.

VI_CH35_FLUSH_0

Flush and Line Timer Events

Send FLUSH signals to ATOM PACKER so that it can flush out partial frames of pixels data. This is done setting a tripline count which will flag a flush to the MW after the number of lines has been reached. The count can be setup to occur periodically. This feature does not count embedded data, however any embedded data on the same channel that is processed prior to a tripline event will also be flushed.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. However flush events are filtered by the cropped region. So the programmer must ensure events occur within the output cropped region.

Offset: 0x6311
Byte Offset: 0x18c44
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a flush notice sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a flush notice is sent out, 0x0 implies 2^16

VI_CH35_LINE_TIMER_0

Send notifications to CPU indicating frame progression. Based on notification certain processes can be started. For example starting a flash in anticipation of an upcoming frame, or setup up a focusing process. This is done setting a tripline count which flags a notification once the number of lines has been reached. The count can be setup to occur on a repeating period of TRIPLINE lines. This feature does not count embedded data.

Note that the TRIPLINE and PERIODIC events are calculated on the input line count, not the output cropped lines. The line timer events are not filtered by the cropped region, they are solely based on the input frame. The notifications are sent line end, but the X position of the crop X coordinate if there is any.

Offset: 0x6312
Byte Offset: 0x18c48
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ENABLE**: Whether to turn off feature.
24	0x0	**PERIODIC**: Whether a notification sould be sent for the first tripline or repeatedly.
15:0	0x0	**TRIPLINE**: Pixel line count at which a notification is sent out, 0x0 implies 2^16

VI_CH35_SKIP_X_0

Image cropping registers (cropping does not apply to embedded data)

Number of 8 pixel packets to SKIP or drop at start of line, 0 means no packets dropped at start of line.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. For 4 PPC mode this still represents 8 pixels (i.e. 2 physical packets on the internal VI pixel bus).

Offset: 0x6313
Byte Offset: 0x18c4c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:0	0x0	**PACKETS**: Number of packets to skip on output at start of line, counted in groups of 8 pixels

VI_CH35_CROP_X_0

Last pixel position on line before following pixels of line are dropped, pixels positions are numbered from 1.

Ensure that 1. (NV_VI_PPC=8)*SKIP_X_PACKETS < CROP_X_WIDTH

2. (NV_VI_PPC=8)*SKIP_X_PACKETS < FRAME_X_WIDTH

It is not necessary that CROP_X_WIDTH <= FRAME_X_WIDTH, if CROP_X_WIDTH >= FRAME_X_WIDTH then no right crop. Ensure CROP_X_WIDTH % 8 == 0 when DPCM is in use. If 0 means 2^16 = 65536, or no cropping on right edge.

Offset: 0x6314
Byte Offset: 0x18c50
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Line width in pixels after which no packets will be transmitted

VI_CH35_SKIP_Y_0

Width parameter includes pixels dropped by SKIP_X

Number of pixels lines to skip at top of frame, 0 means no lines dropped at top of frame.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

Offset: 0x6315
Byte Offset: 0x18c54
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**LINES**: Number of lines to skip at top of the frame

VI_CH35_CROP_Y_0

Last line position before following lines are dropped, line are numbered from 1.

Ensure that 1. SKIP_Y_LINES < CROP_Y_HEIGHT

2. SKIP_Y_LINES < FRAME_Y_HEIGHT

It is not necessary that CROP_Y_HEIGHT <= FRAME_Y_HEIGHT, if CROP_Y_HEIGHT >= FRAME_Y_HEIGHT then no bottom crop.

0 means 2^16 = 65536, or no cropping on bottom edge.

Offset: 0x6316
Byte Offset: 0x18c58
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Height in lines after which no lines will be transmitted

VI_CH35_OUT_X_0

HEIGHT parameter includes lines skipped by SKIP_Y

Expected frame width after cropping from CHANSEL (see SKIP_X and CROP_X above)

If no cropping then OUT_X==FRAME_X and OUT_Y==FRAME_Y)

In other words:

OUT_X = MINIMUM(FRAME_X_WIDTH, CROP_X_WIDTH) - 8*SKIP_X_PACKETS

OUT_Y = MINIMUM(FRAME_Y_HEIGHT,CROP_Y_HEIGHT) - SKIP_Y_LINES

Offset: 0x6317
Byte Offset: 0x18c5c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**WIDTH**: Pixel Width of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH35_OUT_Y_0

Offset: 0x6318
Byte Offset: 0x18c60
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**HEIGHT**: Line Height of cropped frame at output of chansel, if 0x0 then 2^16

VI_CH35_NOTIFY_MASK_0

Offset: 0x6319
Byte Offset: 0x18c64
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**DTYPE_MISMATCH**: Data type mismatch: switched type during line, non-overriden user type or bad pixel enables Embepdded data byte related events
22	0x0	**EMBED_INFRINGE**: Unexpected embedded data in frame
21	0x0	**EMBED_LONG_LINE**: Extra bytes on line
20	0x0	**EMBED_SPURIOUS**: Embedded bytes found between line start and line end
19	0x0	**EMBED_RUNAWAY**: Too many embeded lines in frame
18	0x0	**EMBED_MISSING_LE**: Two embedded line starts without a line end in between
17	0x0	**EMBED_EOF**: Last byte of embedded data
16	0x0	**EMBED_SOF**: First byte of embedded data Pixel data related events
7	0x0	**PIXEL_LINE_TIMER**: Line counting event, see LINE_TIMER register
6	0x0	**PIXEL_SHORT_LINE**: A line has fewer pixels than expected width
5	0x0	**PIXEL_LONG_LINE**: A line has more pixels than expected width, pixels dropped
4	0x0	**PIXEL_SPURIOUS**: A pixel found between line end and line start markers, dropped

Bit	Reset	Description
3	0x0	**PIXEL_RUNAWAY**: Too many pixel lines in frame, extra lines dropped
2	0x0	**PIXEL_MISSING_LE**: Two lines starts without a line end in between
1	0x0	**PIXEL_EOF**: Last pixel of frame
0	0x0	**PIXEL_SOF**: First pixel of frame

VI_CH35_NOTIFY_MASK_XCPT_0

Events that can occur on zero, one or more streams at once (exceptions)

If a NOMATCH occurs concurrently to a LOAD_FRAMED, the NOMATCH is reported embedded to LOAD_FRAMED. The expected use of NOMATCH is for debug purposes.

If a LOAD_FRAMED occurs simultaneous to a SHORT_FRAME or COLLISION fault then the LOAD FRAME notification will be embedded on the payload of the SHORT_FRAME notification. To mask NOMATCH it suffices to program the mask on any single channel.

Offset: 0x631a
Byte Offset: 0x18c68
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	0x0	**NOMATCH**: Do not report on pixel preceeded by a valid frame start but for which no channel matches
8	0x0	**EMBED_OPEN_LINE**: Frame end occurred while still processing embedded byte line
7	0x0	**PIXEL_OPEN_LINE**: Frame end (FE) occurs while still processing a pixel line
6	0x0	**FORCE_FE**: Pixels stopped, an FE was forced due to a latent LOAD event (see EARLY_ABORT).
5	0x0	**STALE_FRAME**: Do not report on channels that did not receive a LOAD prior to each frame start (FS)

Bit	Reset	Description
4	0x0	**COLLISION**: Do not report on frames that match a channel already processing another frame
3	0x0	**EMPTY_FRAME**: Do not report on frames without any pixels
2	0x0	**EMBED_SHORT_FRAME**: Do not report on frames that have received partial embedded data
1	0x0	**PIXEL_SHORT_FRAME**: Do not report on frames that have received partial pixels
0	0x0	**LOAD_FRAMED**: Do not fault on LOAD that occur while in frame

VI_CH35_FRAME_SOURCE_0

Non-Secure Register Indicating Status of last frame seen on channel

Status of current or last frame seen. IN_FRAME bit determines if the frame is current or has elapsed.

Offset: 0x631b
Byte Offset: 0x18c6c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**IN_FRAME**: In frame status
26	X	**VPR**: VPR status of frame
25:22	X	**VC**: Virtual Channel within stream (binary)
21:16	X	**STREAM_HOT**: Stream 1-hot encoded
15:0	X	**FRAMEID**: Frame ID

VI_CH35_FRAME_COUNT_0

The FS and FE count should match (FS=FE) unless channel in the middle of processing a frame.

If in frame then FS=FE+1. FE can be perceived as a checksum check for FS.

Write any value to register to clear all counts.

Offset: 0x631c
Byte Offset: 0x18c70
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BAD**: Count of faulty frame end seen, saturates at 255
23:12	X	**FE**: Count of all frame end seen, saturates at 4095
11:0	X	**FS**: Count of all frame start seen, saturates at 4095

VI_CH35_LINE_TIMER_FIRST_0

Offset: 0x631d
Byte Offset: 0x18c74
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first timer TRIPLINE should be selected when setting timer value at start-of-frame. If SEL is 0 then LINE_TIMER.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first line timer event.

VI_CH35_FLUSH_FIRST_0

Offset: 0x631e
Byte Offset: 0x18c78
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**SEL**: Whether the first flush TRIPLINE should be selected when setting flush value at start-of-frame. If SEL is 0 then FLUSH.TRIPLINE shall be used.
15:0	0x0	**TRIPLINE**: Line count at which to trip the first flush event.

VI_CH35_DOL_CTRL_0

Offset: 0x631f
Byte Offset: 0x18c7c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**HEADER_SEL**: pixel HEADER_SEL in first packet will be the DOL header

VI_CH35_PIXEL_SHORT_FRAME_STATUS_0

Offset: 0x6320
Byte Offset: 0x18c80
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PIXEL_SHORT_FRAME**: indicate whether pixel short frame happened on current channel; write 1 to clear;

Bit	R/W	Reset	Description
30:16	RO	0x0	**FRAME_ID**: The FrameID along with PIXEL_SHORT_FRAME event; Only 15-bit LSB is saved here; while full FrameID can be retrieved in notify info. Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field
15:0	RO	0x0	**LINE_COUNT**: Number of lines received before FE packet; (0 means empty frame, 1 means only single line is received, etc.) Only saves the 1st FrameID if PIXEL_SHORT_FRAME happened multi-times, unless this is cleared by writing 1 to "PIXEL_SHORT_FRAME" field

VI_CH35_PIXFMT_ENABLE_0

Disable PROD chk as SW driver will set the right values

Offset: 0x6321
Byte Offset: 0x18c84
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x001)

Bit	Reset	Description
2	0x0	**PDAF_EN**: This bit enables the PDAF separation for this channel
1	0x0	**COMPAND_EN**: This bit enables companding unit for this channel **0** = DISABLE **1** = ENABLE
0	0x1	**PIXFMT_EN**: This bit enables PIXFMT writing pixels for this channel **0** = DISABLE **1** = ENABLE

VI_CH35_PIXFMT_FORMAT_0

Offset: 0x6322
Byte Offset: 0x18c88
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,0000,0101)

Bit	Reset	Description
17:16	0x0	**T_R16_R32_PAD0_EN**: Enable padding 0 to T_R16/T_R32 memory format. Valid for RAW10/ RAW12/RAW14 of T_R16, RAW20/RAW24 of T_R32. 1: data is right aligned, msb padding 0 2: data is left aligned, lsb padding 0
7:0	T_R8	**FORMAT**: Pixel memory format for the VI channel **1** = T_R5G6B5 **2** = T_B5G6R5 **5** = T_R8 **8** = T_A8B8G8R8 **9** = T_A8R8G8B8 **10** = T_B8G8R8A8 **11** = T_R8G8B8A8 **16** = T_Y8_U8__Y8_V8 **17** = T_Y8_V8__Y8_U8 **18** = T_V8_Y8__U8_Y8 **19** = T_U8_Y8__V8_Y8 **34** = T_Y8__U8V8_N420 **35** = T_Y8__V8U8_N420 **42** = T_B5G5R5A1 **43** = T_R5G5B5A1 **44** = T_Y8__U8V8_N422 **45** = T_Y8__V8U8_N422 **46** = T_Y8__U8__V8_N422 **47** = T_Y8__U8__V8_N420 **64** = T_DPCM_RAW10 **68** = T_A2B10G10R10 **69** = T_A2R10G10B10 **70** = T_B10G10R10A2 **71** = T_R10G10B10A2 **80** = T_A4B4G4R4 **81** = T_A4R4G4B4 **82** = T_B4G4R4A4 **83** = T_R4G4B4A4 **84** = T_A1B5G5R5 **85** = T_A1R5G5B5 **98** = T_Y10__V10U10_N420 **99** = T_Y10__U10V10_N420 **100** = T_Y10__U10__V10_N420 **101** = T_Y10__V10U10_N422 **102** = T_Y10__U10V10_N422 **103** = T_Y10__U10__V10_N422 **128** = T_DPCM_RAW12 **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **197** = T_R16_I **198** = T_R16_X_ISP24 **210** = T_R24 **230** = T_R32 **232** = T_R32_F **254** = T_DPCM_RAW16 **255** = T_DPCM_RAW20

VI_CH35_PIXFMT_WIDE_0

Offset: 0x6323
Byte Offset: 0x18c8c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**ENDIAN**: which order to merge adjacent pixels in 0 = Big Endian 1 = Little Endian **0** = BIG **1** = LITTLE
0	0x0	**ENABLE**: Whether to merge adjacent RAW8/RAW10/RAW12 pixels

VI_CH35_PIXFMT_PDAF_CROP_X_0

Offset: 0x6324
Byte Offset: 0x18c90
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RIGHT**: Within a line, X pixel position at which PDAF separation ends
15:0	0x0	**LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH35_PIXFMT_PDAF_CROP_Y_0

Offset: 0x6325
Byte Offset: 0x18c94
Read/Write: R/W

Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BOTTOM**: line at which PDAF separation ends
15:0	0x0	**TOP**: line at which PDAF separation begins

VI_CH35_PIXFMT_PDAF_CONFIG0_0

Offset: 0x6326
Byte Offset: 0x18c98
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:12	0x0	**REPLACE_VALUE**: Value to replace PDAF pixel with (in nvcsi2vi pixel bus format)
0	0x0	**REPLACE_ENABLE**: Whether to replace PDAF pixels sent to primary surface with an alternative value.

VI_CH35_PIXFMT_PDAF_REPLACE_CROP_X_0

Offset: 0x6327
Byte Offset: 0x18c9c
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_RIGHT**: Within a line, X pixel position at which PDAF separation ends

Bit	Reset	Description
15:0	X	**REPLACE_LEFT**: Within a line, X pixel position at which PDAF separation begins

VI_CH35_PIXFMT_PDAF_REPLACE_CROP_Y_0

Offset: 0x6328
Byte Offset: 0x18ca0
Read/Write: R/W
Parity Protection: N
Shadow: Y
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**REPLACE_BOTTOM**: line at which PDAF separation ends
15:0	X	**REPLACE_TOP**: line at which PDAF separation begins

VI_CH35_PIXFMT_PDAF_LAST_PIXEL_0

Location of last pixel to be output in PDAF.

Offset: 0x6329
Byte Offset: 0x18ca4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**Y**: Y coordinate of last PDAF pixel within the PDAF crop window
15:0	0x0	**X**: X coordinate of last PDAF pixel within the PDAF crop window

VI_CH35_PIXFMT_PDAF_CONFIG1_0

Offset: 0x632a
Byte Offset: 0x18ca8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000000c4 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1100,0100)

Bit	Reset	Description
7:0	T_R16	**PDAF_FORMAT**: Memory format in which the PDAF pixels will be written in. **194** = T_R16_X_ISP20 **195** = T_R16_F **196** = T_R16 **230** = T_R32 **232** = T_R32_F

VI_CH35_DPCM_STRIP_0

Offset: 0x632f
Byte Offset: 0x18cbc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**OVERFETCH**: Number of *packets* in overfetch (0 to disable overfetch)

VI_CH35_DPCM_CHUNK_FIRST_0

Offset: 0x6330
Byte Offset: 0x18cc0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH35_DPCM_CHUNK_BODY_0

Offset: 0x6331
Byte Offset: 0x18cc4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**COUNT**
15:0	0x0	**PACKETS**

VI_CH35_DPCM_CHUNK_PENULTIMATE_0

Offset: 0x6332
Byte Offset: 0x18cc8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH35_DPCM_CHUNK_LAST_0

Offset: 0x6333
Byte Offset: 0x18ccc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PACKETS**

VI_CH35_DPCM_STATISTICS_0_0

Offset: 0x6334
Byte Offset: 0x18cd0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_COMPRESSED**: Number of compressed bits output in previous frame

VI_CH35_DPCM_STATISTICS_1_0

Offset: 0x6335
Byte Offset: 0x18cd4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TOTAL_COMPRESSED**: Number of total compressed bits output. (write to clear)

VI_CH35_DPCM_MODE_0

Offset: 0x6336
Byte Offset: 0x18cd8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	RAW10	**MODE**: **0** = RAW10 **1** = RAW12 **2** = RLE_RAW10 **3** = RLE_RAW12 **4** = RAW16 **5** = RAW20

VI_CH35_DPCM_CLAMP_HIGH_0

Offset: 0x6337
Byte Offset: 0x18cdc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x000fffff (0bxxxx,xxxx,xxxx,1111,1111,1111,1111,1111)

Bit	Reset	Description
19:0	0xfffff	**CLAMP_VALUE**

VI_CH35_DPCM_CLAMP_LOW_0

Offset: 0x6338
Byte Offset: 0x18ce0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**CLAMP_VALUE**

VI_CH35_AXISTREAMID_CFG_0

NOTE: AXI STREAM ID INDEX, not VI STREAM ID

Disable PROD chk as SW driver will set the right values

Offset: 0x633b
Byte Offset: 0x18cec
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**STREAMID**: Indicate SMMU context, the initial value will correspond to global STREAMID Index into Falcon STREAMID table for this channel.

VI_CH35_ATOMP_SURFACE_OFFSET0_0

Offset: 0x633c
Byte Offset: 0x18cf0
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH35_ATOMP_SURFACE_OFFSET0_H_0

Offset: 0x633d
Byte Offset: 0x18cf4
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH35_ATOMP_SURFACE_STRIDE0_0

Offset: 0x633e
Byte Offset: 0x18cf8
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH35_ATOMP_DPCM_CHUNK_0

Offset: 0x633f
Byte Offset: 0x18cfc
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH35_ATOMP_SURFACE_OFFSET1_0

Offset: 0x6340
Byte Offset: 0x18d00
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH35_ATOMP_SURFACE_OFFSET1_H_0

Offset: 0x6341
Byte Offset: 0x18d04
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH35_ATOMP_SURFACE_STRIDE1_0

Offset: 0x6342
Byte Offset: 0x18d08
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH35_ATOMP_SURFACE_OFFSET2_0

Offset: 0x6343
Byte Offset: 0x18d0c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH35_ATOMP_SURFACE_OFFSET2_H_0

Offset: 0x6344
Byte Offset: 0x18d10
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH35_ATOMP_SURFACE_STRIDE2_0

Offset: 0x6345
Byte Offset: 0x18d14
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH35_ATOMP_EMB_SURFACE_OFFSET0_0

Offset: 0x6346
Byte Offset: 0x18d18
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFSET**

VI_CH35_ATOMP_EMB_SURFACE_OFFSET0_H_0

Offset: 0x6347
Byte Offset: 0x18d1c
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**OFFSET_HI**

VI_CH35_ATOMP_EMB_SURFACE_STRIDE0_0

Offset: 0x6348
Byte Offset: 0x18d20
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**STRIDE**

VI_CH35_ATOMP_DVFSFIFO_WATERMARK_0

Offset: 0x6349
Byte Offset: 0x18d24
Read/Write: R/W
Parity Protection: Y
Shadow: Y
SCR Protection: 0
Reset: 0x00008030 (0bxxxx,xxxx,xx00,0000,1000,0000,0011,0000)

Bit	Reset	Description
21:11	0x10	**LOW_WATERMARK**
10:0	0x30	**HIGH_WATERMARK**

VI_CH35_ATOMP_CHANNEL_SUPPORT_0

Offset: 0x634a
Byte Offset: 0x18d28
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**PDAF**
2	X	**SEMI_PLANAR**
1	X	**PLANAR**
0	X	**BAYER**

VI_CH35_ATOMP_HIGH_PRI_REQ_0

Offset: 0x634b
Byte Offset: 0x18d2c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2:1	0x3	**COUNT**
0	0x1	**ENABLE**

VI_CH35_ATOMP_RESERVE_0

Offset: 0x634c
Byte Offset: 0x18d30
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVE**

7.3 Video Image Compositor (VIC)

7.3.1 Overview

The Video Image Compositor (VIC) supports the following features:

- Geometry transform processing for Lens Distortion Correction
  - Programmable nine-point controlled warp patch for distortion correction
  - Real-time on-the-fly position generation from sparse warp map surface
  - Pin-cushion, barrel, or moustache distortion correction
  - Distortion correction of 180- and 360-degree wide field of view (FOV) lens
  - Scene perspective orientation adjustment with IPT
  - Full warp map capability
  - Non-fixed Patch size with 4x4 regions
  - External Mask bit map surface
- Temporal noise reduction
  - New Bilateral Filter as Spatial Filter
  - Improved TNR3 algorithm
- Surface transform processing
  - Surface transformation: Flip/Rotation/Mirror
  - Scaling and filtering:
    - 1, 2, 5, or 10-tap polyphase filters
    - Programmable filter coefficients
    - 1/32 sub-pixel precision
    - Linear interpolation for filter-type (detail/noise/normal) and scale ratio (1:1, 2:1, 4:1, 8:1, 16:1)
    - Sub-picture and normal mode for display subtitles as well as menu buttons.
  - Filter override mode with exact filter kernel for the first five slots
  - Panoramic scaling for different aspect ratios image conversion
  - Detail filter clamping to avoid large ghosting effect
  - Sub-pixel source rectangles to smoothly zoom into image portion
- Pixel Format Processing

    - Surface format conversion: Example: Pitch Linear -> Block Linear
    - Programmable GOB height
    - Extensive pixel format support
      - Component number: T_L8 / T_Y8___U8V8
      - Color space: AYUV or ARGB
      - Bit depths of 8, 10, 12, or 16-bit
      - Chroma formats: YUV420, YUV422, or YUV444
      - Color gamuts: Rec.2020, Rec.709, or sRGB
      - Memory layouts: Planar or Semi-planar
    - Pixel format conversion
      - 4x3 matrix multiplication conversion with soft clamping; Example: RGB -> YUV
      - Color space conversion; Example: REC2020 -> REC709 -> sRGB
      - Different sub-sampled chroma location; Example: (1/2, 1/2) or (0, 1/2)
      - Data size conversion with MSB bits replication: RGB565 -> RGB888
      - 2-tap or 5-tap chroma upsampling filtering
    - Color correct processing
      - YUV Scaling occurs in linear space
      - Fixed lookup table based de-gamma and re-gamma support
      - 16-bit internal data precision: S1.14 (one sign bit, one range bit, 14 fractional bits)
- Video Quality Improvement and Video Edit Tool
    - De-interlacing with DiSi1, BOB, and WEAVE modes for interlaced video playback cases
    - Inverse telecine with 3:2 pull down from 29.97 fps to 24 fps
    - Cadence detection
    - Temporal noise reduction for low-end YUV USB camera
    - Luma keying with the programmable range to extract the region of interest
- Blenders and Compositor
    - Maximum resolution 16384x16384 is supported
    - 16 slots for multiple-stream composition, for example Blu-Ray content
    - Eight clear rectangles
    - Parameterized blender interface enables a variety of blending modes
    - Color Filling the output surface with a programmable background color

7.3.1.1 Use Cases

The VIC key use cases can be put into three main categories: Geometry Processing, 2D operations, and Video Post-processing operations.

7.3.1.1.1 Geometry Processing for Lens Distortion Correction

VIC can perform lens distortion correction on fisheye sensors and other wide field-of-view lenses. VIC can also do stereo rectification on input stereo lens frames.

7.3.1.1.2 2D Operations for UI Composition

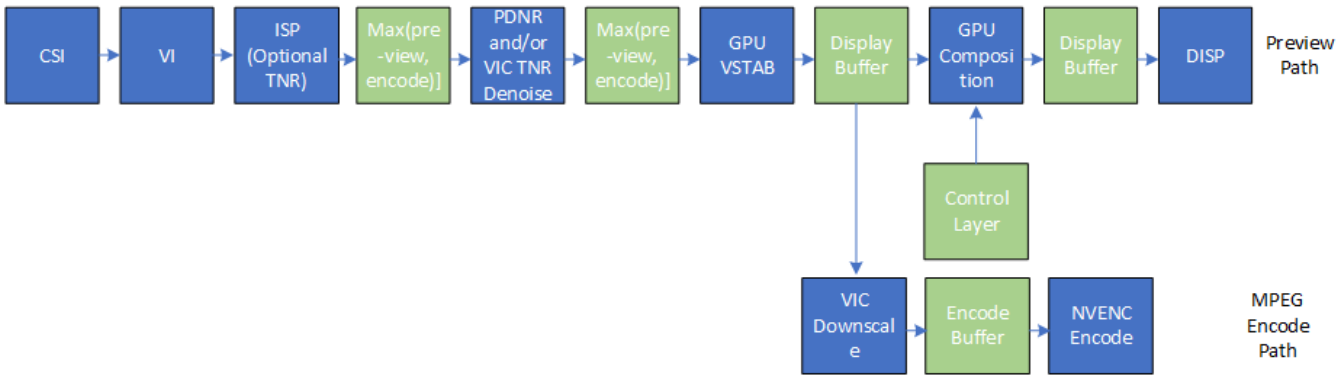
VIC can do multi-surface composition and blending for a UI, so that VIC can be used instead of the GPU in some use cases.

7.3.1.1.3 Camera Still and Video Capture

An example data flow diagram for camera use cases is referenced in the data flow diagram.

- Downscale and rotate still of video images for generating preview images.
- Crop, upscale, and rotate still of video images for digital zoom previews.
- Stabilize video by providing slight upscaling and cropping of an incoming video stream.

Figure 7.19 Camera Data Flow Diagram



7.3.1.2 Hardware Description

VIC hardware is composed of two parts: the Compositor and the LDC_TNR Geometry-transformer.

7.3.1.2.1 Compositor

The Host Interface (THI) handles incoming methods from the Host Controller Bus and passes those methods to the embedded Falcon microcontroller, in addition to handling sync points. The Falcon Microcontroller is used for the host interaction (methods, context switches), controls the hardware via config space, and handles the interrupt from hardware logic.

Some of the actions are:

- The AXI Frame Buffer Interface (AFBIF) handles memory read and write requests and returns the data to the clients.
- The Fetch Control unit calculates the fetch region of the input surface and sends requests to the Surface Cache.
- The Surface Cache handles different input surface overfetch, does the internal pixel format conversion, and feeds the pixel data into the hardware pipeline.
  - YUV chroma up-sampler, YUV -> RGB, Degamma, Gamut remap, Regamma, RGB -> YUV, and Chroma down sampler modules work together to handle different pixel formats and color spaces.
  - Y-scaler and X-scaler handles scale down or scale up between the input image surface and output image surface with programmable interpolation filter length.
- The Blender module blends all input surfaces with tile-by-tile scheme and 16 slot capabilities. SMEM works as one shared memory among submodules in Blender and LDC IP.
- Cadence detect, TNR DI statistics, and de-interlacer handle converting interlace video content to the progressive video content via the advanced algorithm, legacy temporal noise reduction, and inverse telecine operation.

7.3.1.2.2 LDC/TNR3

The LDC/TNR3 IP shares the same THI, Falcon, AFBIF, Surface Cache, and SMEM with the Compositor. Some of the actions are:

- The NCFG module broadcasts all register configurations to all modules and kicks-off to the hardware pipeline.
- The TileGen module handles tile64x16 vertical traversal, mask bitmap surface reading, and TNR3 input surface fetching.
- The IPT module does on-the-fly control point calculations for the inverse perspective transform mode.
- The PosGen module does position interpolation from sparse warp map to full warp map.
- The BBPF does bounding box calculations with the input position information from PosGen, and sends the data fetch requests to Surface Cache.
- PixelGen and LSCALAR modules work together to do sub-pixel interpolation with high throughput.
- The TNR3 module provides spatial bilateral filter and enhanced temporal noise reduction filter.
- The OutGen module outputs pixel data into the external memory via three channels of DMA for different pixel formats and memory layout.
- The PP module does xsobel filter and 4x4 box down sample output for rectification cases.

7.3.1.2.3 Configurable Surface Cache Dimensions

VIC exposes a configurable logical cache line size to software for each input slot. Because the cache height and width dimensions is chosen independent of the surface data memory layout, the 256 bytes of data fetched into the surface cache entry may not be found in contiguous memory. For example, if a cache entry is programmed to logically represent 64x4 bytes of pixel data, and the source data is fetched from a pitch linear surface, then the data needs to be fetched from four non-contiguous 64-byte segments of memory. While a single cache line may now include data from as many as four non-contiguous segments of memory, the tag for each Surface Cache entry continues to point to the data located at the top left corner of the pixels contents within the cache line.

The logical width and height of each surface cache entry can now represent pixel data in the following configurations:

1. 256x1 bytes
2. 128x2 bytes
3. 64x4 bytes
4. 32x8 bytes
5. 16x16 bytes

When programming the logical cache height and width, the dimensions should be chosen to minimize over-fetch based on the function for which the data is used (e.g., de-interlacing, scaling with various numbers of tap filters, etc.). The logical cache height should be chosen to be a multiple of the source surface memory layout height given the memory system fetch atom height, and similarly, the logical cache width should be chosen to be a multiple of the source surface memory layout width given the memory system fetch atom width. The table below summarizes these requirements.

Table 7.32 Programmable Surface Cache Dimensions

Cache width (as programmed in ConfigStruct)	Logical cache-line dimensions	Buffer width restriction	Luma/Chroma height restriction
0	16Bx16	Multiple of 16B	Multiple of 16 lines
1	32Bx8	Multiple of 32B	Multiple of 8 lines
2	64Bx4	Multiple of 64B	Multiple of 4 lines
3	128Bx2	Multiple of 128B	Multiple of 2 lines
4	256Bx1	Multiple of 256B	N/A

For the VIC pre-process pass (noise reduction, motion adaptation values, cadence detection, etc.), the hardware has to fetch 16Bx16 sections of memory and the cache width is forced to 0 by the

Falcon (only for the pre-process pass). Using pitch linear buffers for surfaces that require pre-processing is therefore strongly discouraged.

7.3.1.2.4 Maximum Surface and Rectangle Size Support

VIC supports surface and rectangle sizes up to 16384x16384 pixels. All configurable surface width and height parameters, and configurable rectangle dimensions in the Fetch Control, Surface List, and Output Buffer, support up to 16384x16384 surfaces and rectangles. All related fields in the ConfigStruct and associated registers are 14 bits to support this.

7.3.1.3 Pixel Format Conversion

7.3.1.3.1 Memory Format Support

The following memory formats are supported on input and output. Note that input and output memory formats do not have to match.

1. Pitch Linear
2. Block Linear (16Bx2 kind with sector ordering)
3. 64x8 byte Gob format with 16x2 byte sectors
4. Contiguous 64B atoms are laid out as 32x2 byte blocks

Note: Refer to the Pixel Memory Formats chapter for more details.

The VIC Surface Cache supports the above memory formats on input when fetching pixel data from memory. The VIC Output Buffer supports the above memory formats on input when reading background image data, when pre-processing inverse telecine data from memory, and on the output side when writing composited image data back to memory.

Memory formats supported by VIC are enumerated as follows:

Table 7.33 Memory Format Enumerations

Surface Kind Enumeration	Encoding (four bits)
BLK_KIND_PITCH	0x0
BLK_KIND_GENERIC_16Bx2_TEGRA	0x1

To fully support the Block Linear format, block height should be programmable to any one of the following set of enumerations.

Table 7.34 Block Height Enumerations

Block Height Enumeration	Value	Encoding (four bits, log2 encoding)
ONE_GOB	1	0
TWO_GOBS	2	1
FOUR_GOBS	4	2
EIGHT_GOBS	8	3
SIXTEEN_GOBS	16	4
THIRTYTWO_GOBS	32	5

7.3.1.3.2 Pixel Format Support

The VIC pipeline supports a large number of pixel formats. The VIC pipeline accepts a surface of a given input pixel format and produces a given output pixel format. The following table lists all the pixel formats supported in VIC and any restrictions on their usage. The symbols in color conversion mode column describe the internal VIC representation of the format, (e.g., 1P_1C represents one plane with one component pixels, 2P_1+2C represents two planes, one plane with one component pixels, and a second plane with two component pixels).

Table 7.35 Pixel Format Support

Pixel Format	FOURCC	Input Support	Output Support	Rotation	Scaling	High Quality De-interlacing	TNR/ TNR2
T_A8		Y	Y	Y	Y	N	N
T_L8/ T_Y8/ T_YUV100		Y	Y	Y	Y	N	N
T_A4L4		Y	Y	Y	Y	N	N
T_L4A4		Y	Y	Y	Y	N	N
T_R8		Y	Y	Y	Y	N	N
T_A8L8		Y	Y	Y	Y	N	N
T_L8A8		Y	Y	Y	Y	N	N
T_L16		Y	Y	Y	Y	N	N
T_R8G8		Y	Y	Y	Y	N	N
T_G8R8		Y	Y	Y	Y	N	N
T_B5G6R5		Y	Y	Y	Y	N	N
T_R5G6B5		Y	Y	Y	Y	N	N
T_B6G5R5		Y	Y	Y	Y	N	N

Pixel Format	FOURCC	Input Support	Output Support	Rotation	Scaling	High Quality De-interlacing	TNR/ TNR2
T_R5G5B6		Y	Y	Y	Y	N	N
T_A1B5G5R5		Y	Y	Y	Y	N	N
T_A1R5G5B5		Y	Y	Y	Y	N	N
T_B5G5R5A1		Y	Y	Y	Y	N	N
T_R5G5B5A1		Y	Y	Y	Y	N	N
T_A5B5G5R1		Y	Y	Y	Y	N	N
T_A5R1G5B5		Y	Y	Y	Y	N	N
T_B5G5R1A5		Y	Y	Y	Y	N	N
T_R1G5B5A5		Y	Y	Y	Y	N	N
T_X1B5G5R5		Y	Y	Y	Y	N	N
T_X1R5G5B5		Y	Y	Y	Y	N	N
T_B5G5R5X1		Y	Y	Y	Y	N	N
T_R5G5B5X1		Y	Y	Y	Y	N	N
T_A4B4G4R4		Y	Y	Y	Y	N	N
T_A4R4G4B4		Y	Y	Y	Y	N	N
T_B4G4R4A4		Y	Y	Y	Y	N	N
T_R4G4B4A4		Y	Y	Y	Y	N	N
T_A8B8G8R8		Y	Y	Y	Y	N	N
T_A8R8G8B8		Y	Y	Y	Y	N	N
T_B8G8R8A8		Y	Y	Y	Y	N	N
T_R8G8B8A8		Y	Y	Y	Y	N	N
T_X8B8G8R8		Y	Y	Y	Y	N	N
T_X8R8G8B8		Y	Y	Y	Y	N	N
T_B8G8R8X8		Y	Y	Y	Y	N	N
T_R8G8B8X8		Y	Y	Y	Y	N	N
T_A2B10G10R10		Y	Y	Y	Y	N	N
T_A2R10G10B10		Y	Y	Y	Y	N	N
T_B10G10R10A2		Y	Y	Y	Y	N	N

Pixel Format	FOURCC	Input Support	Output Support	Rotation	Scaling	High Quality De-interlacing	TNR/ TNR2
T_R10G10B10A2		Y	Y	Y	Y	N	N
T_A16B16G16R16		Y	Y	Y	Y	N	N
T_A4P4		Y	N	Y	Y	N	N
T_P4A4		Y	N	Y	Y	N	N
T_P8A8		Y	N	Y	Y	N	N
T_A8P8		Y	N	Y	Y	N	N
T_P8		Y	N	Y	Y	N	N
T_P1		Y	N	Y	Y	N	N
T_U8V8		Y	Y	Y	Y	N	N
T_V8U8		Y	Y	Y	Y	N	N
T_A8Y8U8V8 (AYUV444 packed)		Y	Y	Y	Y	N	N
T_V8U8Y8A8 (AYUV444 packed)		Y	Y	Y	Y	N	N
T_A16Y16U16V16		Y	Y	Y	Y	N	N
T_Y8_U8__Y8_V8 (YUV422 packed)	YUY2/YUYV	Y	Y	Y	Y	Y	Y
T_Y8_V8__Y8_U8 (YUV422 packed)	YVYU	Y	Y	Y	Y	N	N
T_U8_Y8__V8_Y8 (YUV422 packed)	UYVY	Y	Y	Y	Y	Y	Y
T_V8_Y8__U8_Y8 (YUV422 packed)		Y	Y	Y	Y	N	N
T_Y8___U8V8_N444 (YUV444 semi-planar)		Y	Y	Y	Y	N	N
T_Y8___V8U8_N444 (YUV444 semi-planar)		Y	Y	Y	Y	N	N
T_Y8___U8V8_N422 (YUV422 semi-planar)	e.g., NV61	Y	Y	Y	Y	N	N
T_Y8___V8U8_N422 (YUV422 semi-planar)	e.g., NV16	Y	Y	Y	Y	N	N
T_Y8___U8V8_N422R (YUV422R semi-planar)		Y	Y	Y	Y	N	N
T_Y8___V8U8_N422R (YUV422R semi-planar)		Y	Y	Y	Y	N	N

Pixel Format	FOURCC	Input Support	Output Support	Rotation	Scaling	High Quality De-interlacing	TNR/ TNR2
T_Y8___U8V8_N420 (YUV420 semi-planar)	e.g., NV21	Y	Y	Y	Y	N	N
T_Y8___V8U8_N420 (YUV420 semi-planar)	e.g., NV12	Y	Y	Y	Y	Y	Y
T_Y8__*U8*__V8_N444 (YUV444 planar)**	e.g., YV24	Y	Y	Y	Y	N	N
T_Y8__*U8*__V8_N422 (YUV422 planar)**	e.g., YV16	Y	Y	Y	Y	N	N
T_Y8__*U8*__V8_N422R (YUV422R planar) **		Y	Y	Y	Y	N	N
T_Y8__*U8*__V8_N420 (YUV420 planar) **	e.g., YV12	Y	Y	Y	Y	Y	Y
T_Y10___U10V10_N444 (YUV444 semi-planar)		Y	Y	Y	Y	N	N
T_Y10___V10U10_N444 (YUV444 semi-planar)		Y	Y	Y	Y	N	N
T_Y10___U10V10_N422 (YUV422 semi-planar)		Y	Y	Y	Y	N	N
T_Y10___V10U10_N422 (YUV422 semi-planar)		Y	Y	Y	Y	N	N
T_Y10___U10V10_N422R (YUV422R semi-planar)		Y	Y	Y	Y	N	N
T_Y10___V10U10_N422R (YUV422R semi-planar)		Y	Y	Y	Y	N	N
T_Y10___U10V10_N420 (YUV420 semi-planar)		Y	Y	Y	Y	N	N
T_Y10___V10U10_N420 (YUV420 semi-planar)		Y	Y	Y	Y	N	Y
T_Y10__*U10*__V10_N444 (YUV444 planar)**		Y	Y	Y	Y	N	N
T_Y10__*U10*__V10_N422 (YUV422 planar)**		Y	Y	Y	Y	N	N
T_Y10__*U10*__V10_N422R (YUV422R planar) **		Y	Y	Y	Y	N	N
T_Y10__*U10*__V10_N420 (YUV420 planar) **		Y	Y	Y	Y	N	Y
T_Y12___U12V12_N444 (YUV444 semi-planar)		Y	Y	Y	Y	N	N

Pixel Format	FOURCC	Input Support	Output Support	Rotation	Scaling	High Quality De-interlacing	TNR/ TNR2
T_Y12___V12U12_N444 (YUV444 semi-planar)		Y	Y	Y	Y	N	N
T_Y12___U12V12_N422 (YUV422 semi-planar)		Y	Y	Y	Y	N	N
T_Y12___V12U12_N422 (YUV422 semi-planar)		Y	Y	Y	Y	N	N
T_Y12___U12V12_N422R (YUV422R semi-planar)		Y	Y	Y	Y	N	N
T_Y12___V12U12_N422R (YUV422R semi-planar)		Y	Y	Y	Y	N	N
T_Y12___U12V12_N420 (YUV420 semi-planar)		Y	Y	Y	Y	N	N
T_Y12___V12U12_N420 (YUV420 semi-planar)		Y	Y	Y	Y	N	Y***
T_Y12__*U12*__V12_N444 (YUV444 planar)**		Y	Y	Y	Y	N	N
T_Y12__*U12*__V12_N422 (YUV422 planar)**		Y	Y	Y	Y	N	N
T_Y12__*U12*__V12_N422R (YUV422R planar) **		Y	Y	Y	Y	N	N

* DiSi1 de-interlacing only works on NV24 surfaces and field-based variants of NV12, YV12, YUY2, and UYVY formats (with each field stored in a separate surface).
** VIC has support for three independent surface pointers for each of the input slots and output surfaces, one each for luma, chroma-u, and chroma-v.
*** 12-bit YUV pixel formats are supported for the TNR pass with only read/write support with no increase in the internal pipeline width. The pixels therefore are truncated to 10-bit precision before and after processing.
*** 12-bit YUV pixel formats for the composition pass are processed using the existing 14-bit fractional precision pipeline. This means it does not accurately de-gamma these higher precision pixel formats.

In the color format, the notation called "A:B:C" is used to describe how often U and V are sampled relative to Y.

- 444 (4:4:4) means no down sampling of the chroma channels.
- 422 (4:2:2) means 2:1 horizontal down sampling, with no vertical down sampling. Every scan line contains four Y samples for every two U and V sample pairs.
- 420 (4:2:0) means 2:1 horizontal down sampling, with 2:1 vertical down sampling.

- R refers to rotated. 422 by default means horizontal down sampling; while 422R means vertical down sampling. Instead of the chroma being shared between two horizontally adjacent luma components, it's shared between two vertically adjacent luma components.

Note: VIC does not support 24-bit color formats, either RGB or YUV.

7.3.1.1.3.3 Pixel Format Conversion

Color conversion (CC) and proc-amp are implemented using a matrix multiplication on every pixel. The matrix only changes when the surfaces changes.

Current chain of operations:

1. Proc-amp + Output CC (4x3 matrix)
2. Clamping (optional soft clamping, two control entry)

The 4x3 matrix values are specified in the MatrixStruct as S12.8 values. In addition, the result of the matrix multiplication is right shifted by *r_shift*. To allow for highest accuracy, *r_shift* should always be as high as possible without losing any range in the other coefficients. Note that the constant offsets (*c*03, *c*13, *c*23) are S12.8 and are not affected by *r_shift*.

$$\begin{bmatrix} out0 \\ out1 \\ out2 \end{bmatrix} = \begin{bmatrix} c00 & c01 & c02 & c03 \\ c10 & c11 & c12 & c13 \\ c20 & c21 & c22 & c23 \end{bmatrix} \times \begin{bmatrix} (In0 >> r_shift) \\ (In1 >> r_shift) \\ (In2 >> r_shift) \\ 1 \end{bmatrix}$$

Soft clamping defines a piece-wise linear function consisting of three pieces. The control entries define the area of soft clamping, i.e., for the values *a* and *b* we get the following pseudo-code:

```
     if (v < -a)          v' = 0
else if (-a <= v < a)     v' = (v+a)/2
else if (a <= v < b)      v' = v
else if (b <= v < (2-b)) v' = (v+b)/2
else if ((2-b) <= v)      v' = 1
```

When converting between pixel formats with components of varying bit width, bit replication is used for expanding component bit widths, and truncation is used for reducing component bit widths.

Note: Dithering and un-dithering is not supported.

Handling Sub-sampled Chroma Formats

The chroma location for pixel formats with sub-sampled chroma can be specified using the ChromaLocHoriz and ChromaLocVert parameters in the VIC config structure. Each of these parameters can take values 0, 1, or 2 representing a chroma location as listed in the table below.

Table 7.36 Example Chroma Location

ChromaLoc	Chroma Position
0	Co-sited with even luma values
1	Mid-way between even and odd luma values
2	Co-sited with odd luma values

Table 7.37 Example ChromaLoc Values

Format	ChromaLocHoriz	ChromaLocVert	Comment
4:4:4	Don't Care	Don't Care	Not sub-sampled
4:2:2 – BT601	0	Don't Care	
4:2:0 – MPEG1	1	1	
4:2:0 – MPEG2	0	1	

Data Size Conversion

When expanding the number of bits in a component (e.g., from RGB565 to RGB888), expansion should use bit replication of the MSBs of the original value to fill in the LSBs of the expanded bit length value. For example, if converting from a 5-bit value to 8-bit value, the first three MSBs of the source 5-bit value are concatenated as the three new LSBs to the original 5-bit value to produce the new expanded 8-bit value.

e.g., 5-bit R5 = 11000'b expanded to 8 bit R8 = 11000110'b

When reducing the number of bits in a component (e.g., from RGB888 to RGB565), the LSBs of the old value are truncated to arrive at the shortened value.

e.g., 8-bit R8 = 11000000'b reduced to 5 bit R5 = 11000'b

Replication followed by truncation ensures that the truncated value equals the original pre-replicated operand, i.e., RGB565 -> RGB888 expansion followed immediately by RGB888 -> RGB565 reduction results in the same original value.

Luminance Conversion

Conversion from luminance-only color formats (e.g., L8/Y8/YUV100, A8L8, etc.) to and from RGB or YUV formats should be done as outlined below.

Luminance-only to YUV

L -> Y, 128 -> U, 128 -> V

Luminance-only to RGB

L -> R, L -> G, L -> B

YUV to Luminance-only

Y -> L
Clamp L between 0.0 and 1.0

RGB to Luminance-only

(R*5 + G*9 + B*2)/16 -> L
Clamp L between 0.0 and 1.0

Adding and Removing Alpha

If converting from a pixel format with alpha to a pixel format with no alpha, the alpha value is dropped. If converting from a pixel format with no alpha to a pixel format with alpha, the alpha value in the destination pixel format should be set to 1.0.

7.3.1.3.4 Chroma Upsampling

VIC implements a fixed function YUV chroma up-sampling that allows pixel formats with sub-sampled chroma to be up-sampled to YUV444 right at the beginning of the pipeline, allowing them to be converted to RGB, passed through de-gamma and then sent through the scaling/blending pipelines just like other RGB formats. This feature is required to be able to scale YUV surfaces in linear space.

The up-sampling block supports both 2-tap and 5-tap up-sampling modes. It supports both a two component/pixel (UV plane) mode, as well as a one component/pixel mode (U plane or V plane). Any other mode that does not require chroma up-sampling is passed through the pipeline untouched.

5-tap mode is recommended for high-quality video use-cases.

7.3.1.3.5 Color Correct Processing

VIC processes colors "correctly," meaning that it follows industry-standard guidelines when doing image processing like scaling, compositing, etc. In particular, all linear operations like scaling and blending are done in the linear pixel space after correcting for any non-linear gamma inputs.

Full Degamma/Regamma Support

VIC has full support for reading, writing, and processing gamma space color space pixels. This means that VIC puts pixels through a degamma logic shortly after they are read; pixel processing including scaling, gamut mapping, and blending then happen in the linear color space, and there is a regamma conversion to convert pixels back into the gamma space before they are written out.

Since VIC has to process different types of content, each of which might be encoded with a different gamma, the degamma logic allows software to program the parameters of the gamma equation into the VIC config structure as necessary. Similarly, the output regamma logic implemented in a programmable fashion so that software can choose the output gamma space as necessary.

7.3.1.3.6 Surface Transform Processing

Surface Transformation

The blend unit is responsible for generating the destination surface addressing that implements mirroring (across X-axis and/or Y-axis) of tiles within the Target rectangle of the output surface.

The output buffer implements the ability to mirror pixels in the X and Y directions within each output tile. This ability, in addition to the fetch control changes mentioned above, give VIC the ability to flip the contents of the entire Target rectangle about its axis.

The output buffer also implements the hardware necessary to support transposition of the Target Rectangle. This ability, when combined with the target rectangle flips described above, gives VIC the ability to rotate images by any multiple of 90 degrees.

Rotation must be done in the pipeline after de-interlacing because the de-interlacer assumes a normal orientation of the input fields (interlaced field lines run horizontal). Rotation of the image prior to de-interlacing makes the de-interlacer ineffective.

Figure 7.20 Interlaced Field Lines Run Horizontally

The programmed starting address of the source and destination rectangles remain at the upper left corner of the unit. The Source Image diagrammed below shows the logical tile traversal pattern as the input rectangle traverses the VIC unit (tiles are read in raster order from left to right, then top to bottom). The destination rectangle diagrammed below shows the expected transformations when the programmer specifies the combination of flipX, flipY, and transposeXY transformations available to be applied to the output rectangle.

Figure 7.21 Surface Transformation Orientations

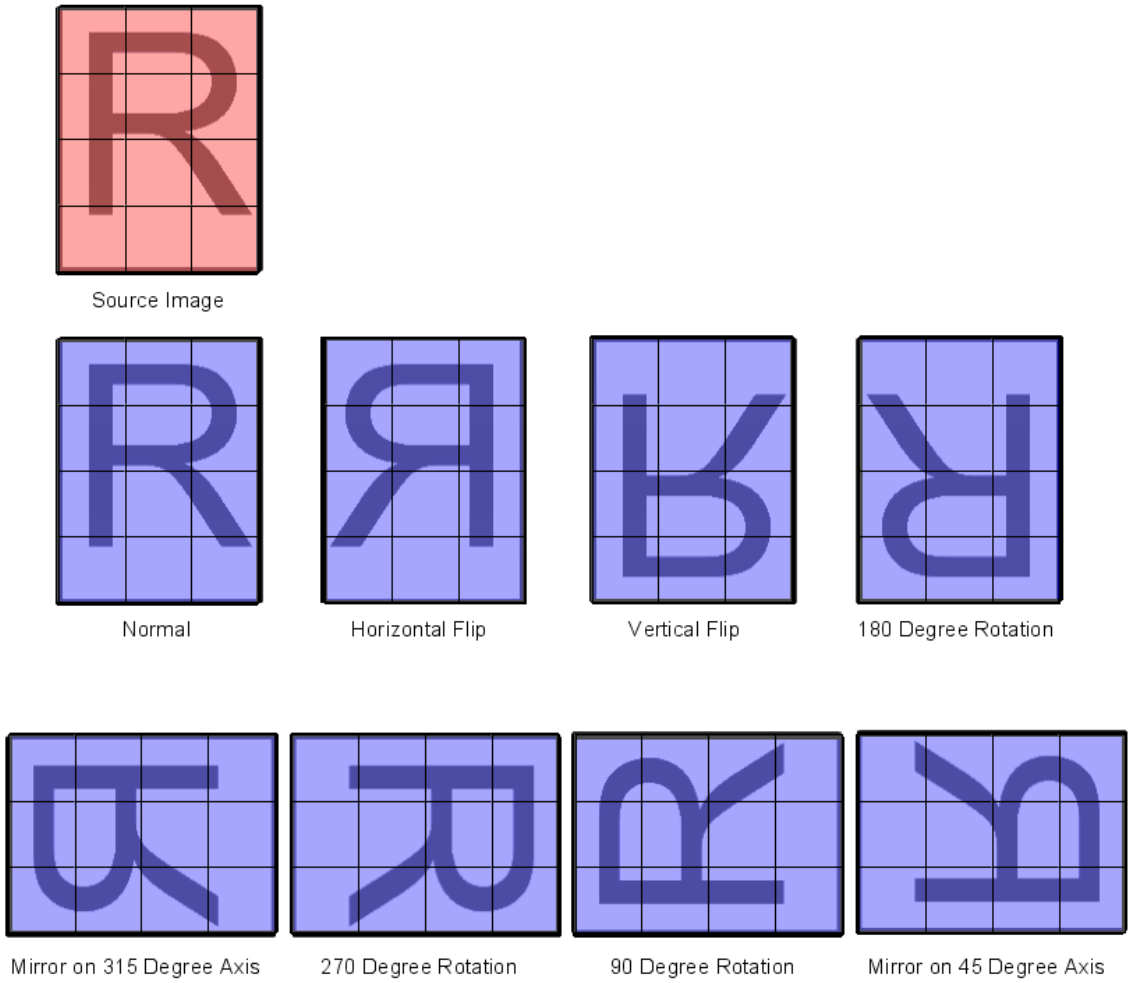


Table 7.38 Output Buffer Surface Transformation Controls

Transformation	transposeXY	flipY	flipX
Normal (no transformation)	0	0	0
H flip (mirror on vertical axis)	0	0	1
V flip (mirror on horizontal axis)	0	1	0
180-degree rotation	0	1	1
Mirror on 315-degree axis	1	0	0
270-degree rotation	1	1	0
90-degree rotation	1	0	1
Mirror on 45-degree axis	1	1	1

Surface mirroring (FlipX/FlipY) is accomplished via programming of the OutputConfig structure. Surface transposition is accomplished via programming of the OutputTranspose field of the OutputConfig structure.

The results of the transformations are summarized in the rules below.

1. The output surface parameters (SurfaceWidth/Height, LumaWidth/Height, and ChromaWidth/Height) are transposed when output transpose is enabled. This means that if the target rectangle/destination rectangles fit within the output surface before transformation, they also fit within the output surface after all transformations are done.
2. The position of the target rectangle in the output surface gets flipped or transposed along with all of its contents (clear rectangles and destination rectangles) when OutputFlipX/ OutputFlipY/OutputTranspose are enabled.
3. When output flips are enabled, the entire contents of the target rectangle are flipped, including all ClearRects and DestinationRects and their contents. The axis of the flip is the axis of the output surface.
4. When OutputTranspose is enabled, the position of the target rectangle is transposed, as does its contents (including all clear/destination rectangles).
5. When transpose and flips are enabled together, the overall effect of the transformation happens as if the flip operations occur first, followed by the transpose operation.

Scaling and Filtering

The VIC contains Y and X-scaler units that together allow 5 or 10 tap polyphase filtering and scaling of the input surface. The hardware is programmed with a set of filter coefficients for each filter-type (Detail/Noise/Normal), phase (0,1/32,2/32, .. 32/32), scale ratio (1:1, 2:1, 4:1, 8:1, 16:1), filter length (5-tap, 10-tap) and stream (sub-picture or normal). Based on the actual scale-ratio, phase etc. the hardware can then interpolate between these programmed values for use in the actual filters.

There are four filter types (5 tap non-substream, 5 tap substream, and 10 tap non-substream, 10 tap substream). Therefore there are five tables for each coefficient type, as well as different tables for different scale down ratio of 1:1, 2:1, 4:1, 8:1, and 16:1. This means there are a total of 20 tables and that the maximum downscale ratio is 16:1.

Each table has 16x4 memory entries (for 5-tap filter) or 16x9 memory entries (for 10-tap filter). Each memory entry has three 10-bit base coefficients (for Detail/Noise/Default) that are feed into a LERP3 simultaneously to interpolate based on different noise and detail weight. The table has 16 columns, and each column stores all the coefficients for phase 0/32, 1/32, 2/32, 3/32, 4/32, 5/32, 6/32, 7/32, 8/32, 9/32, 10/32, 11/32, 12/32, 13/32, 14/32, and 15/32, with the exception that the first column stores coefficients for both 0/32 and 16/32.

Each column stores four sets of base coefficients for 5-tap filter because the last coefficient can be derived from the remaining four. Phase 0/32 and 16/32 are special cases in that only two sets of base coefficients are needed for each of them in 5-tap case and only four sets of base coefficients are needed for phase 0/32 in 10-tap case and five sets of base coefficients are needed for phase

16/32 in 10-tap case. Therefore, phase 0/32 and 16/32 are folded into the same column. Due to symmetry, we don't need to store base coefficients for phases 17 through 31.

VIC also implements 1-tap (nearest-neighbor) and 2-tap (bi-linear) filtering modes. These modes can be used when power and bandwidth are of greater concern than quality. The 1- and 2-tap modes use filter coefficients that are based only on the output pixel phase with respect to the input pixel grid, and do not use the programmable coefficient tables at all. As a result, it doesn't make sense to enable Detail Filtering or Noise Filtering when using these modes.

Subpixel Source Rectangles

VIC has the ability to specify source rectangles to a sub-pixel resolution. This feature is used in cases requiring a smooth zoom into a portion of an image in real-time, for example when using the zoom button on a camera.

7.3.1.3.7 Video Quality Improve

De-interlacing

The VIC supports various modes of de-interlacing the input video content such as DiSi1, BOB, and WEAVE. Weave is only used on progressive content or in case inverse telecine detected a cadence.

7.3.1.3.8 Temporal Noise Reduction

TNR1

VIC has a temporal noise reduction (TNR) algorithm based on a motion adaptive IIR filter, where the filter weight was adapted based on a few constants and the SOS difference between a 3x3 neighborhood of pixels around the current pixel in the current and previous frames.

```
SOS = sum_of_squared_differences between current and prev frame in 3x3 neighborhood.
If (!denoise || (sos >> (20-deblur_const))!=0) {
    alpha = 0;
} else {
    alpha = min(1024, iir - ((iir * sos) >> (20-deblur_const)));
}
mul = hdr? max(0, min(1024, ((c * hdrm)>>10) + hdrb)) : 1024;
alpha = (alpha * mul) >> 10;
diffc  = fc  - c;
dc = diffc * diffc;
ad  = max(4, abs(dc) * ac);
max_alpha = la ? 1024 - 4096/ad : 1024;
```

```
alpha = min(alpha, max_alpha);
out = ((alpha * diffc)>>10) + c;
```

The pixel differences calculated in the above algorithm can be written out as a motion-adaptation buffer to memory for use in the motion-adaptive de-interlacing algorithm (DiSi1).

7.3.1.1.3.9 Blender

Blender Configuration

The blender API allows for symmetric blend modes between the VIC and Display units.

The VIC blender supports the following different blending modes:

- DXVAHD_ALPHA_FILL_MODE_OPAQUE
- DXVAHD_ALPHA_FILL_MODE_BACKGROUND
- DXVAHD_ALPHA_FILL_MODE_DESTINATION
- DXVAHD_ALPHA_FILL_MODE_COMPOSITED
- DXVAHD_ALPHA_FILL_MODE_SOURCE_ALPHA

All modes other than _ALPHA_FILL_MODE_COMPOSITED are as defined in the Microsoft DirectX Video Acceleration 2.0 Enhanced Video Processor specification (DXVA). The mode that allows the VIC and Display blenders to match each other is enabled by setting the AlphaFillMode in the OutputConfig structure to _ALPHA_FILL_MODE_COMPOSITED. When the AlphaFillMode is set to anything other than _ALPHA_FILL_MODE_COMPOSITED, the srcFact and dstFact parameters specified below are ignored and do not affect the processing.

The parameterizable blender interface allows a variety of blend modes, including the following:

- Per-Pixel Non-Premultiplied Alpha Blend output = src*src_alpha + dst*(1-src_alpha)
- Per Pixel Source Premultiplied Alpha Blend output = src + dst*(1-src_alpha)
- Constant-Alpha Blend output = src*const_alpha + dst*(1-const_alpha)
- Per-Pixel Non-Premultiplied Alpha Blend with constant blend output = src*src_alpha*const_alpha + dst*(1-src_alpha*const_alpha)
- Per-Pixel Premultiplied Alpha Blend with constant blend output = src*const_alpha + dst*(1-src_alpha*const_alpha)

Note that color-key is not supported and the following text should be read keeping in mind that ColorKeySelect should always be set to disabled.

The blend data path requires the color key comparison match the result, and per-slot inputs from the config structure to formulate the blend equation. srcFactC, srcFactA, dstFactC, and dstFactA multiplicand inputs are determined by the config structure programming and color key comparison results.

Table 7.39 Per-slot Blend Configuration

Blend Spec Name	Register Name	Description
K1	AlphaK1	8-bit constant alpha value
K2	AlphaK2	8-bit constant alpha value
srcFactC_Match_Select	srcFactCMatchSelect	3-bit enum which sets srcFactC: ▪ K1: AlphaK1 ▪ K1_TIMES_DST: AlphaK1*dstA ▪ NEG_K1_TIMES_DST: 1.0-(AlphaK1*dstA) ▪ K1_TIMES_SRC: AlphaK1*srcA ▪ ZERO: 0
dstFactC_Match_Select	dstFactCMatchSelect	3-bit enum which sets dstFactC: ▪ K1: AlphaK1 ▪ K2: AlphaK2 ▪ K1_TIMES_DST: AlphaK1*dstA ▪ NEG_K1_TIMES_DST: 1.0-(AlphaK1*dstA) ▪ NEG_K1_TIMES_SRC: 1.0-(AlphaK1*srcA) ▪ ZERO: 0 ▪ ONE: 1.0
srcFactA_Match_Select	srcFactAMatchSelect	3-bit enum which sets srcFactA: ▪ K1: AlphaK1 ▪ K2: AlphaK2 ▪ ZERO: 0 ▪ NEG_K1_TIMES_DST: 1.0-(AlphaK1*dstA)
dstFactA_Match_Select	dstFactAMatchSelect	3-bit enum which sets dstFactA: ▪ K2: AlphaK2 ▪ NEG_K1_TIMES_SRC: 1.0-(AlphaK1*srcA) ▪ ZERO: 0 ▪ ONE: 1.0
UseK3	UseOverrideR	1-bit boolean to override srcR with constant values K3R
UseK3	UseOverrideG	1-bit boolean to override srcG with constant values K3G
UseK3	UseOverrideB	1-bit boolean to override srcB with constant values K3B
UseK3	UseOverrideA	1-bit boolean to override srcA with constant values K3A
K3R	OverrideR	10-bit source override R value
K3G	OverrideG	10-bit source override G value

Blend Spec Name	Register Name	Description
K3B	OverrideB	10-bit source override B value
K3A	OverrideA	10-bit source override A value
MaskR	MaskR	1-bit boolean to override outputR to equal dstR
MaskG	MaskG	1-bit boolean to override outputG to equal dstG
MaskB	MaskB	1-bit boolean to override outputB to equal dstB
MaskA	MaskA	1-bit boolean to override outputA to equal dstA

Given the source and destination factor settings based on config structure inputs and color match results, the per-pixel blend equations are as follows, where srcR, srcG, srcB, srcA, dstR, dstG, dstB, and dstA represent the incoming R, G, B, and A components of the incoming source and already present destination pixels in the SFMem buffer (blended results of background and slot[0] through to slot[n-1] for input slot[n]).

outputR = (srcFactC * srcR) + (dstFactC * dstR)
outputG = (srcFactC * srcG) + (dstFactC * dstG)
outputB = (srcFactC * srcB) + (dstFactC * dstB)
outputA = (srcFactA * srcA) + (dstFactA * dstA)

srcFactC, srcFactA, dstFactC, and dstFactA are programmed by srcFactCMatchSelect, srcFactCMatchSelect, srcFactCMatchSelect, and srcFactCMatchSelect respectively.

UseK3 allows the incoming source components to be overridden by constant color components K3R, K3G, K3B, and K3A.

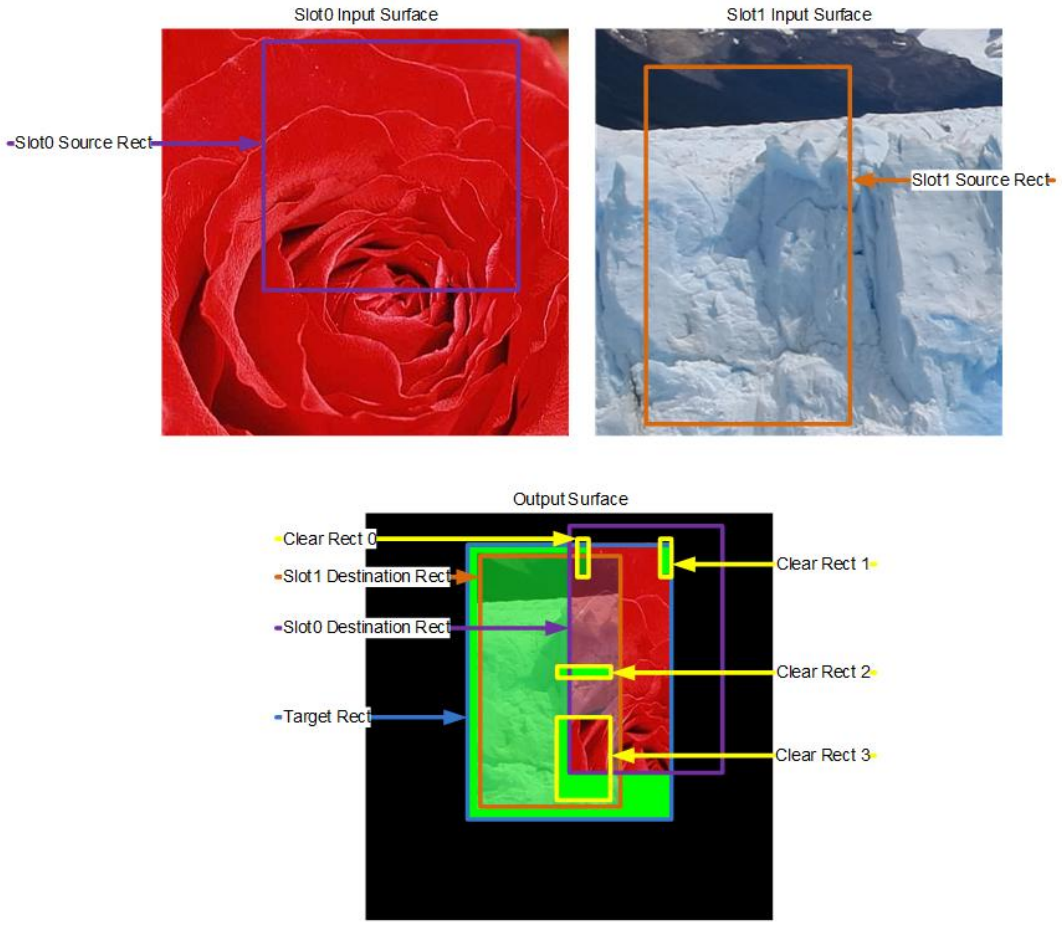
MaskR, MaskG, MaskB, and MaskA boolean settings provide additional per-component mask bits to allow pass-through of the destination pixel components to the output pixel.

A programmable background color can still be set to fill the target rectangle background via the OutputConfig structure. The programmed background color should follow these rules:

- The background color is programmed in the same color space as the blending color space. For example, if the blending is happening in an RGB color space, then the background color is treated as an ARGB value. If the blending happens in a YUV color space, the background color is treated as an AYUV value.
- Similarly, if the blending happens in a linear color space, the background color is treated as a linear color space value. If the blending is in non-linear space, then the background color is also expected to be non-linear.

7.3.1.3.10 Surface Composition

Figure 7.22 Programming Input and Output Surface Parameters



The maximum dimension of any image on the input or output sides of the VIC is 16384 pixels. The dimensions of each input slot surface can be defined along with the pixel and memory format types in the VIC Config Structure.

A source rectangle defines the region of pixels of the input slot surface that contributes to the composition of the output surface, and a destination rectangle defines the region of the output surface that is affected by a given input slot. Together, the source and destination rectangle parameters specify the scaling ratios desired. Each slot can also dictate how pixel data is laid out in each surface cache entry by specifying the amount of pixel data, i.e., the number of bytes wide, logically represented by each cache entry; this provides flexibility in reducing the memory over fetch associated with fetch of surfaces under various use cases (e.g., scaling ratios, de-interlacing modes, etc.).

Via the OutputConfig structure, the dimensions of the output surface can be defined along with the pixel and memory format types. The same structure also defines a target rectangle to restrict the pixel processing output to a certain rectangle in the output surface. A programmable color can also be set to fill the background of the target rectangle.

The VIC allows the programming of clear rectangles that prevent the fetch of pixels from specific rectangular regions of input slots. The use of the clear rectangles can be used to reduce redundant pixel fetches, for example, if an opaque surface is known to occlude a layer below it, a clear rectangle can be specified to prevent fetching of pixels from the occluded region of the lower layer. Up to eight clear rectangles are specified via the ClearRectStruct structs, and per-slot clear rectangle mask enables are specified via the SlotConfig struct. Clear rectangles are specified relative to the output surface coordinate system.

With the addition of the sub-pixel source rectangle feature, source rectangle coordinates are specified in the config structure in a U14.16 format, allowing us to specify the source rectangle at a sub-pixel resolution. Destination, target, and clear rectangles coordinates are pixel aligned.

7.3.1.4 Geometry Transform Processing

7.3.1.4.1 SubFeature List

- Four pixel/clock throughput
- 16-bit input pixel format
- Full warp map capability
- Uniform spaced sparse warp map
- Programmable nine control points spline patch
- Non-fixed patch size
- 4x4 bicubic pixel interpolation filter
- Out-of-the-bound check via MaskBitMap
- On-the-fly inverse perspective transform only support
- Sub-frame level processing
- Xsobel and 4x4 downsample

Table 7.40 Supported Pixel Format

Enumerant	Value
T_R8	4
T_Y8___V8U8_N444	62
T_Y8___V8U8_N422	64
T_Y8___V8U8_N420	68
T_R16	112

Enumerant	Value
T_Y16___V16U16_N444	116
T_Y16___V16U16_N422	117
T_Y16___V16U16_N420	118
T_Y10___V10U10_N420 (YUV420 semi-planar)	82
T_Y12___V12U12_N420 (YUV420 semi-planar)	99

Note: Output format is always same as input format.

Table 7.41 Surface Memory Format

Surface Name	Value
Pixel image surface	Pitch Linear / Block Linear (16x2) Programmable GOB height
Sparse Warp Map Surface MAP_XY	Pitch Linear, X/Y Interleaved, S15.5 precision, and 8 bytes per control points
MskBitMap Surface	Tile64x16 Linear
Pixel interpolation coefficient surface	Linear
XSobel Gradient surface	Block Linear (16x2) / Pitch Linear
Xsobel 4x4 downsample surface	Block Linear (16x2) / Pitch Linear
Xsobel Neighbor surface	Internal work buffer

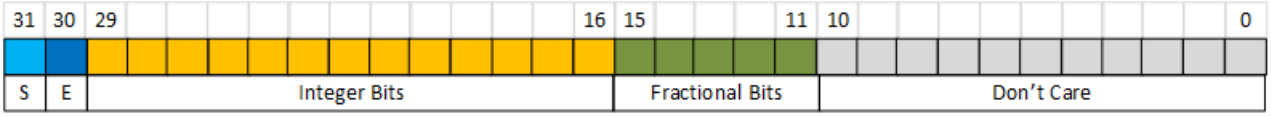
MskBitMap Surface stores in Tile64x16 the linear format.

Linear array: bool MskBitMap[y*stride + x]
Tile64x16 linear array: bool MskBitMap[((y/16) * (stride/64) + (x/64)) * (64x16) + (y%16) * 64 + x%64]

For one tile 64x16, it costs the address-continuous 128Bytes to store the mask bit information.

Control point format in memory:

Figure 7.23 Control Point Format



7.3.1.4.2 Temporal Noise Reduction

Processing video to reduce noise can significantly improve the quality. A low complexity Motion-adaptive Temporal filtering noise reduction mechanism which also adapts to video content and ambient lighting conditions is described here.

Table 7.42 TNR Feature List

Feature	Change from Parker
4 Pixel/clock throughput	New
Bilateral Spatial Filter	New
Pixel-wise SAD calculation	Changed
Improved SAD-alpha curve	Changed
Temporal alpha restrict	New
Non-temporal beta blending	Changed
Flexible alpha clip for luma and chroma	Changed
Flexible alpha-beta curve	Changed
Register-level programming	Changed
Sub-frame level processing	New
Merged LDC_TNR in one pass	New
TNR bypass mode	New

Table 7.43 Supported Pixel Formats

Enumerant	Value
T_Y8_U8__Y8_V8 (YUV422 packed) (YUY2/YUYV)	57
T_U8_Y8__V8_Y8 (YUV422 packed) (UYVY)	59
T_Y8___V8U8_N420 (YUV420 semi-planar)	68
T_Y8__*U8*__V8_N420 (YUV420 planar, YV12)	72
T_Y10___V10U10_N420 (YUV420 semi-planar)	82
T_Y10__*U10*__V10_N420 (YUV420 planar)	86
T_Y12___V12U12_N420 (YUV420 semi-planar)	99
T_Y12__*U12*__V12_N420 (YUV420 planar)	103

Notes:

1. Output format is always same as input format
2. T_Y8__*U8*__V8_N420 is equal to YV12

Table 7.44 Surface Memory Format

Surface Name	Value
Pixel image surface	Pitch Linear / Block Linear (16x2) Programmable GOB height
TNR alpha, top and left data	Internal work buffer

7.3.1.4.3 16-bit New Pixel Format Scale

Scaling support in the CV pipeline requires 16-bit data format support throughout the VIC scaling pipeline. VIC is used to crop the ROI from the original wide camera to get the DLA network input resolution. For example:

960x540@540fps-Y16 -> VIC -> 224x224@540-Y16 -> DLA (downscale)

192x108@1080fps-Y16 -> VIC -> 224x224@1080-Y16 -> DLA (upscale)

SubFeature List

- 16-bit input pixel format
- Internal data-path precision promotion

7.3.1.5 Power Management

In the SoC, VIC is spread over two layout partitions namely VICA and VICB, but all of it belongs to a single PG domain VIC.

7.3.1.5.1 Clock Gating Requirements

Hierarchical Clock Gating

VIC supports clock gating. First-level clock gating (FLCG) refers to clock gating of the top-level VIC_CLK clock input into the VIC unit which is controlled by software. First-level clock gating can be enabled by software to reduce power when VIC is inactive.

Second-level clock gating (SLCG) refers to clock gating of individual portions of the VIC pipeline, controlled by the VIC hardware, with software overrides. When each frame completes, no further work is pending in the method stream and all portions of the VIC pipeline are idle, then each major portion of the VIC pipeline is clock gated automatically by the hardware.

A block-level clock gating (BLCG) controller monitors vic_idle (idling of all logic in the vic_clk domain). When vic_clk logic is idle, the vic_clk clk_en signal to CAR is deasserted, automatically disabling the vic_clk at the root and thus saving power in the clock tree. A free running vic_clk_nobg clock is required for the BLCG controller to work. The BLCG controller is configurable (e.g., enable, idle_delay count, wakeup delay count, etc.) via CSB registers in the VIC MISC space.

Activity Monitor Activity Signal

The VIC does not include any direct power monitoring features. It exports an activity monitoring signal that is utilized by the DVFS software to program the VIC clock frequencies and voltages.

A Per-Unit Activity Monitor (ACTMON) monitors VIC activity. The ACTMON receives an activity signal from the VIC engine, and using that information, generates an IIR average that is compared against watermark thresholds, and interrupts software whenever certain activity thresholds are exceeded.

The ACTMON is monitored by software and aids in power management policies, such as setting VIC's clock frequency, enabling and disabling first level clock gating, and power gating of the engine.

The VIC activity sent to the ACTMON is asserted from the time that VIC first fetches a method for processing from THI, until the moment that pixel data processing completes and end of frame is asserted. Any activity within the THI either before the start of method fetch by VIC or after end of frame does not add to the accumulated VIC activity.

7.3.1.5.2 Power Gating

The VIC is located in its own partition(s) which are power-gateable. VIC power-gating is enabled in LP0/LP1 states, as well as after extended periods of inactivity. When the VIC is split across multiple partitions, it is expected that all partitions are power-gated (or not) together; there is no use-case where some partitions are power-gated and other partitions are not.

Software is responsible for making sure VIC is idle, saving/restoring context, reloading the Falcon firmware, and rebooting it whenever power-gating the unit.

7.3.1.5.3 Power Gating Sequence

VIC Clocks and Resets

The following are the CAR register fields associated with VIC clocks:

- CLK_RST_CONTROLLER_CLK_OUT_ENB_VIC_0.CLK_ENB_VIC

The following are the CAR register fields associated with VIC resets:

- CLK_RST_CONTROLLER_RST_DEV_VIC_0.SWR_VIC_RST

VIC Power Gating

1. Enable VIC clock(s)
2. Flush MC client VIC
   a. Set the MC register bit:
      MC_CLIENT_HOTRESET_CTRL_0.VIC_FLUSH_ENABLE
   b. Poll MC_CLIENT_HOTRESET_STATUS_0 until the following bits are set:
      VIC_HOTRESET_STATUS
3. Apply clamps
   a. Write to PMC_IMPL_PART_VIC_CLAMP_CONTROL_0 with CLAMP=ON
   b. Wait 100 ns for clamps to engage
4. Assert VIC resets
5. Disable VIC clocks
6. Set VIC power state to PG
   a. Write PMC_IMPL_PART_VIC_POWER_GATE_CONTROL_0
   b. Poll PMC_IMPL_PART_VIC_POWER_GATE_CONTROL_0 until bit START==DONE
   c. Read PMC_IMPL_PART_VIC_POWER_GATE_STATUS_0 and verify that desired state is reached. If there is a mismatch, flag ERROR

VIC Power Ungating

1. Set VIC power state to ON
   a. Write PMC_IMPL_PART_VIC_POWER_GATE_CONTROL_0
   b. Poll PMC_IMPL_PART_VIC_POWER_GATE_CONTROL_0 until bit START==DONE
   c. Read PMC_IMPL_PART_VIC_POWER_GATE_STATUS_0 and verify that the desired state is reached. If there is a mismatch, flag ERROR
2. Enable VIC clock(s)
3. Assert VIC reset(s)
   a. Write CAR register(s) specified above
   b. Add 2 µs delay to allow flushing of non-resettable flops
4. Release clamps
   a. Write to PMC_IMPL_PART_VIC_CLAMP_CONTROL_0 with CLAMP=OFF
   b. Wait 100 ns for clamps to disengage
5. Deassert VIC reset(s)
6. Enable MC client VIC by clearing the following bits:
   a. MC_CLIENT_HOTRESET_CTRL_0.VIC_FLUSH_ENABLE
7. Optionally (if requested by the unit's driver), disables VIC clock(s).

7.3.1.5.4 Instantiate Error Collator and Hook Up Faults to HSM

To standardize the reporting and testability of fault reports to the HSM, the VIC adopts the standard Error Collator plug-ins. The VIC instantiates these plugins to collect the various safety related errors and provide two fault lines (correctable and uncorrectable) to the HSM.

The errors reported to the HSM are the Falcon IMEM-ECC (correctable, uncorrectable), Falcon DMEM-Parity, Falcon firmware (correctable, uncorrectable) including timeout, and errors discovered during periodic testing.

Table 7.45 List of VIC Error Types

Error name	Produced by	Output error signal
Falcon IMEM ECC SEC	ECC plugin terminal	vic_fault_corrected
Falcon IMEM ECC DED	ECC plugin terminal	vic_fault_critical
Falcon DMEM parity	Parity plugin terminal	vic_fault_critical
Falcon Software correctable Error	Software corrected errors	vic_fault_corrected
Falcon Software critical Error	Software critical errors	vic_fault_critical

Embedded safety within Falcon provides two ERBs (Embedded RuBys) for firmware to send fault reports. These ERBs connect to the error collator, while Falcon reports both corrected errors and critical errors detected by firmware based protection.

The errors include:

- Corruption of critical configuration registers
- Internal IP hardware timeout
- Other firmware based safety errors

7.3.1.6 Virtualization

Virtualization is implemented by the Host Interface (THI); the Host Interface serves as an interface between the SoC Host Controller and Falcon microcontroller(s).

- Two StreamIDs should be available for VIC engine, one (StreamID1) for internal Falcon use, and the other (StreamID0) for VIC processing pipeline use.
- Both StreamIDs should be programmable by Host Controller RM privilege (TrustZone is optional), but NOT by other security privileges.
- StreamID0 should be programmable by either Channel programming or MMIO, StreamID1 should be only programmable by MMIO.
- Channel Lock (CH_LOCK) should be supported, when enabled (CH_LOCK=1), push buffer commands should NOT be allowed to access non-THI registers.

7.3.2 VIC Programming Guidelines

The compositor implements the DirectX Video Acceleration 2.0 Enhanced Video Processor (DXVA) specification. This does de-interlacing, scaling, color conversion, proc-amp, and compositing for up to eight input surfaces. It also supports features like gamma correction and non-linear color enhancement.

7.3.2.1 Class Overview

The software programming interface for the VIC is similar to that of the other video engines in the SoC like NVENC and NVDEC. This compatibility is maintained by virtue of the shared front-ends of both these engines on the Host1x interface, consisting of a THI module and an embedded Falcon microcontroller.

7.3.2.1.1 Method Interface

VIC has its own class definition, which includes a couple of registers named METHOD_OFFSET and METHOD_DATA, which are interpreted by the THI and are used to create method writes (corresponding to methods in the VIC class specification) into the VIC. These methods are used to set up pointers to the various surfaces that the VIC needs to read or write, as well as pointers to needed config structures. The VIC class also defines trigger methods such as Execute(), used to start the actual processing of the data for the next output image.

7.3.2.1.2 Sync-Points

Interactions between hardware and software are controlled and synchronized through the sync-point mechanism. For example software can request to be notified when the VIC engine is done processing a certain frame's worth of data by enqueuing a sync-point method after the Execute method that is sent down for that frame, and by programming Host1x to raise an interrupt to the CPU when the sync-point is updated by the VIC hardware.

The THI block is responsible for keeping track of outstanding sync-points and sending appropriate sync-point ACKs back to Host1x. It does this by handling the interrupts raised by the VIC after completing an Execute_Notify() method, and translating that to a sync-point counter update for Host1x.

7.3.2.1.3 Context-Switch

Context switching uses a method-based context switching interface. This has two methods: context save and context restore. When using method-based context switching, VIC supports only WFI (wait-for-idle) context-switching.

The same context switch mechanism can be used to save and restore contexts around power-gating events. When software wishes to power gate the VIC, a context save method should be sent down to save the current context before power gating. On resuming from ELPG, software has to issue a soft-reset to the VIC engine, and then go through the entire re-initialization sequence for VIC, including boot-strapping of the Falcon microcontroller. After the power-up of the engine is complete, software should send a restore method to VIC, which causes the saved context to be restored.

Note that VIC is primarily a state-less engine. When processing each frame, methods are issued to the VIC to point to the various surfaces used during each frame's processing, and programming the VIC engine involves writing a configuration structure that specifies all functionality required within the current frame. The engine reads in this configuration structure at the start of every frame of processing. The only state that may require context switch saving would be any pending methods required for upcoming frame executes that have not yet processed by the Falcon microcontroller. There are no security-specific programmable configuration bits in the engine, and thus no concern with handling context switching of security configuration bits.

7.3.2.1.4 Interrupts

The THI block creates sync-point acknowledges that it sends out to Host1x. Thus, the only events that cause an interrupt to be generated to the CPU should be fatal events that need software intervention (often a full engine soft-reset) in order to fix.

7.3.2.1.5 Driver Programming Model

The compositor operation requires the following buffers to be provided. These buffers are explained in more detail in the following sections. When the Execute() method is sent, the Falcon program is initiated which in turn triggers the various VIC operations in the correct sequence.

Buffers	I/O	Producer	Description/Restrictions
Config Buffer	I	Driver	Configuration parameters
Palette Buffer	I	Driver	Palette table
History Buffer	I/O	App	Histogram and Cadence parameters
Input surfaces	I	Driver/VIC	Input Surfaces
Noise reduced surfaces	I/O	VIC	Noise reduced version of input
Motion map surfaces	I/O	VIC	Motion map of input
Output surface	O	VIC	Final output surface (post-composition)
FCE microcode buffer	I	Driver	FCE microcode
CrcStruct Offset	O	VIC	CRC struct

7.3.2.1.6 Surface Padding and Alignment

The base address of all buffers (input and output) programmed into the VIC have to be aligned to a multiple of the block size for block-linear surfaces, and a multiple of 256B for pitch linear surfaces. Thus, the buffer base addresses are always aligned to at least 256B; the base address pointers programmed into the VIC reflect this and only accept address bits [39:8].

The parameters SurfaceWidth and SurfaceHeight used in the ConfigStruct refer to the actual image dimensions. LumaWidth and LumaHeight refer to the padded buffer dimensions, and apply to both luma and RGBA surfaces. ChromaWidth and ChromaHeight refer to the padded buffer dimensions for the chroma planes of planar or semi-planar YUV surfaces. All the Surface/Luma/Chroma widths andThis is for BOB_FIELD only heights are programmed in terms of pixels, not bytes. The constraints below describe the restrictions on the buffer widths and heights.

The VIC does not assume that the widths of block-linear buffers are automatically padded to the next multiple of 64B. The VIC Luma/Chroma Width fields are still used to describe the padded surface pitch for the block-linear buffer, and can be any multiple of the block width.

When using VIC configurable surface cache entry modes of 16Bx16, 32Bx8, or 64Bx4, VIC can read in any block linear surface. Note however, the VIC cache entry mode of 128Bx2 only works where the buffer widths are multiples of 128B.

The constraint for any image buffer size is for it to be a multiple of the surface memory format block size as well as the configurable surface cache entry size. The memory format block size is considered to be 256x1 for pitch linear, and 64x(8<<BLK_HEIGTH) for block linear buffers. BLK_HEIGHT is the log2 encoded block height as usual.

For input surfaces:

- If pitch linear, then the configurable surface cache entry size can be set to 256Bx1, 128Bx2, or 64Bx4. The buffer width should be padded to a multiple of 256B, and the height should be padded to a multiple of the configurable surface cache entry height.
- If block linear, then the configurable surface cache entry size can be set to 128Bx2, 64Bx4, 32Bx8, or 16Bx16. The buffer width should be padded to the next multiple of the max of 64B or the configurable surface cache entry width, and the height should be padded to the maximum of the configurable surface cache entry height and the block height in lines (8 << BLK_HEIGHT, where BLK_HEIGHT is the number of GOBs tall per block linear block, log2 encoded).

For output surfaces:

- If pitch linear, the buffer width should be padded to a multiple of 256B.
- If block linear, the buffer width should be padded to the next multiple of 64B, and the height should be padded to a multiple of the block height in lines (8 << BLK_HEIGHT, where BLK_HEIGHT is the number of GOBs tall per block linear block, log2 encoded).

As noted previously, the VIC has to process 16Bx16 blocks of memory in the pre-process phase. This would be very inefficient to do if the surfaces are set up as pitch-linear buffers, and therefore the usage of pitch-linear buffers for images that need pre-processing is strongly discouraged. For block linear buffers, the Falcon micro-code forces the cache-line dimensions for the pre-process phase alone to 16Bx16 to increase efficiency. As a result of this, software should make sure to make the image buffer height a multiple of 16 when enabling pre-processing.

While any pixel data outside of the boundaries defined by SurfaceWidth/SurfaceHeight of an input surface is not used to contribute to the output image, it may still be read into the VIC surface cache. If the image buffer is padded as described above, the hardware does not issue any read or write accesses outside of the image buffer range.

7.3.2.2 Application Enumerations

The following enumerators are used in the configuration structures passed by the driver:

- PIXEL_FORMAT
- DXVAHD_FRAME_FORMAT
- DXVAHD_DEINTERLACE_MODE_PRIVATE
- DXVAHD_ALPHA_FILL_MODE
- BLK_KIND
- BLEND_SRCFACTC
- BLEND_DSTFACTC
- BLEND_SRCFACTA
- BLEND_DSTFACTA
- GAMMA_MODE
- FILTER_LENGTH
- FILTER_TYPE

Table 7.46 PIXEL_FORMAT

Enumerant	Value
T_A8	0
T_L8	1
T_A4L4	2
T_L4A4	3
T_R8	4
T_A8L8	5
T_L8A8	6

Enumerant	Value
T_R8G8	7
T_G8R8	8
T_B5G6R5	9
T_R5G6B5	10
T_B6G5R5	11
T_R5G5B6	12
T_A1B5G5R5	13
T_A1R5G5B5	14
T_B5G5R5A1	15
T_R5G5B5A1	16
T_A5B5G5R1	17
T_A5R1G5B5	18
T_B5G5R1A5	19
T_R1G5B5A5	20
T_X1B5G5R5	21
T_X1R5G5B5	22
T_B5G5R5X1	23
T_R5G5B5X1	24
T_A4B4G4R4	25
T_A4R4G4B4	26
T_B4G4R4A4	27
T_R4G4B4A4	28
T_B8_G8_R8	29
T_R8_G8_B8	30
T_A8B8G8R8	31
T_A8R8G8B8	32
T_B8G8R8A8	33
T_R8G8B8A8	34
T_X8B8G8R8	35

Enumerant	Value
T_X8R8G8B8	36
T_B8G8R8X8	37
T_R8G8B8X8	38
T_A2B10G10R10	39
T_A2R10G10B10	40
T_B10G10R10A2	41
T_R10G10B10A2	42
T_A4P4	43
T_P4A4	44
T_P8A8	45
T_A8P8	46
T_P8	47
T_P1	48
T_U8V8	49
T_V8U8	50
T_A8Y8U8V8	51
T_V8U8Y8A8	52
T_Y8_U8_V8	53
T_Y8_V8_U8	54
T_U8_V8_Y8	55
T_V8_U8_Y8	56
T_Y8_U8__Y8_V8	57
T_Y8_V8__Y8_U8	58
T_U8_Y8__V8_Y8	59
T_V8_Y8__U8_Y8	60
T_Y8___U8V8_N444	61
T_Y8___V8U8_N444	62
T_Y8___U8V8_N422	63
T_Y8___V8U8_N422	64

Enumerant	Value
T_Y8___U8V8_N422R	65
T_Y8___V8U8_N422R	66
T_Y8___U8V8_N420	67
T_Y8___V8U8_N420	68
T_Y8__*U8*__V8_N444	69
T_Y8__*U8*__V8_N422	70
T_Y8__*U8*__V8_N422R	71
T_Y8__*U8*__V8_N420	72
T_U8	73
T_V8	74
T_Y10___U10V10_N444	75
T_Y10___V10U10_N444	76
T_Y10___U10V10_N422	77
T_Y10___V10U10_N422	78
T_Y10___U10V10_N422R	79
T_Y10___V10U10_N422R	80
T_Y10___U10V10_N420	81
T_Y10___V10U10_N420	82
T_Y10__*U10*__V10_N444	83
T_Y10__*U10*__V10_N422	84
T_Y10__*U10*__V10_N422R	85
T_Y10__*U10*__V10_N420	86
T_U10	87
T_V10	88
T_L10	89
T_U10V10	90
T_V10U10	91
T_Y12___U12V12_N444	92
T_Y12___V12U12_N444	93

Enumerant	Value
T_Y12___U12V12_N422	94
T_Y12___V12U12_N422	95
T_Y12___U12V12_N422R	96
T_Y12___V12U12_N422R	97
T_Y12___U12V12_N420	98
T_Y12___V12U12_N420	99
T_Y12__*U12*__V12_N444	100
T_Y12__*U12*__V12_N422	101
T_Y12__*U12*__V12_N422R	102
T_Y12__*U12*__V12_N420	103
T_U12	104
T_V12	105
T_L12	106
T_U12V12	107
T_V12U12	108
T_L16	109
T_A16B16G16R16	110
T_A16Y16U16V16	111

Table 7.47 DXVAHD_FRAME_FORMAT

Enumerant	Value	Description
DXVAHD_FRAME_FORMAT_PROGRESSIVE	0	A_PROGRESSIVE frame format refers to a non-interlaced surface. The only de-interlacing mode that is valid for this frame format is _WEAVE.
DXVAHD_FRAME_FORMAT_INTERLACED_TOP_FIELD_FIRST	1	An interlaced surface, but the top and bottom fields (from different time instants) are weaved together into a single input surface with the top field on the first and every alternating line. The valid de-interlacing modes for this frame format are _WEAVE and _BOB.
DXVAHD_FRAME_FORMAT_INTERLACED_BOTTOM_FIELD_FIRST	2	An interlaced surface, but the top and bottom fields (from different time instants) are weaved together into a single input surface with the bottom field on the first line. The valid de-interlacing modes for this frame format are _WEAVE and _BOB.

Enumerant	Value	Description
DXVAHD_FRAME_FORMAT_TOP_FIELD	3	A surface with frame format _TOP_FIELD is a field surface and is hence half the height of the original progressive source. The lines in the current surface should be displayed above the lines of the previous field (which would be of type _BOTTOM_FIELD). The valid de-interlacing modes for this frame format are _WEAVE, _BOB_FIELD, _NEWBOB, _DISI1 and _WEAVE_LUMA_BOB_FIELD_CHROMA.
DXVAHD_FRAME_FORMAT_BOTTOM_FIELD	4	A surface with frame format _BOTTOM_FIELD is a field surface and is hence half the height of the original progressive source. The lines in the current surface should be displayed below the lines of the previous field (which would be of type _TOP_FIELD). The valid de-interlacing modes for this frame format are _WEAVE, _BOB_FIELD, _NEWBOB, _DISI1 and _WEAVE_LUMA_BOB_FIELD_CHROMA.
DXVAHD_FRAME_FORMAT_SUBPIC_PROGRESSIVE	5	The _SUBPIC frame formats are similar to the non-subpic frame formats, but are meant for the sub-picture layers from the Blu-ray spec. Choosing a sub-picture vs. a non-sub-picture frame format affects the filter kernels that are used to scale the slot.
DXVAHD_FRAME_FORMAT_SUBPIC_INTERLACED_TOP_FIELD_FIRST	6	
DXVAHD_FRAME_FORMAT_SUBPIC_INTERLACED_BOTTOM_FIELD_FIRST	7	
DXVAHD_FRAME_FORMAT_SUBPIC_TOP_FIELD	8	
DXVAHD_FRAME_FORMAT_SUBPIC_BOTTOM_FIELD	9	
DXVAHD_FRAME_FORMAT_TOP_FIELD_CHROMA_BOTTOM	10	A field surface with the luma corresponding to the _TOP_FIELD, but with chroma corresponding to the _BOTTOM_FIELD. The valid de-interlacing modes for this frame format are _WEAVE, _BOB_FIELD, _NEWBOB, _DISI1 and _WEAVE_LUMA_BOB_FIELD_CHROMA.
DXVAHD_FRAME_FORMAT_BOTTOM_FIELD_CHROMA_TOP	11	A field surface with the luma corresponding to the _BOTTOM_FIELD, but with chroma corresponding to the _TOP_FIELD. The valid de-interlacing modes for this frame format are _WEAVE, _BOB_FIELD, _NEWBOB, _DISI1 and _WEAVE_LUMA_BOB_FIELD_CHROMA.
DXVAHD_FRAME_FORMAT_SUBPIC_TOP_FIELD_CHROMA_BOTTOM	12	
DXVAHD_FRAME_FORMAT_SUBPIC_BOTTOM_FIELD_CHROMA_TOP	13	

Table 7.48 DXVAHD_DEINTERLACE_MODE_PRIVATE

Enumerant	Value	Description
DXVAHD_DEINTERLACE_MODE_PRIVATE_WEAVE	0	Weave the top and bottom fields together. Requires previous and current field surfaces to be enabled.

Enumerant	Value	Description
DXVAHD_DEINTERLACE_MODE_PRIVATE_BOB_FIELD	1	Double the current field in height. Only requires current field to be enabled.
DXVAHD_DEINTERLACE_MODE_PRIVATE_BOB	2	Double the current field in height. Only works with the _INTERLACED frame formats, and picks the current field lines out of the input interleaved surface. Only requires current field to be enabled.
DXVAHD_DEINTERLACE_MODE_PRIVATE_NEWBOB	3	Similar to the BOB_FIELD mode, but uses a motion adaptive algorithm to decide whether to BOB or WEAVE fields together. Requires at least the following fields enabled: previous, next, current, previous motion, motion, combined motion.
DXVAHD_DEINTERLACE_MODE_PRIVATE_DISI1	4	Uses a motion adaptive algorithm to decide whether to BOB or WEAVE fields together. In addition, uses an edge-directed spatial interpolation algorithm in order to avoid any weave artifacts. Requires at least the following fields enabled: previous, next, current, previous motion, motion, combined motion.
DXVAHD_DEINTERLACE_MODE_PRIVATE_WEAVE_LUMA_BOB_FIELD_CHROMA	5	Uses the WEAVE algorithm on the luma, and the BOB_FIELD algorithm on the chroma components.
DXVAHD_DEINTERLACE_MODE_PRIVATE_MAX	15	This is not a valid mode for de-interlacing and should not be programmed into the hardware.

Table 7.49 DXVAHD_ALPHA_FILL_MODE

Enumerant	Value	Description
DXVAHD_ALPHA_FILL_MODE_OPAQUE	0	Opaque (all alpha inside target rect are set to 1.0).
DXVAHD_ALPHA_FILL_MODE_BACKGROUND	1	Background (all alpha inside target rect are set to background alpha).
DXVAHD_ALPHA_FILL_MODE_DESTINATION	2	Destination (alpha remains unchanged).
DXVAHD_ALPHA_FILL_MODE_SOURCE_STREAM	3	Source stream (alpha from source stream without planar alpha).
DXVAHD_ALPHA_FILL_MODE_COMPOSITED	4	Composited (composited alpha, starting with background). In this mode, the blend parameters specified by SrcFactA/DstFactA/SrcFactC/DstFactC etc. are used.
DXVAHD_ALPHA_FILL_MODE_SOURCE_ALPHA	5	Source alpha (alpha from source stream with planar alpha).

Table 7.50 BLK_KIND

Enumerant	Value	Description
BLK_KIND_PITCH	0	This enum is used to program the various "BlkKind" fields in the ConfigSructure.
BLK_KIND_GENERIC_16Bx2	1	Block Linear (16Bx2) memory format

Enumerant	Value	Description
BLK_KIND_BL_NAIVE	2	This format is not supported in the VIC hardware and should not be programmed in the config structure.
BLK_KIND_BL_KEPLER_XBAR_RAW	3	This format is not supported in the VIC hardware and should not be programmed in the config structure.
BLK_KIND_VP2_TILED	15	This format is not supported in the VIC hardware and should not be programmed in the config structure.

Table 7.51 BLEND_SRCFACTC

Enumerant	Value	Description
BLEND_SRCFACTC_K1	0	SrcFactC = AlphaK1
BLEND_SRCFACTC_K1_TIMES_DST	1	SrcFactC = AlphaK1 * dstA
BLEND_SRCFACTC_NEG_K1_TIMES_DST	2	SrcFactC = 1.0 - (AlphaK1 * dstA)
BLEND_SRCFACTC_K1_TIMES_SRC	3	SrcFactC = AlphaK1 * srcA
BLEND_SRCFACTC_ZERO	4	SrcFactC = 0

Table 7.52 BLEND_DSTFACTC

Enumerant	Value	Description
BLEND_DSTFACTC_K1	0	DstFactC = AlphaK1
BLEND_DSTFACTC_K2	1	DstFactC = AlphaK2
BLEND_DSTFACTC_K1_TIMES_DST	2	DstFactC = AlphaK1 * dstA
BLEND_DSTFACTC_NEG_K1_TIMES_DST	3	DstFactC = 1.0 - (AlphaK1 * dstA)
BLEND_DSTFACTC_NEG_K1_TIMES_SRC	4	DstFactC = 1.0 - (AlphaK1 * srcA)
BLEND_DSTFACTC_ZERO	5	DstFactC = 0.0
BLEND_DSTFACTC_ONE	6	DstFactC = 1.0

Table 7.53 BLEND_SRCFACTA

Enumerant	Value	Description
BLEND_SRCFACTA_K1	0	SrcFactA = AlphaK1
BLEND_SRCFACTA_K2	1	SrcFactA = AlphaK2
BLEND_SRCFACTA_NEG_K1_TIMES_DST	2	SrcFactA = 1.0 - (AlphaK1 * dstA)
BLEND_SRCFACTA_ZERO	3	SrcFactA = 0.0
BLEND_SRCFACTA_MAX	7	Invalid mode - should not be programmed.

Table 7.54 BLEND_DSTFACTA

Enumerant	Value	Description
BLEND_DSTFACTA_K2	0	DstFactA = AlphaK2
BLEND_DSTFACTA_NEG_K1_TIMES_SRC	1	DstFactA = 1.0 - (AlphaK1 * srcA)
BLEND_DSTFACTA_ZERO	2	DstFactA = 0.0
BLEND_DSTFACTA_ONE	3	DstFactA = 1.0
BLEND_DSTFACTA_MAX	7	Invalid mode - should not be programmed.

Table 7.55 GAMMA_MODE

Enumerant	Value	Description
GAMMA_MODE_NONE	0	Do not apply any degamma or regamma to the color components.
GAMMA_MODE_SRGB	1	Apply the sRGB degamma/regamma curve to the color components.
GAMMA_MODE_REC709	2	Apply the Rec.709 degamma/regamma curve to the color components.
GAMMA_MODE_RESERVED	3	Apply the Rec.2020 degamma/regamma curve to the color components.

Table 7.56 FILTER_LENGTH

Enumerant	Value	Description
FILTER_LENGTH_1TAP	0	Nearest-neighbor filtering
FILTER_LENGTH_2TAP	1	Bi-linear filtering
FILTER_LENGTH_5TAP	2	5-tap filters with filter kernel specified using the FilterStruct
FILTER_LENGTH_10TAP	3	10-tap filters with filter kernel specified using the FilterStruct

Table 7.57 FILTER_TYPE

Enumerant	Value	Description
FILTER_TYPE_NORMAL	0	Filter kernel that is used for regular scaling.
FILTER_TYPE_NOISE	1	Filter kernel that is used for spatial smoothing. The weight of this filter is specified using the FilterNoise/ChromaNoise parameters in the SlotConfig structure.
FILTER_TYPE_DETAIL	2	Filter kernel that is used for spatial sharpening. The weight of this filter is specified using the FilterDetail/ChromaDetail parameters in the SlotConfig structure.

7.3.2.3 Application Memory Structures

7.3.2.3.1 Config Structure

The Config structure is read from the frame buffer by the surface cache unit and is stored in small memories inside each subunit. The entire ConfigStruct is the concatenation of the Surface Cache, Surface List, Color Conversion, Blending and Fetch Control structs (in this order). The structs are read in 128-bit units from memory and sent to the according subunits memory with the correct memory address by the Surface Cache.

The config structure is broken up into smaller structs that are used depending on the active slots and their content. Each structure is a multiple of 128 bits in size, and the start of the struct needs to be aligned to a 256 byte boundary.

Note: NV12 and NV24 share the same pixel format but can be distinguished by the fact that NV24 is a field based format (i.e., every surface contains a single field) whereas NV12 is frame based (i.e., every surface contains an entire frame).

Note: For highest quality, filter override mode as defined in FetchControlCoeffStruct should always be used.

Table 7.58 ConfigStruct Substructures

Name	Data Type	Offset	Notes
pipeConfig	PipeConfig (128 bits)	0	
outputConfig	OutputConfig (128 bits)	128	
outputSurfaceConfig	OutputSurfaceConfig (128 bits)	256	
outColorMatrixStruct	MatrixStruct (256 bits)	384	
clearRectStruct[4]	ClearRectStruct (128 bits)	640 + (i * 128)	Up to eight clear rectangles supported. Each ClearRectStruct specifies two rectangles
slotStruct[8]	SlotStruct (1408 bits)	SlotStruct (1408 bits)	1152 + (i * 1408)

Table 7.59 PipeConfig Details

Name	Data Type	Offset	Notes
DownsampleHoriz	fixed<0,11,0> (11 bits)	0	TargetWidth/DownsampleWidth (U9.2 used to load bias filter) Set to 0 to enable filter override mode (mapping between stream and filter explained below)
reserved0	fixed<0,5,0> (5 bits)	11	

Name	Data Type	Offset	Notes
DownsampleVert	fixed<0,11,0> (11 bits)	16	TargetHeight/DownsampleHeight (U9.2 used to load bias filter) Set to 0 to enable filter override mode (mapping between stream and filter explained below)
reserved1	fixed<0,5,0> (5 bits)	27	
reserved2	fixed<0,32,0> (32 bits)	32	
reserved3	fixed<0,32,0> (32 bits)	64	
reserved4	fixed<0,32,0> (32 bits)	96	

Table 7.60 outputConfig Details

This structure specifies the output flip enables and target rectangle.

Name	Data Type	Offset	Notes
AlphaFillMode	fixed<0,3,0> (3 bits)	0	Alpha fill mode (DXVAHD_ALPHA_FILL_MODE)
AlphaFillSlot	fixed<0,3,0> (3 bits)	3	SlotId for when AlphaFillMode == Source stream/Source alpha
BackgroundAlpha	fixed<0,10,0> (10 bits)	6	Background color A
BackgroundR	fixed<0,10,0> (10 bits)	16	Background color R
BackgroundG	fixed<0,10,0> (10 bits)	26	Background color G
BackgroundB	fixed<0,10,0> (10 bits)	36	Background color B
RegammaMode	fixed<0,2,0> (2 bits)	46	The regamma curve to be used for the blended output
OutputFlipX	fixed<0,1,0> (1 bit)	48	Horizontal flip enable
OutputFlipY	fixed<0,1,0> (1 bit)	49	Vertical flip enable
OutputTranspose	fixed<0,1,0> (1 bit)	50	Transpose enable
reserved1	fixed<0,1,0> (1 bit)	51	
reserved2	fixed<0,12,0> (12 bits)	52	
TargetRectLeft	fixed<0,14,0> (14 bits)	64	Target rectangle. Restricts the output to a certain rectangle inside the output surface. Pixels outside of this area are guaranteed to remain unmodified.
reserved3	fixed<0,2,0> (2 bits)	78	
TargetRectRight	fixed<0,14,0> (14 bits)	80	
reserved4	fixed<0,2,0> (2 bits)	94	
TargetRectTop	fixed<0,14,0> (14 bits)	96	

Name	Data Type	Offset	Notes
reserved5	fixed<0,2,0> (2 bits)	110	
TargetRectBottom	fixed<0,14,0> (14 bits)	112	
reserved6	fixed<0,2,0> (2 bits)	126	

Table 7.61 outputSurfaceConfig

Name	Data Type	Offset	Notes
OutPixelFormat	fixed<0,7,0> (7 bits)	0	Pixel format of output surface (PIXEL_FORMAT)
OutChromaLocHoriz	fixed<0,2,0> (2 bits)	7	Chroma location of output surface (See SurfaceListSurfaceStruct)
OutChromaLocVert	fixed<0,2,0> (2 bits)	9	
OutBlkKind	fixed<0,4,0> (4 bits)	11	The block linear kind of the output surface (BLK_KIND)
OutBlkHeight	fixed<0,4,0> (4 bits)	15	Block-linear height (in gobs, log2)
reserved0	fixed<0,3,0> (3 bits)	19	
reserved1	fixed<0,10,0> (10 bits)	22	
OutSurfaceWidth	fixed<0,14,0> (14 bits)	32	Output surface width minus 1
OutSurfaceHeight	fixed<0,14,0> (14 bits)	46	Output surface height minus 1
reserved2	fixed<0,4,0> (4 bits)	60	
OutLumaWidth	fixed<0,14,0> (14 bits)	64	Padded output surface luma width minus 1
OutLumaHeight	fixed<0,14,0> (14 bits)	78	Padded output surface luma height minus 1
reserved3	fixed<0,4,0> (4 bits)	92	
OutChromaWidth	fixed<0,14,0> (14 bits)	96	Padded output surface chroma width minus 1
OutChromaHeight	fixed<0,14,0> (14 bits)	110	Padded output surface chroma height minus 1
reserved4	fixed<0,4,0> (4 bits)	124	

Table 7.62 outColorMatrixStruct Details

The matrices defined in these structures are used to specify the color space conversion between input and output pixel formats if any.

Name	Data Type	Offset	Notes
matrix_coeff00	fixed<0,20,0> (20 bits)	0	Matrix entry (0,0) of 4x3 color conversion matrix. Precision and right shift are the same as for the luma vector.
matrix_coeff10	fixed<0,20,0> (20 bits)	20	Matrix entry (1,0) of 4x3 color conversion matrix
matrix_coeff20	fixed<0,20,0> (20 bits)	40	Matrix entry (2,0) of 4x3 color conversion matrix
matrix_r_shift	fixed<0,4,0> (4 bits)	60	Right shift value for matrix
matrix_coeff01	fixed<0,20,0> (20 bits)	64	Matrix entry (0,1) of 4x3 color conversion matrix
matrix_coeff11	fixed<0,20,0> (20 bits)	84	Matrix entry (1,1) of 4x3 color conversion matrix
matrix_coeff21	fixed<0,20,0> (20 bits)	104	Matrix entry (2,1) of 4x3 color conversion matrix
reserved0	fixed<0,3,0> (3 bits)	124	
matrix_enable	fixed<0,1,0> (1 bit)	127	
matrix_coeff02	fixed<0,20,0> (20 bits)	128	Matrix entry (0,2) of 4x3 color conversion matrix
matrix_coeff12	fixed<0,20,0> (20 bits)	148	Matrix entry (1,2) of 4x3 color conversion matrix
matrix_coeff22	fixed<0,20,0> (20 bits)	168	Matrix entry (2,2) of 4x3 color conversion matrix
reserved1	fixed<0,4,0> (4 bits)	188	
matrix_coeff03	fixed<0,20,0> (20 bits)	192	Matrix entry (0,3) of 4x3 color conversion matrix
matrix_coeff13	fixed<0,20,0> (20 bits)	212	Matrix entry (1,3) of 4x3 color conversion matrix
matrix_coeff23	fixed<0,20,0> (20 bits)	232	Matrix entry (2,3) of 4x3 color conversion matrix
reserved2	fixed<0,4,0> (4 bits)	252	

Table 7.63 ClearRectStruct[4] Details

The ClearRectStruct structures together define eight clear rectangles. These rectangles are enabled or disabled for each slot using the fields in SurfaceList0Struct.

Name	Data Type	Offset	Notes
ClearRect0Left	fixed<0,14,0> (14 bits)	0	First clear rectangle of 128-bit chunk
reserved0	fixed<0,2,0> (2 bits)	14	

Name	Data Type	Offset	Notes
ClearRect0Right	fixed<0,14,0> (14 bits)	16	
reserved1	fixed<0,2,0> (2 bits)	30	
ClearRect0Top	fixed<0,14,0> (14 bits)	32	
reserved2	fixed<0,2,0> (2 bits)	46	
ClearRect0Bottom	fixed<0,14,0> (14 bits)	48	
reserved3	fixed<0,2,0> (2 bits)	62	
ClearRect1Left	fixed<0,14,0> (14 bits)	64	Second clear rectangle of 128-bit chunk
reserved4	fixed<0,2,0> (2 bits)	78	
ClearRect1Right	fixed<0,14,0> (14 bits)	80	
reserved5	fixed<0,2,0> (2 bits)	94	
ClearRect1Top	fixed<0,14,0> (14 bits)	96	
reserved6	fixed<0,2,0> (2 bits)	110	
ClearRect1Bottom	fixed<0,14,0> (14 bits)	112	
reserved7	fixed<0,2,0> (2 bits)	126	

Table 7.64 slotStruct[8] Details

Name	Data Type	Offset
slotConfig	SlotConfig (512 bits)	0
slotSurfaceConfig	SlotSurfaceConfig (128 bits)	512
lumaKeyStruct	LumaKeyStruct (128 bits)	640
colorMatrixStruct	MatrixStruct (256 bits)	768
gamutMatrixStruct	MatrixStruct (256 bits)	1024
blendingSlotStruct	BlendingSlotStruct (128 bits)	1280

SlotConfig

This structure contains enable bits for each slot, which need to be set as described below.

- Noise reduction requires a forward and backward reference field for interlaced content and a backward reference for progressive content (see Methods section about how to set surfaces). Both require a noise reduced surface (field for interlaced, frame for progressive).

- MotionMap calculation is required for DiSi1/DiNewBob and also needs a forward and backward reference. Behavior is not defined for progressive streams so enabling it needs to be flagged as an error.
- Cadence Detection enables artifact and weave counts. It also enables Falcon code to force deinterlace mode to weave if a cadence was detected.

Field Detail:

Name	Data Type	Offset	Description
SlotEnable	fixed<0,1,0> (1 bit)	0	Enable or disable bit for this slot
DeNoise	fixed<0,1,0> (1 bit)	1	Enable bit for noise reduction
AdvancedDenoise	fixed<0,1,0> (1 bit)	2	Enable the advanced denoising algorithm (TNR2) for each slot.
CadenceDetect	fixed<0,1,0> (1 bit)	3	Cadence detection enable bit
MotionMap	fixed<0,1,0> (1 bit)	4	Motion map enable bit for each slot. Required for DiSi1/DiNewBob
MMapCombine	fixed<0,1,0> (1 bit)	5	Enable bit for combine motion map for each slot. This is required for DiSi1/DiNewBob. This requires MotionMap being enabled and also required a prevMotionMap surface (see Methods). Behavior without MotionMap enabled is undefined and has to be flagged as an error.
IsEven	fixed<0,1,0> (1 bit)	6	Select if current field is even (used for cadence detection)
ChromaEven	fixed<0,1,0> (1 bit)	7	Enable if chroma of current field is even
CurrentFieldEnable	fixed<0,1,0> (1 bit)	8	Indicates which surfaces are enabled for fetching Bit 0: Current field Bit 1: Previous field Bit 2: Next field Bit 3: Next field (noise filtered, takes priority over unfiltered field) Bit 4: Current motion field Bit 5: Previous motion field Bit 6: Previous motion field Bit 7: Combined motion field
PrevFieldEnable	fixed<0,1,0> (1 bit)	9	
NextFieldEnable	fixed<0,1,0> (1 bit)	10	
NextNrFieldEnable	fixed<0,1,0> (1 bit)	11	
CurMotionFieldEnable	fixed<0,1,0> (1 bit)	12	

Name	Data Type	Offset	Description
PrevMotionFieldEnable	fixed<0,1,0> (1 bit)	13	
PpMotionFieldEnable	fixed<0,1,0> (1 bit)	14	
CombMotionFieldEnable	fixed<0,1,0> (1 bit)	15	
FrameFormat	fixed<0,4,0> (4 bits)	16	Frame format of the frame (DXVAHD_FRAME_FORMAT)
FilterLengthY	fixed<0,2,0> (2 bits)	20	Filter length for each stream: 0: 1-tap (nearest neighbor) 1: 2-tap (bi-linear) 2: 5-tap 3: 10-tap
FilterLengthX	fixed<0,2,0> (2 bits)	22	
Panoramic	fixed<0,12,0> (12 bits)	24	Panoramic scaling parameter.
reserved1	fixed<0,22,0> (22 bits)	36	
DetailFltClamp	fixed<0,6,0> (6 bits)	58	The maximum range allowed for the difference between the filter output and center pixel of the input filter support. Set to 0 to disable clamping.
FilterNoise	fixed<0,10,0> (10 bits)	64	Strength of the spatial noise filter for slot 0. All detail and noise filter values (including chroma) become meaningless and should be set to 0 as soon as filter override is enabled
FilterDetail	fixed<0,10,0> (10 bits)	74	Strength of the detail filter for slot 0
ChromaNoise	fixed<0,10,0> (10 bits)	84	Strength of the noise filter for slot 0
ChromaDetail	fixed<0,10,0> (10 bits)	94	Strength of the detail filter for slot 0
DeinterlaceMode	fixed<0,4,0> (4 bits)	104	De-interlace mode (DXVAHD_DEINTERLACE_MODE_PRIVATE)
MotionAccumWeight	fixed<0,3,0> (3 bits)	108	Accumulation weight for motion IIR filter for slot 0 (default should be 6). The first time an even/odd motion field is calculated AccumWeight should be set to 0 to avoid having to clear the motion buffer beforehand.
Noiselir	fixed<0,11,0> (11 bits)	111	Accumulation weight for noise reduction IIR filter for slot 0 (default value should be 0x300).
LightLevel	fixed<0,4,0> (4 bits)	122	Describes the level of lighting present when the input image was captured. This parameter is used along with the AdvancedDenoise bits to determine the exact denoising algorithm to be applied for noise reduction.
reserved4	fixed<0,2,0> (2 bits)	126	

Name	Data Type	Offset	Description
SoftClampLow	fixed<0,10,0> (10 bits)	128	Lower soft clamping bound
SoftClampHigh	fixed<0,10,0> (10 bits)	138	Upper soft clamping bound
reserved5	fixed<0,3,0> (3 bits)	148	
reserved6	fixed<0,9,0> (9 bits)	151	
PlanarAlpha	fixed<0,10,0> (10 bits)	160	10-bit Planar alpha value. This planar alpha is multiplied with the alpha value coming from the stream. In case of palletized alpha formats (like AI44/A8P8/AI88) the stream alpha value is the alpha value from the surface multiplied with the alpha value from the palette.
ConstantAlpha	fixed<0,1,0> (1 bit)	170	If true, planar alpha value is used instead of stream alpha, if false, stream alpha is multiplied with planar alpha. Constant alpha has to be set for all surfaces not containing any alpha data (like XRGB/NV12/YUY2/UYVY/YV12).
StereoInterleave	fixed<0,3,0> (3 bits)	171	Enable pixel interleave for auto-stereoscopic panels (STEREO_INTERLEAVE)
ClipEnabled	fixed<0,1,0> (1 bit)	174	Enable clip against negative values after gamut matrix
ClearRectMask	fixed<0,8,0> (8 bits)	175	Clear rectangle mask
DegammaMode	fixed<0,2,0> (2 bits)	183	The de-gamma curve to be used for the slot (GAMMA_MODE)
reserved7	fixed<0,1,0> (1 bit)	185	
DecompressEnable	fixed<0,1,0> (1 bit)	186	Decompression enable bit for the slot. If set to false, then all requests bypass the CDE and go directly to MCCIF. If this bit is set, then all requests to the surface are treated as potentially compressed, and the compress bits are looked up for every ROP tile in the surface
reserved9	fixed<0,5,0> (5 bits)	187	
DecompressCtbCount	fixed<0,8,0> (8 bits)	192	The number of CompTagBuffer (CTB) entries that are assigned to this slot. Software has to allocate the total available comptag entries to each of the slots, based on the scaling ratios etc.
DecompressZbcColor	fixed<0,32,0> (32 bits)	200	The Zero-Bandwidth-Clear color to be used when a ROP tile is ZBC compressed.
reserved12	fixed<0,24,0> (24 bits)	232	
SourceRectLeft	fixed<0,30,0> (30 bits)	256	The source rectangle defines the region of pixels that are read from the source surface. Any pixel data outside of this is not used inside VIC but might still be read. The source rectangle coordinates are specified in a U14.4 format, allowing us to specify fractional coordinates. The source rectangle should lie entirely within the input surface for the slot, that is, negative values, or values greater than the size of the surface are illegal.

Name	Data Type	Offset	Description
reserved14	fixed<0,2,0> (2 bits)	286	
SourceRectRight	fixed<0,30,0> (30 bits)	288	
reserved15	fixed<0,2,0> (2 bits)	318	
SourceRectTop	fixed<0,30,0> (30 bits)	320	
reserved16	fixed<0,2,0> (2 bits)	350	
SourceRectBottom	fixed<0,30,0> (30 bits)	352	
reserved17	fixed<0,2,0> (2 bits)	382	
DestRectLeft	fixed<0,14,0> (14 bits)	384	The destination rectangle defines the region of the output surface that is affected by this slot. Together with the source rectangle it also defines the scaling rations. Any pixel outside of this region are not affected by this input stream. For any non-4:4:4 format all corners need to fall on a multiple of 2 (Right and Bottom are minus 1 encoded though).
reserved18	fixed<0,2,0> (2 bits)	398	
DestRectRight	fixed<0,14,0> (14 bits)	400	
reserved19	fixed<0,2,0> (2 bits)	414	
DestRectTop	fixed<0,14,0> (14 bits)	416	
reserved20	fixed<0,2,0> (2 bits)	430	
DestRectBottom	fixed<0,14,0> (14 bits)	432	
reserved21	fixed<0,2,0> (2 bits)	446	
reserved22	fixed<0,32,0> (32 bits)	448	
reserved23	fixed<0,32,0> (32 bits)	480	

SlotSurfaceConfig

Surface parameters for each slot

Field Detail:

Name	Data Type	Offset	Description
SlotPixelFormat	fixed<0,7,0> (7 bits)	0	Pixel format for each stream (PIXEL_FORMAT) Only NV12, YUY2, and UYVY allow for motion buffer calculation and DiSi1/ DiNewBob de-interlacing and noise reduction.
SlotChromaLocHoriz	fixed<0,2,0> (2 bits)	7	Horizontal chroma location (0: co-located with even luma; 1: in between even and odd luma; 2: co-located with odd luma; 3: between odd and next even luma
SlotChromaLocVert	fixed<0,2,0> (2 bits)	9	Vertical chroma location (0: co-located with even luma; 1: in between even and odd luma; 2: co-located with odd luma; 3: between odd and next even luma)
SlotBlkKind	fixed<0,4,0> (4 bits)	11	Block-linear kind (BLK_KIND)
SlotBlkHeight	fixed<0,4,0> (4 bits)	15	Block-linear height (in gobs, log2)
SlotCacheWidth	fixed<0,3,0> (3 bits)	19	Number of horizontal bytes per surface-cache cache-line. Each 256B cache line can store a source region of size as enumerated below. 0: 16Bx16 (BL16Bx2) 1: 32Bx8 (BL16Bx2) 2: 64Bx4 (BL16Bx2, PL) 3: 128Bx2 (BL16Bx2, PL) 4: 256Bx1 (PL)
reserved0	fixed<0,10,0> (10 bits)	22	
SlotSurfaceWidth	fixed<0,14,0> (14 bits)	32	Width of surface minus 1. Any pixel data outside of this is not used inside VIC but might still be read.
SlotSurfaceHeight	fixed<0,14,0> (14 bits)	46	Height of surface minus 1. Any pixel data outside of this is not used inside VIC but might still be read.
reserved1	fixed<0,4,0> (4 bits)	60	
SlotLumaWidth	fixed<0,14,0> (14 bits)	64	Padded luma width of surface minus 1. Any pixel data outside of this is not read.
SlotLumaHeight	fixed<0,14,0> (14 bits)	78	Padded luma height of surface minus 1. Any pixel data outside of this is not read.
reserved2	fixed<0,4,0> (4 bits)	92	
SlotChromaWidth	fixed<0,14,0> (14 bits)	96	Padded chroma width of surface minus 1. Any pixel data outside of this is not read. This value is not required for pixel interleaved surfaces such as ARGB
SlotChromaHeight	fixed<0,14,0> (14 bits)	110	Padded chroma height of surface minus 1. Any pixel data outside of this is not read. This value is not required for pixel interleaved surfaces such as ARGB
reserved3	fixed<0,4,0> (4 bits)	124	

LumaKeyStruct

Used to enable Luma keying and plane alpha parameters.

Field Detail:

Name	Data Type	Offset	Description
luma_coeff0	fixed<0,20,0> (20 bits)	0	Matrix entry (0) of 4x1 luma conversion matrix in S12.8 format
luma_coeff1	fixed<0,20,0> (20 bits)	20	Matrix entry (1) of 4x1 luma conversion matrix in S12.8 format
luma_coeff2	fixed<0,20,0> (20 bits)	40	Matrix entry (2) of 4x1 luma conversion matrix in S12.8 format
luma_r_shift	fixed<0,4,0> (4 bits)	60	The result of the matrix multiplication is right shifted by r_shift. To allow for highest accuracy, r_shift should always be as high as possible without losing any range in the other coefficients.
luma_coeff3	fixed<0,20,0> (20 bits)	64	Matrix entry (3) of 4x1 luma conversion matrix in S12.8 format. Is not affected by r_shift.
LumaKeyLower	fixed<0,10,0> (10 bits)	84	Lower luma key value
LumaKeyUpper	fixed<0,10,0> (10 bits)	94	Upper luma key value
LumaKeyEnabled	fixed<0,1,0> (1 bit)	104	Luma key enable
reserved0	fixed<0,2,0> (2 bits)	105	
reserved1	fixed<0,21,0> (21 bits)	107	

MatrixStruct

The matrices defined in these structures are used to specify the color space conversion between input and output pixel formats if any.

Field Detail:

Name	Data Type	Offset	Description
matrix_coeff00	fixed<0,20,0> (20 bits)	0	Matrix entry (0,0) of 4x3 color conversion matrix. Precision and right shift are the same as for the luma vector.
matrix_coeff10	fixed<0,20,0> (20 bits)	20	Matrix entry (1,0) of 4x3 color conversion matrix

Name	Data Type	Offset	Description
matrix_coeff20	fixed<0,20,0> (20 bits)	40	Matrix entry (2,0) of 4x3 color conversion matrix
matrix_r_shift	fixed<0,4,0> (4 bits)	60	Right shift value for matrix
matrix_coeff01	fixed<0,20,0> (20 bits)	64	Matrix entry (0,1) of 4x3 color conversion matrix
matrix_coeff11	fixed<0,20,0> (20 bits)	84	Matrix entry (1,1) of 4x3 color conversion matrix
matrix_coeff21	fixed<0,20,0> (20 bits)	104	Matrix entry (2,1) of 4x3 color conversion matrix
reserved0	fixed<0,3,0> (3 bits)	124	
matrix_enable	fixed<0,1,0> (1 bit)	127	
matrix_coeff02	fixed<0,20,0> (20 bits)	128	Matrix entry (0,2) of 4x3 color conversion matrix
matrix_coeff12	fixed<0,20,0> (20 bits)	148	Matrix entry (1,2) of 4x3 color conversion matrix
matrix_coeff22	fixed<0,20,0> (20 bits)	168	Matrix entry (2,2) of 4x3 color conversion matrix
reserved1	fixed<0,4,0> (4 bits)	188	
matrix_coeff03	fixed<0,20,0> (20 bits)	192	Matrix entry (0,3) of 4x3 color conversion matrix
matrix_coeff13	fixed<0,20,0> (20 bits)	212	Matrix entry (1,3) of 4x3 color conversion matrix
matrix_coeff23	fixed<0,20,0> (20 bits)	232	Matrix entry (2,3) of 4x3 color conversion matrix
reserved2	fixed<0,4,0> (4 bits)	252	

BlendingSlotStruct

Input parameters to blend equations in DXVAHD_ALPHA_FILL_MODE_COMPOSITED alpha fill mode

- outputR = (srcFactC * srcR) + (dstFactC * dstR)
- outputG = (srcFactC * srcG) + (dstFactC * dstG)
- outputB = (srcFactC * srcB) + (dstFactC * dstB)
- outputA = (srcFactA * srcA) + (dstFactA * dstA)

Field Detail:

Name	Data Type	Offset	Description
AlphaK1	fixed<0,10,0> (10 bits)	0	Constant Alpha value
reserved0	fixed<0,6,0> (6 bits)	10	Constant Alpha value
AlphaK2	fixed<0,10,0> (10 bits)	16	Constant Alpha value
reserved1	fixed<0,6,0> (6 bits)	26	
SrcFactCMatchSelect	fixed<0,3,0> (3 bits)	32	Blend Source Factor for Color if the color key comparison matches (BLEND_SRCFACTC)
reserved2	fixed<0,1,0> (1 bit)	35	
DstFactCMatchSelect	fixed<0,3,0> (3 bits)	36	Blend Destination Factor for Color if the color key comparison matches (BLEND_DSTFACTC)
reserved3	fixed<0,1,0> (1 bit)	39	
SrcFactAMatchSelect	fixed<0,3,0> (3 bits)	40	Blend Source Factor for Alpha if the color key comparison matches (BLEND_SRCFACTA)
reserved4	fixed<0,1,0> (1 bit)	43	
DstFactAMatchSelect	fixed<0,3,0> (3 bits)	44	Blend Source Factor for Alpha if the color key comparison matches (BLEND_DSTFACTA)
reserved5	fixed<0,1,0> (1 bit)	47	
reserved6	fixed<0,4,0> (4 bits)	48	
reserved7	fixed<0,4,0> (4 bits)	52	
reserved8	fixed<0,4,0> (4 bits)	56	
reserved9	fixed<0,4,0> (4 bits)	60	
reserved10	fixed<0,2,0> (2 bits)	64	
OverrideR	fixed<0,10,0> (10 bits)	66	Source RGBA override values
OverrideG	fixed<0,10,0> (10 bits)	76	
OverrideB	fixed<0,10,0> (10 bits)	86	

Name	Data Type	Offset	Description
OverrideA	fixed<0,10,0> (10 bits)	96	
reserved11	fixed<0,2,0> (2 bits)	106	
UseOverrideR	fixed<0,1,0> (1 bit)	108	Enable source RGBA value override
UseOverrideG	fixed<0,1,0> (1 bit)	109	
UseOverrideB	fixed<0,1,0> (1 bit)	110	
UseOverrideA	fixed<0,1,0> (1 bit)	111	
MaskR	fixed<0,1,0> (1 bit)	112	Per-component blend output override enables (replace blend output with destination RGBA values)
MaskG	fixed<0,1,0> (1 bit)	113	
MaskB	fixed<0,1,0> (1 bit)	114	
MaskA	fixed<0,1,0> (1 bit)	115	
reserved12	fixed<0,12,0> (12 bits)	116	

7.3.2.3.2 Surface Padding and Alignment

The base address of all buffers (input and output) programmed into the VIC have to be aligned to a multiple of the block size for block-linear surfaces, and a multiple of 256B for pitch linear surfaces. Thus, the buffer base addresses is always aligned to at least 256B; the base address pointers programmed into the VIC reflect this and only accept address bits [39:8].

The parameters SurfaceWidth and SurfaceHeight used in the ConfigStruct refer to the actual image dimensions. LumaWidth and LumaHeight refer to the padded buffer dimensions, and apply to both luma and RGBA surfaces. ChromaWidth and ChromaHeight refer to the padded buffer dimensions for the chroma planes of planar or semi-planar YUV surfaces. All the Surface/Luma/Chroma widths and heights are programmed in terms of pixels, not bytes. The constraints below describe the restrictions on the buffer widths and heights. The VIC does not assume that the widths of block-linear buffers are automatically padded to the next multiple of 64B. The VIC Luma/Chroma Width fields are still used to describe the padded surface pitch for the block-linear buffer, and can be any multiple of the block width.

When using VIC configurable surface cache entry modes of 16Bx16, 32Bx8 or 64Bx4, VIC can read in any block linear surface. Note, however, the VIC cache entry mode of 128Bx2 only works where the buffer widths are multiples of 128B. The constraint for any image buffer size is for it to be a multiple of the surface memory format block size as well as the configurable surface cache entry size. The memory format block size is considered to be 256x1 for pitch linear, and 64x(8<<BLK_HEIGTH) for block linear buffers. BLK_HEIGHT is the log2 encoded block height as usual.

For input surfaces:

- If pitch linear, then the configurable surface cache entry size can be set to 256Bx1, 128Bx2, or 64Bx4. The buffer width should be padded to a multiple of 256B, and the height should be padded to a multiple of the configurable surface cache entry height.
- If block linear, then the configurable surface cache entry size can be set to 128Bx2, 64Bx4, 32Bx8 or 16Bx16. The buffer width should be padded to the next multiple of the max of 64B or the configurable surface cache entry width, and the height should be padded to the max of the configurable surface cache entry height and the block height in lines (8 << BLK_HEIGHT, where BLK_HEIGHT is the number of GOBs tall per block linear block, log2 encoded).

For output surfaces:

- If pitch linear, the buffer width should be padded to a multiple of 256B.
- If block linear, the buffer width should be padded to the next multiple of 64B, and the height should be padded to a multiple of the block height in lines (8 << BLK_HEIGHT, where BLK_HEIGHT is the number of GOBs tall per block linear block, log2 encoded).

As noted previously, the VIC has to process 16Bx16 blocks of memory in the pre-process phase. This would be very inefficient to do if the surfaces are set up as pitch-linear buffers, and therefore the usage of pitch-linear buffers for images that need pre-processing is strongly discouraged. For block linear buffers, the Falcon micro-code forces the cache-line dimensions for the pre-process phase alone to 16Bx16 to increase efficiency. As a result of this, software should make sure to make the image buffer height a multiple of 16 when enabling pre-processing.

While any pixel data outside of the boundaries defined by SurfaceWidth/SurfaceHeight of an input surface is not used to contribute to the output image, it may still be read into the VIC surface cache. If the image buffer is padded as described above, the hardware does not issue any read or write accesses outside of the image buffer range.

7.3.2.3.3 Palette Structure

Palette Structure sends palette information for indexed formats. Each slot has its own space in the palette buffer even if the format is not indexed. For non-indexed formats the space should be left blank and should not be used by any other slots to send palette data. The number of palette entries for each slot can be at most 256.

7.3.2.3.4 History Buffer

The History Buffer is used to store inter-frame communication data. The structure holds cadence related information for each slot.

First dword of the hist_control (control_vector) should be set by the driver and informs Falcon whether to calculate the LUT for histogram enhancement and to calculate cadence or not.

If histogram enhancement is enabled for a slot and FALCON_CONTROL is set (explained in the SetControlParams() method) then Falcon calculates the LUT and updates the HistogramLutStruct's multiplier table of the config structure. If histogram enhancement is enabled and FALCON_CONTROL is 0, then driver should calculate the LUT. If the histogram enhancement flag is 0, then the table should be initialized to 1024.

Second part of the history buffer is used to store the 64-bin histogram data. This data is produced by the VIC and is used by either Falcon or driver to calculate the LUT.

7.3.2.3.5 FCE Microcode Buffer

This buffer holds the Fetch Control Engine (FCE) microcode that Falcon should load to the FCE unit. The driver should also pass the size of the FCE microcode with the SetFceUcodeSize() method.

7.3.2.3.6 Input Picture Buffers

Each picture is a separate buffer with its own surface offset method. The driver needs to send these offset methods for only the used pictures (source and reference) for the current execute.

7.3.2.3.7 Noise-Reduced Picture Buffers

This buffer is used by the app to keep a noise-reduced picture data of future fields if DeNoise is enabled. In the next execute, noise-reduced picture data is used as the current field, and noise-reduced picture data of the previous execute is used as previous picture. Noise reduction requires a forward and a backward reference field for interlaced content and a backward reference for progressive content.

In the NV24 case, the VIC app uses an additional surface to keep noise reduced field of the current picture. In this case, the current noise reduced surfaces are used as previous reference surfaces. Noise-reduced picture buffer dimensions for a slot should be same as input buffer of that slot.

7.3.2.3.8 Motion Map Buffers

These buffers keep motion map data of the current and previous fields if Motion Map is enabled. The previous motion map buffer should be initialized to 255 (both luma and chroma) at the start of

the clip. Use the motion map buffer of the previous execute pass prev motion map in the current execute for subsequent frames. Motion map calculation is required for DiSi1 and also needs a forward and a backward reference. Behavior is not defined for progressive streams. Motion map buffer dimensions for a slot should be same as input buffer of that slot.

7.3.2.3.9 Output Buffer

This buffer stores output data.

7.3.2.3.10 CRC Buffer

The InterfaceCrcStruct and InputCrcStruct are shared to store the CRC information. These are selected based on method SetCrcMode.

Programming for LDC and TNR3

7.3.2.3.11 LDC Related Class Define

Table 7.65 LDCConfigParamStruct

Name	Data Type	Offset	Notes
GeoTranEn	fixed<0, 1,0>	0	Geometry transform processing enable: 0: Disable / Bypass 1: Enable
GeoTranMode	fixed<0, 2,0>	1	Geometry transform processing modes: 0: Sparse warp mode 1: IPT-only mode x: Reserved
IPTMode	fixed<0, 1,0>	3	Inverse Perspective Transform mode: 0: Perspective transform 1: Affine transform
PixelFilterType	fixed<0, 2,0>	4	Pixel interpolation filter type: 0: 1-tap, Nearest 1: 2-tap, Bilinear 2: 4-tap, Bicubic 3: Reserved
PixelFormat	fixed<0, 7,0>	6	Pixel format of input and output surface
CacheWidth	fixed<0, 3,0>	13	Number of horizontal bytes per surface-cache cache-line. Each 256B cache line can store a source region of size as enumerated below. 0: 16Bx16 (BL16Bx2) 2: 64Bx4 (PL) x: Prohibited
SrcBlkKind	fixed<0, 4,0>	16	Source Block-linear kind: 0: Pitch Linear (PL) 1: Block Linear (BL16Bx2)

Name	Data Type	Offset	Notes
SrcBlkHeight	fixed<0, 4,0>	20	Block-linear height (in gobs, log2): 0: 1 gob 1: 2 gobs 2: 4 gobs 3: 8 gobs 4: 16 gobs (normal) 5: 32 gobs x: Reserved
DestBlkKind	fixed<0, 4,0>	24	Destination Block-linear kind: 0: Pitch Linear (PL) 1: Block Linear (BL16Bx2)
DestBlkHeight	fixed<0, 4,0>	28	Block-linear height (in gobs, log2): 0: 1 gob 1: 2 gobs 2: 4 gobs 3: 8 gobs 4: 16 gobs (normal) 5: 32 gobs x: Reserved
MskBitMapEn	fixed<0, 1,0>	32	External Mask Bit Map Feature Enable:
MaskedPixelFillMode	fixed<0, 1,0>	33	Masked pixel in destRect region control bit: 2'b0: Do not write these pixel data 2'b1: Fill with default color
XSobelMode	fixed<0, 2,0>	34	XSobel control bit: 2'b00: Disable xsobel and disable 4x4 down sample 2'b01: Disable/bypass xsobel and enable 4x4 down sample 2'b10: Enable xsobel and disable 4x4 downs ample 2'b11: Enable xsobel and enable 4x4 down sample
SubFrameEn	fixed<0, 1,0>	36	Sub Frame mode kick-off
XSobelBlkKind	fixed<0, 4,0>	40	Xsobel gradient image Block-linear kind: 0: Pitch Linear (PL) 1: Block Linear (BL16Bx2)
XSobelBlkHeight	fixed<0, 4,0>	44	Xsobel gradient image Block-linear height (in gobs, log2): 0: 1 gob 1: 2 gobs 2: 4 gobs 3: 8 gobs 4: 16 gobs (normal) 5: 32 gobs x: Reserved
XSobelDSBlkKind	fixed<0, 4,0>	48	Xsobel gradient downsample image Block-linear kind: 0: Pitch Linear (PL) 1: Block Linear (BL16Bx2)

Name	Data Type	Offset	Notes
XSobelDSBlkHeight	fixed<0,4,0>	52	Xsobel gradient downsample image Block-linear height (in gobs, log2): 0: 1 gob 1: 2 gobs 2: 4 gobs 3: 8 gobs 4: 16 gobs (normal) 5: 32 gobs x: Reserved
NonFixedPatchEn	fixed<0,1,0>	64	Non-Fixed Patch support in the whole frame: Row and column based patch size control 0: Disable, normal mode 1: Enable
HorRegionNum	fixed<0,2,0>	65	Horizontal region number: 0: 1 horizontal region 1: 2 horizontal regions 2: 3 horizontal regions 3: 4 horizontal regions
VerRegionNum	fixed<0,2,0>	67	Vertical region number: 0: 1 vertical region 1: 2 vertical regions 2: 3 vertical regions 3: 4 vertical regions
log2HorSpace_0	fixed<0,3,0>	72	log2 value of Horizontal spacing factor; log2 value of Vertical spacing factor. Horizontal or Vertical Space size in one patch of one region: 3'b0: 1 pixel (full warp map) 3'b1: 2 pixels (half warp map) 3'b2: 4 pixels (quad warp map) 3'b3: 8 pixels 3'b4: 16 pixels 3'b5: 32 pixels 3'b6: 64 pixels 3'b7: 128 pixels
log2VerSpace_0	fixed<0,3,0>	75	
log2HorSpace_1	fixed<0,3,0>	78	
log2VerSpace_1	fixed<0,3,0>	81	
log2HorSpace_2	fixed<0,3,0>	84	
log2VerSpace_2	fixed<0,3,0>	87	
log2HorSpace_3	fixed<0,3,0>	90	
log2VerSpace_3	fixed<0,3,0>	93	
horRegionWidth_0	fixed<0,14,0>	96	Horizontal Region width minus 1

Name	Data Type	Offset	Notes
horRegionWidth_1	fixed<0, 14,0>	112	
horRegionWidth_2	fixed<0, 14,0>	128	
horRegionWidth_3	fixed<0, 14,0>	144	
verRegionHeight_0	fixed<0, 14,0>	160	Vertical Region height minus 1
verRegionHeight_1	fixed<0, 14,0>	176	
verRegionHeight_2	fixed<0, 14,0>	192	
verRegionHeight_3	fixed<0, 14,0>	208	
IPT_M11	fixed<0, 32,0>	224	Inverse Perspective Transform 3x3 Matrix M IPT matrix coefficients. IEEE754 float point compliant format: 1S + 8E + 23F [M11, M12, M13] [M21, M22, M23] [M31, M32, M33]
IPT_M12	fixed<0, 32,0>	256	
IPT_M13	fixed<0, 32,0>	288	
IPT_M21	fixed<0, 32,0>	320	
IPT_M22	fixed<0, 32,0>	352	
IPT_M23	fixed<0, 32,0>	384	
IPT_M31	fixed<0, 32,0>	416	
IPT_M32	fixed<0, 32,0>	448	
IPT_M33	fixed<0, 32,0>	480	
SourceRectLeft	fixed<0, 14,0>	512	The source rectangle defines the region of pixels that are read from the source surface. Any pixel data outside of this is not used inside VIC but might still be read.
SourceRectRight	fixed<0, 14,0>	528	
SourceRectTop	fixed<0, 14,0>	544	

Name	Data Type	Offset	Notes
SourceRectBottom	fixed<0, 14,0>	560	
SrcImgWidth	fixed<0, 14,0>	576	Source image width minus 1 in pixel unit
SrcImgHeight	fixed<0, 14,0>	592	Source image height minus 1 in pixel unit
SrcSfcLumaWidth	fixed<0, 14,0>	608	Source Surface Luma Stride Actual_stride_in_bytes = (SrcSfcLumaWidth + 1) * 64 Bytes For PL: Actual_stride_in_bytes should be 256 Bytes aligned
SrcSfcLumaHeight	fixed<0, 14,0>	624	Source Surface Luma Height minus 1
SrcSfcChromaWidth	fixed<0, 14,0>	640	Source Surface Chroma Stride Actual_stride_in_bytes = (SrcSfcChromaWidth + 1) * 64 Bytes For PL: Actual_stride_in_bytes should be 256 Bytes aligned
SrcSfcChromaHeight	fixed<0, 14,0>	656	Source Surface Chroma Height minus 1
DestRectLeft	fixed<0, 14,0>	672	DestRect indicates destination frame resolution and location in the destination surface Dest Frame_width = DestRectRight – DestRectLeft + 1 Dest Frame_height = DestRectBottom – DestRectTop + 1
DestRectRight	fixed<0, 14,0>	688	
DestRectTop	fixed<0, 14,0>	704	
DestRectBottom	fixed<0, 14,0>	720	
SubFrameRectTop	fixed<0, 14,0>	736	SubFrameRect resolution: The value is related to DestRectTop. SubFrameRectTop and SubFrameRectBottom should be within 0 and Dest Frame_height-1
SubFrameRectBottom	fixed<0, 14,0>	752	
DestSfcLumaWidth	fixed<0, 14,0>	768	Dest Surface Luma Stride Actual_stride_in_bytes = (DestSfcLumaWidth + 1) * 64 Bytes Actual_stride_in_bytes should be one tile aligned
DestSfcLumaHeight	fixed<0, 14,0>	784	Destination Surface Luma Height minus 1 Actual_DestSfcLumaHeight = DestSfcLumaHeight + 1 Actual_DestSfcLumaHeight should be one tile aligned
DestSfcChromaWidth	fixed<0, 14,0>	800	Dest Surface Chroma Stride Actual_stride_in_bytes = (DestSfcChromaWidth + 1) * 64 Bytes Actual_stride_in_bytes should be one tile aligned
DestSfcChromaHeight	fixed<0, 14,0>	816	Destination Surface Chroma Height minus 1 Actual_DestSfcChromaHeight = DestSfcChromaHeight + 1 Actual_DestSfcChromaHeight should be one tile aligned

Name	Data Type	Offset	Notes
SparseWarpMapWidth	fixed<0, 14,0>	832	SparseWarpMap horizontal control point number minus 1
SparseWarpMapHeight	fixed<0, 14,0>	848	SparseWarpMap vertical control point number minus 1 SparseWarpMapHeight + 1 should cover DestRect's tile-aligned region Ex: DestRectBottom = 1079, DestRectTop = 0, log2VerSpace_0 = 2 SparseWarpMapHeight + 1 = (ALIGNUP(1079–0+1, 16) >> 2) + 1
SparseWarpMapStride	fixed<0, 14,0>	864	SparseWarpMap Surface Stride (control points line) Actual_stride_in_bytes = (SparseWarpMapStride + 1) * 64 Bytes
MaskBitMapWidth	fixed<0, 14,0>	896	MaskBitMap horizontal pixel number minus 1, related to DestRect
MaskBitMapHeight	fixed<0, 14,0>	912	MaskBitMap vertical pixel number minus 1, related to DestRect
MaskBitMapStride	fixed<0, 14,0>	928	MaskBitMap Surface Stride: (one tile line, 128 Bytes per tile) Actual_stride_in_bytes = (MaskBitMapStride + 1) * 64 Bytes
XSobelWidth	fixed<0, 14,0>	960	XSobel output gradient image width minus 1
XSobelHeight	fixed<0, 14,0>	976	XSobel output gradient image height minus 1
XSobelStride	fixed<0, 14,0>	992	Xsobel Gradient Surface Stride (pixel line) Actual_stride_in_bytes = (XSobelStride + 1) * 64 Bytes
DSStride	fixed<0, 14,0>	1008	Xsobel Gradient downsample Surface Stride Actual_stride_in_bytes = (DSStride + 1) * 64 Bytes
XSobelTopOffset	fixed<0, 32,0>	1024	PP Top Offset in the neighbor surface (Left Buffer + Top buffer)
maskY	fixed<0, 16,0>	1088	Masked default color Y component (MSB aligned)
maskU	fixed<0, 16,0>	1104	Masked default color U component (MSB aligned)
maskV	fixed<0, 16,0>	1120	Masked default color V component (MSB aligned)

7.3.2.4 Application Methods

The VIC application method registers are accessed directly by the driver. Because the VIC uses the shared MSDEC Falcon OS, two ranges of methods are used (in the MSDEC, there are common and application specific ranges). The range of method registers is:

- 0x100-0x6FF, 0x1114, (OS methods range)
- 0x700-0x7FF (APP common method range)
- 0x1200-0x17FF (method range corresponding to microcode overlay 1)

VIC has its own class definition, which include registers named METHOD_OFFSET and METHOD_DATA that are interpreted by the THI and are used to create method writes (corresponding to methods in the VIC class specification) into the VIC. These methods are used to set up pointers to the various surfaces that the VIC needs to read or write, as well as pointers to config structures needed. The VIC class also defines trigger methods such as Execute(), used to start the actual processing of the data for the next output image.

Table 7.66 Method Mapping

Address	Method	Comment
0x0100	VIC.Nop	
0x0140	VIC.PmTrigger	
0x0200	VIC.SetApplicationID	1: Legacy compositor 2: New engine
0x0204	VIC.SetWatchdogTimer	
0x0240	VIC.SemaphoreA	
0x0244	VIC.SemaphoreB	
0x0248	VIC.SemaphoreC	
0x024c	VIC.CtxSaveArea	
0x0250	VIC.CtxSwitch	
0x0300	VIC.Execute	
0x0304	VIC.SemaphoreD	
0x0700	VIC.SetPictureIndex	
0x0704	VIC.SetControlParams	
0x0708	VIC.SetConfigStructOffset	Class input surface
0x070c	VIC.SetFilterStructOffset	
0x0710	VIC.SetPaletteOffset	
0x0714	VIC.SetHistOffset	
0x0718	VIC.SetContexID	
0x071c	VIC.SetFceUcodeSize	
0x0720	VIC.SetOutputSurfaceLumaOffset	Output Luma surface
0x0724	VIC.SetOutputSurfaceChromaU_Offset	Output Chroma U surface
0x0728	VIC.SetOutputSurfaceChromaV_Offset	Output Chroma V surface
0x072c	VIC.SetFceUcodeOffset	
0x0730	VIC.SetCrcStructOffset	

Address	Method	Comment
0x0734	VIC.SetCrcMode	
0x0738	VIC.SetStatusOffset	
0x0800	VIC.SetSparseWarpMap_Offset	Sparse warpmap surface
0x0804	VIC.SetMaskBitMap_Offset	Mask BitMap surface
0x0808	VIC.SetXsobelSurface_Offset	XSobel gradient image surface
0x080c	VIC.SetXsobelDsSurface_Offset	4x4 downsample image surface
0x0810	VIC.SetXSobelNeighborBuffer_Offset	XSobel neighbor buffer
0x0814	VIC.SetTNR3PrevFrmSurfaceLumaOffset	TNR previous frame Luma surface
0x0818	VIC.SetTNR3PrevFrmSurfaceChromaU_Offset	TNR previous frame Chroma U surface
0x081c	VIC.SetTNR3PrevFrmSurfaceChromaV_Offset	TNR previous frame Chroma V surface
0x0820	VIC.SetTNR3CurAlphaSurfaceOffset	TNR current alpha surface
0x0824	VIC.SetTNR3PrevAlphaSurfaceOffset	TNR previous alpha surface
0x0828	VIC.SetTNR3NeighborBufferOffset	TNR neighbor buffer
0x082c	VIC.SetStatusNotifierInputOffset	STATUS Notifier Input buffer
0x0830	VIC.SetStatusNotifierOutputOffset	STATUS Notifier Output buffer
0x1114	VIC.PmTriggerEnd	
0x1200	VIC.SetSurface0LumaOffset[0]	Input Luma surface
0x1204	VIC.SetSurface0ChromaU_Offset[0]	VIC.SetSurface0ChromaU_Offset[0]
0x1208	VIC.SetSurface0ChromaV_Offset[0]	VIC.SetSurface0ChromaV_Offset[0]

7.3.2.5 Method Naming and Programming

The VIC uses generic names to avoid confusing name overloading to the 24 surfaces (eight luma, eight chroma_u, eight chroma_v) methods for the MAX_SLOTS

- SurfacexSlotyLumaOffset[y]
- SetSurfacexSlotyChromaU_Offset[y]

SetSurfacexSlotyChromaV_Offset[y] with x in [0..7] and y in [0..MAX_SLOTS-1] The following sets of methods are required for a given ConfigStruct setup. Any setup not covered in this section should be regarded as illegal.

Notes:

- NV24 frames are treated as NV12 fields. The frame format for NV24 has to be set to FRAME_FORMAT_TOP_FIELD.
- In case of noise reduction, all previously noise reduced surfaces should be used as references instead of the original ones. In case of interlaced fields this also applies to the current fields as they have previously been noise reduced.
- Surface6 has been obsoleted and should not be used.
- History buffer is always required and should be owned by Falcon.

7.3.2.5.1 Application ID

Application ID is not needed as there is only one application for the engine (unlike different codecs for the MSDEC). But to use it with MSDEC infrastructure, the driver needs to pass the ApplicationId as 1 (corresponding to overlay 1).

7.3.2.5.2 Progressive Frames

No Preprocessing

Enable bits have to be set to 0x01 in fetchControl0Struct.
Required methods for all pixel formats:

- SetSurface0SlotLumaOffset()

Required methods for pixel formats with separate chroma plane:

- SetSurface0SlotChromaOffset()

Noise Reduction

Enable bits have to be set to 0x07 in fetchControl0Struct.
Required methods for all pixel formats:

- SetSurface0SlotLumaOffset()
- SetSurface1SlotLumaOffset() (backward reference surface)
- SetSurface2SlotLumaOffset() (noise reduced surface)

Required methods for pixel formats with separate chroma plane:

- SetSurface0SlotChromaOffset()
- SetSurface1SlotChromaOffset() (backward reference surface)
- SetSurface2SlotChromaOffset() (noise reduced surface)

7.3.2.5.3 Progressive Fields

No Preprocessing

Enable bits have to be set to 0x03 in fetchControl0Struct.
Required methods for all pixel formats:

- SetSurface0SlotLumaOffset() (top field)
- SetSurface1SlotLumaOffset() (bottom field)

Required methods for pixel formats with separate chroma plane:

- SetSurface0SlotChromaOffset() (top field)
- SetSurface1SlotChromaOffset() (bottom field)

Noise Reduction

Enable bits have to be set to 0x03f in fetchControl0Struct.
Required methods for all pixel formats:

- SetSurface0SlotLumaOffset() (top field)
- SetSurface1SlotLumaOffset() (bottom field)
- SetSurface2SlotLumaOffset() (top field of backward reference surface)
- SetSurface3SlotLumaOffset() (bottom field of backward reference surface)
- SetSurface4SlotLumaOffset() (top field of noise reduced surface)
- SetSurface5SlotLumaOffset() (bottom field of noise reduced surface)

Required methods for pixel formats with separate chroma plane:

- SetSurface0SlotChromaOffset() (top field)
- SetSurface1SlotChromaOffset() (bottom field)
- SetSurface2SlotChromaOffset() (top field of backward reference surface)
- SetSurface3SlotChromaOffset() (bottom field of backward reference surface)
- SetSurface4SlotChromaOffset() (top field of noise reduced surface)
- SetSurface5SlotChromaOffset() (bottom field of noise reduced surface)

7.3.2.5.4 Interlaced Frames

No Preprocessing

Enable bits have to be set to 0x01 in fetchControl0Struct.
Required methods for all pixel formats:

- SetSurface0SlotLumaOffset()

Required methods for pixel formats with separate chroma plane:

- SetSurface0SlotChromaOffset()

Noise Reduction

Enable bits have to be set to 0x07 in fetchControl0Struct.
Required methods for all pixel formats:

- SetSurface0SlotLumaOffset()
- SetSurface1SlotLumaOffset() (backward reference surface)
- SetSurface2SlotLumaOffset() (noise reduced surface)

Required methods for pixel formats with separate chroma plane:

- SetSurface0SlotChromaOffset()
- SetSurface1SlotChromaOffset() (backward reference surface)
- SetSurface2SlotChromaOffset() (noise reduced surface)

7.3.2.5.5 Interlaced Fields

No Preprocessing

This is for BOB_FIELD only. For WEAVE, see No Preprocessing under VIC Programming Guidelines.
Enable bits have to be set to 0x01 in fetchControl0Struct.

Required methods for all pixel formats:

- SetSurface0SlotLumaOffset()

Required methods for pixel formats with separate chroma plane:

- SetSurface0SlotChromaOffset()

Cadence Detection

Enable bits have to be set to 0x07 in fetchControl0Struct. This also requires cadence detection to be enabled.
Required methods for all pixel formats:

- SetSurface0SlotLumaOffset()
- SetSurface1SlotLumaOffset() (backward reference field)
- SetSurface2SlotLumaOffset() (forward reference field)

Required methods for pixel formats with separate chroma plane:

- SetSurface0SlotChromaOffset()
- SetSurface1SlotChromaOffset() (backward reference field)

- SetSurface2SlotChromaOffset() (forward reference field)

Motion Calculation

Enable bits have to be set to 0x17 in fetchControl0Struct. The data in the current motion field is read first and IIR filtered with the newly calculated motion before it is written back.
Required methods for all pixel formats:

- SetSurface0SlotLumaOffset()
- SetSurface1SlotLumaOffset() (backward reference field)
- SetSurface2SlotLumaOffset() (forward reference field)
- SetSurface4SlotLumaOffset() (current motion field)

Required methods for pixel formats with separate chroma plane:

- SetSurface0SlotChromaOffset()
- SetSurface1SlotChromaOffset() (backward reference field)
- SetSurface2SlotChromaOffset() (forward reference field)
- SetSurface4SlotChromaOffset() (current motion field)

Motion Calculation and Cadence Detection

Same as above in Motion Calculation except that cadence detection is enabled.

DISI1 (Motion Calculation and Motion Combine)

Enable bits have to be set to 0xb7 in fetchControl0Struct. The data in the current motion field is read first and IIR filtered with the newly calculated motion before it is written back. The combined motion field is generated out of current motion and previous motion to be used in DiSi1 deinterlacer.
Required methods for all pixel formats:

- SetSurface0SlotLumaOffset()
- SetSurface1SlotLumaOffset() (backward reference field)
- SetSurface2SlotLumaOffset() (forward reference field)
- SetSurface4SlotLumaOffset() (current motion field)
- SetSurface5SlotLumaOffset() (previous motion field)
- SetSurface7SlotLumaOffset() (combined motion field)

Required methods for pixel formats with separate chroma plane:

- SetSurface0SlotChromaOffset()
- SetSurface1SlotChromaOffset() (backward reference field)
- SetSurface2SlotChromaOffset() (forward reference field)
- SetSurface4SlotChromaOffset() (current motion field)

- SetSurface5SlotChromaOffset() (previous motion field)
- SetSurface7SlotChromaOffset() (combined motion field)

DISI1 and Cadence Detection

Same as above in DiSi1 except that cadence detection is enabled.

Noise Reduction

Enable bits have to be set to 0x0f in fetchControl0Struct.
Required methods for all pixel formats:

- SetSurface0SlotLumaOffset()
- SetSurface1SlotLumaOffset() (previously noise reduced backward reference field)
- SetSurface2SlotLumaOffset() (forward reference field)
- SetSurface3SlotLumaOffset() (noise reduced forward reference field)

Required methods for pixel formats with separate chroma plane:

- SetSurface0SlotChromaOffset()
- SetSurface1SlotChromaOffset() (previously noise reduced backward reference field)
- SetSurface2SlotChromaOffset() (forward reference field)
- SetSurface3SlotChromaOffset() (noise reduced forward reference field)

Noise Reduction and Cadence Detection

Same as above in Noise Reduction except that cadence detection is enabled.

Noise Reduction and Motion Calculation

Enable bits have to be set to 0x1f in fetchControl0Struct. The data in the current motion field is read first and IIR filtered with the newly calculated motion before it is written back.
Required methods for all pixel formats:

- SetSurface0SlotLumaOffset()
- SetSurface1SlotLumaOffset() (backward reference field)
- SetSurface2SlotLumaOffset() (forward reference field)
- SetSurface3SlotLumaOffset() (noise reduced forward reference field)
- SetSurface4SlotLumaOffset() (current motion field)

Required methods for pixel formats with separate chroma plane:

- SetSurface0SlotChromaOffset()
- SetSurface1SlotChromaOffset() (backward reference field)
- SetSurface2SlotChromaOffset() (forward reference field)

- SetSurface3SlotChromaOffset() (noise reduced forward reference field)
- SetSurface4SlotChromaOffset() (current motion field)

Noise Reduction and Motion Calculation and Cadence Detection

Same as above in Noise Reduction and Motion Calculation except that cadence detection is enabled.

Noise Reduction and DISI1

Enable bits have to be set to 0xbf in fetchControl0Struct. The data in the current motion field is read first and IIR filtered with the newly calculated motion before it is written back. The combined motion field is generated out of current motion and previous motion to be used in DiSi1 deinterlacer.
Required methods for all pixel formats:

- SetSurface0SlotLumaOffset()
- SetSurface1SlotLumaOffset() (backward reference field)
- SetSurface2SlotLumaOffset() (forward reference field)
- SetSurface3SlotLumaOffset() (noise reduced forward reference field)
- SetSurface4SlotLumaOffset() (current motion field)
- SetSurface5SlotLumaOffset() (previous motion field)
- SetSurface7SlotLumaOffset() (combined motion field)

Required methods for pixel formats with separate chroma plane:

- SetSurface0SlotChromaOffset()
- SetSurface1SlotChromaOffset() (backward reference field)
- SetSurface2SlotChromaOffset() (forward reference field)
- SetSurface3SlotChromaOffset() (noise reduced forward reference field)
- SetSurface4SlotChromaOffset() (current motion field)
- SetSurface5SlotChromaOffset() (previous motion field)
- SetSurface7SlotChromaOffset() (combined motion field)

Noise Reduction and DISI1 and Cadence Detection

Same as above in Noise Reduction and DiSi1 except that cadence detection is enabled.

7.3.2.6 Programming Restrictions

7.3.2.6.1 Input Parameters

- When a slot is enabled in the slotConfig structure, it should have one or more surfaces enabled in the SurfaceEnable bits.
- The source and destination rectangles should be of non-zero sizes (i.e., right >= left and bottom >= top).
- Since the Luma Width/Height parameters represent padded surface sizes, the values in these parameters should be greater than or equal to the corresponding surface width/height parameters.
- For single plane surfaces like RGB surfaces, the Chroma Width/Height parameters should be programmed to zero since the Falcon microcode uses these parameters to check for the existence of multi-plane formats.
- The Chroma Width/Height parameters should be greater than or equal to the sub-sampled size of the surface width/height parameters.
- For surfaces with sub-sampled chroma
- The source rectangle should be big enough to have at least one chroma sample within the rectangle.
- The input surface size should be a multiple of two in the direction of the sub-sampling
- For operations involving video pre-process operations like IVTC, TNR/TNR2, DiSi1 de-interlacing, the supported pixel formats are restricted as below.

Table 7.67 Supported Pixel Formats

Operations	Pixel Formats
De-interlacing/ Cadence Detection	T_Y8__*V8U8_N420, T_Y8_U8_V8_N420, T_Y8_U8 Y8_V8, T_U8_Y8*_V8_Y8
TNR/TNR2	T_Y8__*V8U8_N420, T_Y8_U8_V8_N420, T_Y8_U8 Y8_V8, T_U8_Y8*_V8_Y8

- When the input uses interlaced surfaces, there are the following additional restrictions:
- The source rectangle height should be a multiple of 2 to account for the surface being comprised of two fields.
- If the input pixel format uses chroma sub-sampling in the vertical direction (420 or 422R), then the source rectangle height should be a multiple of 4.
- Note that the source rectangle is always specified in "frame coordinates". Therefore, when de-interlacing two fields of height y each, the source rectangle height should be within $2*y$.

Scaling

- When panoramic scaling is enabled
- The destination rectangle should be at least 4 pixels wide and 4 pixels tall

- The maximum downscaling ratio is 7:1
- For non-panoramic scaling
- The maximum downscaling ratio is 16:1
- When enabling the filter override mode (see DownsampleVert/Horiz), only five slots can be enabled. This is because of the available size of the coefficient structure.

Luma Keying

- When Luma keying is enabled, the luma key range should be non-empty (i.e., lower <= upper)

Soft Clamping

- The soft clamping lower and upper thresholds should be set such that the lower threshold is less than or equal to the upper threshold at all times

De-interlacing

- The BOB de-interlacing mode is designed to work on interlaced-frame input (i.e., the source is interlaced, but the two fields are stored interleaved together in a single frame-surface.)
- The BOB_FIELD, DISI1, NEWBOB de-interlacing modes are designed to work on interlaced-field input (i.e., the fields are stored as separate surfaces.)
- The only allowed de-interlacing mode for progressive inputs is WEAVE, where it behaves as a de-interlacing bypass.
- For interlaced inputs, the WEAVE algorithm expects field surfaces and weaves them together.

Cadence Detection

- Requires field based surfaces
- Needs forward and backward reference surfaces enabled

Motion Calculation

- Motion map calculation requires field based surfaces with forward and backward reference surfaces and writes out a current motion surface
- Combine motion map calculation requires field based surfaces with current and previous motion surfaces enabled and writes out a combined motion surface

Noise Reduction

- The allowed range for the IIR strength parameter is 0x0 to 0x400 (inclusive)

Field Based Noise Reduction

- Requires forward and backward reference surfaces enabled, as well as a noise reduced output surface

Frame Based Noise Reduction

- Requires current and backward reference surfaces enabled, as well as a noise reduced output surface

Output Parameters

- The list of pixel formats supported on the output is not exactly the same as the list of input pixel formats. For example, the palette pixel formats are not supported on output.
- When output transformations are enabled, then all destination and clear rectangles should lie entirely within the target rectangle.
- Since the Luma Width/Height parameters represent padded surface sizes, the values in these parameters should be greater than or equal to the corresponding surface width/height parameters.
- The Chroma Width/Height parameters should be greater than or equal to the sub-sampled size of the surface width/height parameters.
- For surfaces with sub-sampled chroma
- The target and destination rectangles should be big enough to have at least one chroma sample within the rectangle.
- The output surface size should be a multiple of two in the direction of the sub-sampling

Target Rectangle

- Target rectangle should lie entirely within the output surface (no negative co-ordinates, or co-ordinates outside of the output surface size).
- The target rectangle should be non-empty (i.e., right >= left and bottom >= top).

Clear Rectangles

- When a clear rectangle is enabled, it should have a non-zero width and height (i.e., right >= left and bottom >= top).

Alpha Fill Mode

- For alpha fill mode _SOURCE_ALPHA, the AlphaFillSlot parameter should be set to a valid and enabled slot

7.3.2.7 Performance and Power Tuning Guidelines

7.3.2.7.1 Fully Opaque/Fully Transparent Surfaces

VIC hardware cannot know when a certain slot is fully opaque or fully transparent. Thus, when for example a fully opaque slot lies over other slots, the data for the other slots is still fetched from memory and is blended with the data for the opaque slot, even though the slots below the opaque slot do not affect the output. Software should therefore program VIC to avoid the fetch of the data

below the opaque surface by putting a clear rectangle over the area beneath the opaque slot, and enabling the ClearRectMask for all the slots below the opaque slot. This software enhancement, when implemented should significantly reduce the VIC bandwidth requirements, and also enhance the VIC performance for use cases involving opaque surfaces.

7.3.2.7.2 Smaller Target Rectangles

For cases where the displayed image does not cover all of the screen, software should optimize bandwidth by only generating/compositing the portion of the screen which actually has usable data, and let either Display (preferably), or VIC fill in the rest of the screen with a background color. This saves both read and write bandwidth in such use cases.

7.3.2.7.3 Memory Formats

The CacheWidth parameter in the VIC ConfigStruct defines the mapping of the 256B surface cache cache-lines to rectangular screen space. Based on our performance testing simulations, the recommendation for programming this parameter is as below

- For pitch-linear surfaces, set the CacheWidth to 64Bx4 - i.e., CacheWidth=2
- For block-linear surfaces, set the CacheWidth to 32Bx8 - i.e., CacheWidth=1

7.3.2.7.4 Clock Gating

VIC has three levels of clock gating. Software should ensure that 1st and 2nd level clock gating is enabled for VIC at all times so that the VIC hardware can try to turn off clock branches whenever any portion of the logic is unused.

- 2nd level clock gating control: Bit 5 of CLK_RST_CONTROLLER_LVL2_CLK_GATE_OVRE_0 should be set to 0.
- NV_PVIC_THI_SLCG_OVERRIDE_HIGH_A and NV_PVIC_THI_SLCG_OVERRIDE_LOW_A should both be set to 0x0 to enable automatic clock-gating of VIC sub-units when idle.

7.3.2.7.5 Power Gating

The VIC partition(s) can be power-gated when not in use, and software should put VIC in power gate mode for any use case that does not make use of VIC. Software should follow the power gating/un-gating in order to put VIC in PG mode, or take VIC out of PG mode.

7.3.2.7.6 Parameter Constraints

Table 7.68 Supported Work Modes

Work mode	GeoTranEn	GeoTran Mode	TNR3En	XSobel Mode	MskBitMapEn	SubFrameEn
New Blit	0	x	0	0	0	0/1

Work mode	GeoTranEn	GeoTran Mode	TNR3En	XSobel Mode	MskBitMapEn	SubFrameEn
Rectify + Xsobel	1	1	0	0/1/2/3	0/1	0/1
LDC + Xsobel	1	0	0	0/1/2/3	0/1	0/1
TNR3	0	x	1	0	0	0/1
Rectify/LDC+TNR3	1	1/0	1	0	0/1	0/1

Notes:

1. TNR and Xsobel can't work together.
2. MaskBitMap must work together with Rectify or LDC.
3. IPTBlit mode means Rectify work mode with Identity IPT matrix coefficients.

Table 7.69 Supported Pixel Format For Different Work Modes

Pixel Format	New Blit	Rectify/LDC	TNR3	Rectify/LDC+TNR3
T_R8	Y	Y		
T_Y8___V8U8_N444		Y		
T_Y8___V8U8_N422	Y	Y		
T_Y8___V8U8_N420	Y	Y	Y	Y
T_R16	Y	Y		
T_Y16___V16U16_N444		Y		
T_Y16___V16U16_N422	Y	Y		
T_Y16___V16U16_N420	Y	Y		
T_Y10___V10U10_N420 (YUV420 semi-planar)	Y	Y	Y	Y
T_Y12___V12U12_N420 (YUV420 semi-planar)	Y	Y	Y	Y
T_Y8_U8__Y8_V8 (YUV422 packed) (YUY2/YUYV)	Y		Y	
T_U8_Y8__V8_Y8 (YUV422 packed) (UYVY)	Y		Y	
T_Y8__*U8*__V8_N420 (YUV420 planar, YV12)	Y		Y	
T_Y10__*U10*__V10_N420 (YUV420 planar)	Y		Y	
T_Y12__*U12*__V12_N420 (YUV420 planar)	Y		Y	

Notes: T_Y8__*V8U8_N444, T_Y16*__V16U16_N444 can use IPTBlit mode.

Table 7.70 Parameter Constraint Table

Class parameter	Constraints
horRegionWidth_0/1/2/3 verRegionHeight_0/1/2/3	horRegionWidth must be the multiple of 64 pixels expect the most right valid region. verRegionRegion must be the multiple of 16 pixels expect the most bottom valid region. Sum of all valid horRegionWidth and verRegionHeight should be equal to DestFrame_Width and DestFrame_Height
SrcImgWidth SrcImgHeight	SrcImgWidth plus 1 and SrcImgHeight plus 1 should be even number. SrcImgWidth and SrcImgHeight should be larger than 64x16. SrcImgWidth and SrcImgHeight should be larger than 16384x16384.
SrcSfcLumaWidth SrcSfcLumaHeight SrcSfcChromaWidth SrcSfcChromaHeight	For BL, actual_stride_in_bytes for luma and chroma surface should be 64 Bytes aligned. For PL, actual_stride_in_bytes for luma and chroma surface should be 256 Bytes aligned. For NewBlit and TNR3 work mode, actual_stride_in_bytes for luma and chroma surface should also be one tile's pixel width aligned. For PL, SrcSfcLumaHeight and SrcSfcChromaHeight should be 4 pixel aligned. For BL, SrcSfcLumaHeight and SrcSfcChromaHeight should be GOB height aligned. For NewBlit and TNR3 work mode, SrcSfcLumaHeight and SrcSfcChromaHeight should also be one tile's pixel height aligned.
SourceRectLeft SourceRectRight SourceRectTop SourceRectBottom	SourceRectLeft and SourceRectTop should be even number. SourceRectWidth and SourceRectHeight should be even number. SourceRect should be within SRC image region.
DestRectLeft DestRectRight DestRectTop DestRectBottom	DestRectLeft and DestRectTop should be even number. DestRectWidth and DestRectHeight should be even number. DestRect should be within dest surface region. DestRectWidth, DestRectHeight should be larger than 64x16. DestRectWidth, DestRectHeight should be smaller than 16. DestRectRight and DestRectBottom should be smaller than 16k.
DestSfcLumaWidth DestSfcLumaHeight DestSfcChromaWidth DestSfcChromaHeight	actual_stride_in_bytes for luma and chroma surface should be one tile's pixel width aligned. Actual_DestSfcLumaHeight and Actual_DestSfcChromaHeight should be one tile's pixel height aligned. For BL, Actual_DestSfcLumaHeight and Actual_DestSfcChromaHeight should be GOB height aligned.
SubFrameRectTop SubFrameRectBottom	SubFrameRectTop should be always 32-pixel height aligned. SubFrameRectBottom should be 32-pixel height aligned. Valid lines in each subframe >= 32
SparseWarpMapWidth SparseWarpMapHeight SparseWarpMapStride	Actual SparseWarpMapHeight should cover DestRect's tile-aligned region. Ex: DestRectBottom = 1079, DestRectTop = 0, log2VerSpace_0 = 2 SparseWarpMapHeight + 1 = (ALIGNUP (1079–0+1, 16) >> 2) + 1 Actual_stride_in_bytes should be 64 Bytes aligned. Different regions don't share any control points in sparse warp map.
MaskBitMapWidth MaskBitMapHeight MaskBitMapStride	MaskBitMapWidth and MaskBitMapHeight should be related to DestRectWidth and DestRectHeight. MaskBitMap Surface Stride should be one tile line (128 Bytes) aligned.

Class parameter	Constraints
XSobelWidth XSobelHeight XSobelStride DSStride	XSobelWidth and XSobelHeight should be related to DestRectWidth and DestRectHeight. XSobelStride and DSStride should be 256 Bytes aligned. Xsobel and DS surface height should be one tile's pixel height aligned, and for BL, they should be also GOB height aligned.

7.3.2.8 Program Guideline

The BlockLinear layout is the recommended setting for DestBlkKind, XSobelBlkKind, and XSobelDSBlkKind because of memory access bandwidth efficiency.

7.3.2.8.1 Method Interface

VIC has its own class definition, which include registers named METHOD_OFFSET and METHOD_DATA that are interpreted by the THI and are used to create method writes (corresponding to methods in the VIC class specification) into the VIC. These methods are used to set up pointers to the various surfaces that the VIC needs to read or write, as well as pointers to config structures needed. The VIC class also defines trigger methods such as Execute(), used to start the actual processing of the data for the next output image.

Address	Method	Comment
0x0100	VIC.Nop	
0x0140	VIC.PmTrigger	
0x0200	VIC.SetApplicationID	1: Legacy compositor 2: New engine
0x0204	VIC.SetWatchdogTimer	
0x0240	VIC.SemaphoreA	
0x0244	VIC.SemaphoreB	
0x0248	VIC.SemaphoreC	
0x024c	VIC.CtxSaveArea	
0x0250	VIC.CtxSwitch	
0x0300	VIC.Execute	
0x0304	VIC.SemaphoreD	
0x0700	VIC.SetPictureIndex	
0x0704	VIC.SetControlParams	
0x0708	VIC.SetConfigStructOffset	Class input surface
0x070c	VIC.SetFilterStructOffset	

Address	Method	Comment
0x0710	VIC.SetPaletteOffset	
0x0714	VIC. SetHistOffset	
0x0718	VIC.SetContexID	
0x071c	VIC.SetFceUcodeSize	
0x0720	VIC.SetOutputSurfaceLumaOffset	Output Luma surface
0x0724	VIC.SetOutputSurfaceChromaU_Offset	Output chroma U surface
0x0728	VIC.SetOutputSurfaceChromaV_Offset	Output chroma V surface
0x072c	VIC.SetFceUcodeOffset	
0x0730	VIC.SetCrcStructOffset	
0x0734	VIC.SetCrcMode	
0x0738	VIC.SetStatusOffset	
0x0800	VIC.SetSparseWarpMap_Offset	Sparse warpmap surface
0x0804	VIC.SetMaskBitMap_Offset	Mask BitMap surface
0x0808	VIC.SetXsobelSurface_Offset	XSobel gradient image surface
0x080c	VIC.SetXsobelDsSurface_Offset	4x4 downsample image surface
0x0810	VIC.SetXSobelNeighborBuffer_Offset	XSobel neighbor buffer
0x0814	VIC.SetTNR3PrevFrmSurfaceLumaOffset	TNR previous frame Luma surface
0x0818	VIC.SetTNR3PrevFrmSurfaceChromaU_Offset	TNR previous frame Chroma U surface
0x081c	VIC.SetTNR3PrevFrmSurfaceChromaV_Offset	TNR previous frame Chroma V surface
0x0820	VIC.SetTNR3CurAlphaSurfaceOffset	TNR current alpha surface
0x0824	VIC.SetTNR3PrevAlphaSurfaceOffset	TNR previous alpha surface
0x0828	VIC.SetTNR3NeighborBufferOffset	TNR neighbor buffer
0x082c	VIC.SetStatusNotifierInputOffset	STATUS Notifier Input buffer
0x0830	VIC.SetStatusNotifierOutputOffset	STATUS Notifier Output buffer
0x1114	VIC.PmTriggerEnd	
0x1200	VIC.SetSurface0LumaOffset[0]	Input Luma surface
0x1204	VIC.SetSurface0ChromaU_Offset[0]	Input Chroma U surface
0x1208	VIC.SetSurface0ChromaV_Offset[0]	Input Chroma V surface

7.3.2.8.2 TNR3 Related Class Define

TNR3 uses the same 10-bpp parameter setting for all 8-bit, 10-bit, and 12-bit use cases via internal pixel value normalization as a result of no 12-bpp requirement for deep-color/HDR in IVA/TP market.

TNR3 feature related class definitions are as follows:

Table 7.71 TNR3ConfigParamStruct

Name	Data Type	Offset	Notes
TNR3En	fixed<0,1,0> (1 bit)	0	Temporal Noise Reduction (TNR3) enable 0: Disable/Bypass 1: Enable
BetaBlendingEn	fixed<0,1,0> (1 bit)	1	Beta Blending enable with input frame and spatial filtered frame 0: Notemp frame = SF frame 1: Notemp frame = beta * SF frame + (1-beta) * Cur Frame
AlphaBlendingEn	fixed<0,1,0> (1 bit)	2	Alpha Blending enable with non-temporal frame and previous output frame 0: Output frame = notemp frame 1: Output frame = (1-alpha)*notemp frame + alpha * prev output frame
AlphaSmoothEn	fixed<0,1,0> (1 bit)	3	Spatial Alpha Smooth enable
TempAlphaRestrictEn	fixed<0,1,0> (1 bit)	4	Temporal alpha restrict enable
AlphaClipEn	fixed<0,1,0> (1 bit)	5	Alpha output clip enable
BFRangeEn	fixed<0,1,0> (1 bit)	6	Bilateral Range Filter enable 0: colorDist = 64 for any pixel value difference 1: Normal operation
BFDomainEn	fixed<0,1,0> (1 bit)	7	Bilateral Domain Filter enable 0: C00 = 64, CXY = 0, (X,Y≠0) 1: Normal operation
BFRangeLumaShift	fixed<0,4,0> (4 bits)	8	Bilateral Filter LumaRangeShift
BFRangeChromaShift	fixed<0,4,0> (4 bits)	12	Bilateral Filter ChromaRangeShift
SADMultiplier	fixed<0,6,0> (6 bits)	16	SAD Multiplier param (U3.3)
SADWeightLuma	fixed<0,6,0> (6 bits)	24	SAD Weight for luma channel (U1.5) SADWeightChroma = 32 - SADWeightLuma
TempAlphaRestrictIncCap	fixed<0,11,0> (11 bits)	32	Temporal Alpha Restrict Increase Capability (U1.10)

Name	Data Type	Offset	Notes
AlphaScaleIIR	fixed<0,11,0> (11 bits)	48	Alpha Scale IIR for strength (U1.10) 1024 indicates the max filter strength
AlphaClipMaxLuma	fixed<0,11,0> (11 bits)	64	Alpha Clip alpha Max Luma Value (U1.10)
AlphaClipMinLuma	fixed<0,11,0> (11 bits)	80	Alpha Clip alpha Min Luma Value (U1.10)
AlphaClipMaxChroma	fixed<0,11,0> (11 bits)	96	Alpha Clip for Max Chroma Value (U1.10)
AlphaClipMinChroma	fixed<0,11,0> (11 bits)	112	Alpha Clip for Min Chroma Value (U1.10)
BetaCalcMaxBeta	fixed<0,11,0> (11 bits)	128	Beta Calculation MaxBeta threshold (U1.10)
BetaCalcMinBeta	fixed<0,11,0> (11 bits)	144	Beta Calculation MinBeta threshold (U1.10)
BetaCalcBetaX1	fixed<0,11,0> (11 bits)	160	Beta Calculation BetaX1 (U1.10)
BetaCalcBetaX2	fixed<0,11,0> (11 bits)	176	Beta Calculation BetaX2 (U1.10)
BetaCalcStepBeta	fixed<0,11,0> (11 bits)	192	Beta Calculation Step Beta (U6.5) beta = maxBeta - (alpha – betaX1) * StepBeta StepBeta value here is positive.
BFDomainLumaCoeffC00	fixed<0,7,0> (7 bits)	224	TNR3 Bilateral Filter Coefficients CXY = CYX, X,Y = 0, 1, 2 C00, C01, C02, C11, C12, C22 6 coefficients are independent. Luma 5x5 filter: (U1.6) [C22 C12 C02 C12 C22 C12 C11 C01 C11 C12 C02 C01 C00 C01 C02 C12 C11 C01 C11 C12 C22 C12 C02 C12 C22]
BFDomainLumaCoeffC01	fixed<0,7,0> (7 bits)	232	
BFDomainLumaCoeffC02	fixed<0,7,0> (7 bits)	240	
BFDomainLumaCoeffC11	fixed<0,7,0> (7 bits)	248	
BFDomainLumaCoeffC12	fixed<0,7,0> (7 bits)	256	
BFDomainLumaCoeffC22	fixed<0,7,0> (7 bits)	264	
BFDomainChromaCoeffC00	fixed<0,7,0> (7 bits)	288	TNR3 Bilateral Filter Coefficients CXY = CYX, X,Y = 0, 1, 2 C00, C01, C02, C11, C12, C22 6 coefficients are independent. Chroma 5x5 filter: (U1.6) [C22 C12 C02 C12 C22 C12 C11 C01 C11 C12 C02 C01 C00 C01 C02 C12 C11 C01 C11 C12 C22 C12 C02 C12 C22]
BFDomainChromaCoeffC01	fixed<0,7,0> (7 bits)	296	
BFDomainChromaCoeffC02	fixed<0,7,0> (7 bits)	304	

Name	Data Type	Offset	Notes
BFDomainChromaCoeffC11	fixed<0,7,0> (7 bits)	312	
BFDomainChromaCoeffC12	fixed<0,7,0> (7 bits)	320	
BFDomainChromaCoeffC22	fixed<0,7,0> (7 bits)	328	
LeftBufSize	fixed<0,7,0> (7 bits)	352	LeftBufSize should be 256 Bytes aligned
TopBufSize	fixed<0,32,0> (32 bits)	384	TopBufSize should be 256 Bytes aligned
AlphaSufStride	fixed<0,14,0> (14 bits)	416	TNR3 Prev and Cur Alpha Surface Stride, (one tile 32x8 line, 256 Bytes per tile) Actual_stride_in_bytes = (AlphaSufStride + 1) * 64 Bytes

Since the first frame does not have a valid previous frame, and the second frame does not have a valid previous alpha, several work modes handled by software driver are defined to solve the issues:

Name	First frame	Second frame	Followed frame
BetaBlendingEn	0	1	1
AlphaBlendingEn	0	1	1
AlphaSmoothEn	0	0/1	0/1
TempAlphaRestrictEn	0	0	1
AlphaClipEn	0	1	1
BFRangeEn	1	1	1
BFDomainEn	1	1	1

7.3.2.8.3 16-bit Pixel Format Scale

Scaling support in CV pipeline requires 16-bit data format support throughout VIC scaling pipeline. VIC is used to crop the ROI from the original wide camera to get the DLA network input resolution. For example:

960x540@540fps-Y16 -> VIC -> 224x224@540-Y16 -> DLA (downscale)

192x108@1080fps-Y16 -> VIC -> 224x224@1080-Y16 -> DLA (upscale)

Table 7.72 Supported Pixel Format

Enumerant	Value
T_R16	112
T_V16U16	120
T_Y16___V16U16_N444	116
T_Y16___V16U16_N422	117
T_Y16___V16U16_N420	118
T_A16R16G16B16	110
T_A16Y16U16V16	111

Notes: Output format is always the same as input format

The 16-bit scaler can not directly support the input formats such as T_Y16___V16U16_N420, etc., but it can be supported in two pass mode; one pass is for T_L16, the second pass is T_V16U16.

Table 7.73 Hardware Supported Pixel Format

Enumerant	Value
T_L16	109
T_V16U16	120
T_A16Y16U16V16	111

7.3.2.8.4 16-bit Scaler Architecture

The 16-bit scaler uses legacy VIC's Fetch Control for source image over-fetching from surface cache, and to calculate filter coefficients for the scaler. Y_scaler and X_scaler should be extended to support 16-bit input. The output image is written into the external memory via out_gen unit.

Several constraints are listed as follows:

- Hardware only supports T_L16, T_V16U16, T_A16Y16U16V16 three pixel format, it needs software to map other 16-bit pixel formats into them via multiple pass.
- 16-bit scaler only support one slot configure.
- 16-bit scaler don't support flip and rotation operation.
- 16-bit scaler don't support sub-frame processing.
- 16-bit scaler only support maxim16kx8k and 8kx16k resolution for T_A16Y16U16V16 pixel format.
- 16-bit scaler's DestRectTop and DestRectLeft should be both zeros.

Table 7.74 SlotConfig

Name	Data Type	Offset	Description
Legacy fields	fixed<0,448,0> (448 bits)	0	Legacy fields for VIC4.1
B16ScalerEnable	fixed<0,1,0> (1 bits)	448	Enable bit for new 16-bit scaler mode
reserved22	fixed<0,31,0> (31 bits)	449	
reserved23	fixed<0,32,0> (32 bits)	480	

7.3.3 VIC Registers

7.3.3.1 THI Registers

Refer to Reading Register Tables in the Introduction section for the register table protocol as well as recommendations for accessing registers.

NV_PVIC_THI_INCR_SYNCPT_0

Offset: 0x0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
17:10	0x0	**COND**: **0** = IMMEDIATE **1** = OP_DONE **2** = RD_DONE **3** = REG_WR_SAFE **4** = ENGINE_IDLE
9:0	0x0	**INDX**: **0** = INIT

NV_PVIC_THI_INCR_SYNCPT_CTRL_0

Offset: 0x4
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,0000,0000,xxxx,xxx0,xxxx,xxx0)

Bit	Reset	Description
25	0x0	**NO_STALL_4**: **0** = INIT
24	0x0	**SOFT_RESET_4**: **0** = INIT
23	0x0	**NO_STALL_3**: **0** = INIT
22	0x0	**SOFT_RESET_3**: **0** = INIT
21	0x0	**NO_STALL_2**: **0** = INIT
20	0x0	**SOFT_RESET_2**: **0** = INIT
19	0x0	**NO_STALL_1**: **0** = INIT
18	0x0	**SOFT_RESET_1**: **0** = INIT
17	0x0	**NO_STALL_0**: **0** = INIT
16	0x0	**SOFT_RESET_0**: **0** = INIT
8	0x0	**NO_STALL**: **0** = INIT
0	0x0	**SOFT_RESET**: **0** = INIT

NV_PVIC_THI_INCR_SYNCPT_ERR_0

Offset: 0x8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**COND_STS_ENGINE_IDLE**: **0** = INIT **1** = CLEAR

Bit	Reset	Description
3	0x0	**COND_STS_REG_WR_SAFE**: **0** = INIT **1** = CLEAR
2	0x0	**COND_STS_RD_DONE**: **0** = INIT **1** = CLEAR
1	0x0	**COND_STS_OPDONE**: **0** = INIT **1** = CLEAR
0	0x0	**COND_STS_IMM**: **0** = INIT **1** = CLEAR

NV_PVIC_THI_CTXSW_INCR_SYNCPT_0

Offset: 0xc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9:0	0x0	**INDX**: **0** = INIT

NV_PVIC_THI_CTXSW_0

Offset: 0x20
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0001fc00 (0bxxxx,xxxx,xxx0,0001,1111,1100,0000,0000)

Bit	R/W	Reset	Description
20:11	RW	0x3f	**CURR_CHANNEL**: **63** = INIT
10	RO	0x1	**AUTO_ACK**: **1** = INIT

Bit	R/W	Reset	Description
9:0	RW	0x0	**CURR_CLASS**: **0** = INIT

NV_PVIC_THI_CTXSW_NEXT_0

Offset: 0x24
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
19:10	X	**NEXT_CHANNEL**
9:0	X	**NEXT_CLASS**: **0** = INIT

NV_PVIC_THI_CONT_SYNCPT_EOF_0

Offset: 0x28
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,x000,0000,0000)

Bit	Reset	Description
10	0x0	**COND**: **0** = INIT
9:0	0x0	**INDEX**: **0** = INIT

NV_PVIC_THI_CONT_SYNCPT_L1_0

Offset: 0x2c
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,x000,0000,0000)

Bit	Reset	Description
10	0x0	**COND**: **0** = INIT
9:0	0x0	**INDEX**: **0** = INIT

NV_PVIC_THI_STREAMID0_0

Offset: 0x30
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
6:0	X	**ID**

NV_PVIC_THI_STREAMID1_0

Offset: 0x34
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
6:0	X	**ID**

NV_PVIC_THI_THI_SEC_0

Offset: 0x38
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000100 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx1,xxx0,xxx0)

Bit	Reset	Description
8	0x1	**CH_LOCK**: **0** = FALSE **1** = TRUE
4	0x0	**TZ_AUTH**: **0** = FALSE **1** = TRUE
0	0x0	**TZ_LOCK**: **0** = FALSE **1** = TRUE

NV_PVIC_THI_METHOD0_0

Offset: 0x40
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0000,0000)

Bit	Reset	Description
11:0	0x0	**OFFSET**: **0** = INIT

NV_PVIC_THI_METHOD1_0

Offset: 0x44
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**: **0** = INIT

NV_PVIC_THI_CONTEXT_SWITCH_0

Offset: 0x60
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b00xx,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:30	0x0	**TARGET**: **0** = INIT **0** = LOCAL_FB **1** = SYSMEM_COH **2** = SYSMEM_NONCOH
27:0	0x0	**PTR**: **0** = INIT

NV_PVIC_THI_INT_STATUS_0

Offset: 0x78
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**FALCON_INT**: **0** = INIT **1** = CLEAR

NV_PVIC_THI_INT_MASK_0

Offset: 0x7c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	0x1	**FALCON_INT**: **1** = INIT

NV_PVIC_THI_CONFIG0_0

Offset: 0x80
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,xxx0)

Bit	Reset	Description
4	0x0	**IDLE_SYNCPT_INC_ENG**: **0** = FALSE **1** = TRUE
0	0x0	**RETURN_SYNCPT_ON_ERR**: **0** = INIT

NV_PVIC_THI_DBG_MISC_0

Offset: 0x84
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
3	RW	X	**THI_IDLE_EN**: **0** = CLEAR **1** = SET
2	RO	X	**THI_SYNCPT_PENDING_STATUS**: **0** = CLEAR **1** = SET
1	RO	X	**THI_IDLE_STATUS**: // For checking status first enable the bit 3 in DBG_MISC register **0** = CLEAR **1** = SET
0	RO	X	**CLIENT_IDLE_STATUS**: **0** = CLEAR **1** = SET

NV_PVIC_THI_SLCG_OVERRIDE_HIGH_A_0

Offset: 0x88
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000000ff (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1111,1111)

Bit	Reset	Description
7:0	0xff	**REG**: **255** = INIT

NV_PVIC_THI_SLCG_OVERRIDE_LOW_A_0

Offset: 0x8c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**REG**: **0** = INIT

NV_PVIC_THI_CLK_OVERRIDE_0

Offset: 0xe00
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CYA**: **0** = INIT

7.3.3.2 FALCON Registers

NV_PVIC_FALCON_IRQSSET_0

Offset: 0x1000
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
16	X	**DMA**: **1** = SET
15:8	X	**EXT**
7	X	**SWGEN1**: **1** = SET
6	X	**SWGEN0**: **1** = SET
5	X	**EXTERR**: **1** = SET
4	X	**HALT**: **1** = SET
3	X	**CTXSW**: **1** = SET
2	X	**MTHD**: **1** = SET
1	X	**WDTMR**: **1** = SET
0	X	**GPTMR**: **1** = SET

NV_PVIC_FALCON_IRQSCLR_0

Offset: 0x1004
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
16	X	**DMA**: **1** = SET
15	X	**EXT_EXTIRQ8**: **1** = SET
14	X	**EXT_EXTIRQ7**: **1** = SET
13	X	**EXT_EXTIRQ6**: **1** = SET
12	X	**EXT_EXTIRQ5**: **1** = SET
11	X	**EXT_EXTIRQ4**: **1** = SET
10	X	**EXT_EXTIRQ3**: **1** = SET
9	X	**EXT_EXTIRQ2**: **1** = SET
8	X	**EXT_EXTIRQ1**: **1** = SET
7	X	**SWGEN1**: **1** = SET
6	X	**SWGEN0**: **1** = SET
5	X	**EXTERR**: **1** = SET
4	X	**HALT**: **1** = SET
3	X	**CTXSW**: **1** = SET
2	X	**MTHD**: **1** = SET
1	X	**WDTMR**: **1** = SET
0	X	**GPTMR**: **1** = SET

NV_PVIC_FALCON_IRQSTAT_0

Offset: 0x1008
Read/Write: RO

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**DMA**: **0** = FALSE **1** = TRUE
15	0x0	**EXT_EXTIRQ8**: **0** = FALSE **1** = TRUE
14	0x0	**EXT_EXTIRQ7**: **0** = FALSE **1** = TRUE
13	0x0	**EXT_EXTIRQ6**: **0** = FALSE **1** = TRUE
12	0x0	**EXT_EXTIRQ5**: **0** = FALSE **1** = TRUE
11	0x0	**EXT_EXTIRQ4**: **0** = FALSE **1** = TRUE
10	0x0	**EXT_EXTIRQ3**: **0** = FALSE **1** = TRUE
9	0x0	**EXT_EXTIRQ2**: **0** = FALSE **1** = TRUE
8	0x0	**EXT_EXTIRQ1**: **0** = FALSE **1** = TRUE
7	0x0	**SWGEN1**: **0** = FALSE **1** = TRUE
6	0x0	**SWGEN0**: **0** = FALSE **1** = TRUE
5	0x0	**EXTERR**: **0** = FALSE **1** = TRUE
4	0x0	**HALT**: **0** = FALSE **1** = TRUE
3	0x0	**CTXSW**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
2	0x0	**MTHD**: **0** = FALSE **1** = TRUE
1	0x0	**WDTMR**: **0** = FALSE **1** = TRUE
0	0x0	**GPTMR**: **0** = FALSE **1** = TRUE

NV_PVIC_FALCON_IRQMODE_0

Offset: 0x100c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000fc24 (0bxxxx,xxxx,xxxx,xxx0,1111,1100,0010,0100)

Bit	Reset	Description
16	0x0	**LVL_DMA**: **0** = FALSE **0** = INIT **1** = TRUE
15:8	0xfc	**LVL_EXT**: **252** = INIT
7	0x0	**LVL_SWGEN1**: **0** = FALSE **0** = INIT **1** = TRUE
6	0x0	**LVL_SWGEN0**: **0** = FALSE **0** = INIT **1** = TRUE
5	0x1	**LVL_EXTERR**: **0** = FALSE **1** = INIT **1** = TRUE
4	0x0	**LVL_HALT**: **0** = FALSE **0** = INIT **1** = TRUE
3	0x0	**LVL_CTXSW**: **0** = FALSE **0** = INIT **1** = TRUE

Bit	Reset	Description
2	0x1	**LVL_MTHD**: **0** = FALSE **1** = INIT **1** = TRUE
1	0x0	**LVL_WDTMR**: **0** = FALSE **0** = INIT **1** = TRUE
0	0x0	**LVL_GPTMR**: **0** = FALSE **0** = INIT **1** = TRUE

NV_PVIC_FALCON_IRQMSET_0

Offset: 0x1010
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
16	X	**DMA**: **1** = SET
15	X	**EXT_EXTIRQ8**: **1** = SET
14	X	**EXT_EXTIRQ7**: **1** = SET
13	X	**EXT_EXTIRQ6**: **1** = SET
12	X	**EXT_EXTIRQ5**: **1** = SET
11	X	**EXT_EXTIRQ4**: **1** = SET
10	X	**EXT_EXTIRQ3**: **1** = SET
9	X	**EXT_EXTIRQ2**: **1** = SET
8	X	**EXT_EXTIRQ1**: **1** = SET

Bit	Reset	Description
7	X	**SWGEN1**: **1** = SET
6	X	**SWGEN0**: **1** = SET
5	X	**EXTERR**: **1** = SET
4	X	**HALT**: **1** = SET
3	X	**CTXSW**: **1** = SET
2	X	**MTHD**: **1** = SET
1	X	**WDTMR**: **1** = SET
0	X	**GPTMR**: **1** = SET

NV_PVIC_FALCON_IRQMCLR_0

Offset: 0x1014
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
16	X	**DMA**: **1** = SET
15	X	**EXT_EXTIRQ8**: **1** = SET
14	X	**EXT_EXTIRQ7**: **1** = SET
13	X	**EXT_EXTIRQ6**: **1** = SET
12	X	**EXT_EXTIRQ5**: **1** = SET
11	X	**EXT_EXTIRQ4**: **1** = SET

Bit	Reset	Description
10	X	**EXT_EXTIRQ3**: **1** = SET
9	X	**EXT_EXTIRQ2**: **1** = SET
8	X	**EXT_EXTIRQ1**: **1** = SET
7	X	**SWGEN1**: **1** = SET
6	X	**SWGEN0**: **1** = SET
5	X	**EXTERR**: **1** = SET
4	X	**HALT**: **1** = SET
3	X	**CTXSW**: **1** = SET
2	X	**MTHD**: **1** = SET
1	X	**WDTMR**: **1** = SET
0	X	**GPTMR**: **1** = SET

NV_PVIC_FALCON_IRQMASK_0

Offset: 0x1018
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**DMA**: **0** = DISABLE **1** = ENABLE
15	0x0	**EXT_EXTIRQ8**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
14	0x0	**EXT_EXTIRQ7**: **0** = DISABLE **1** = ENABLE
13	0x0	**EXT_EXTIRQ6**: **0** = DISABLE **1** = ENABLE
12	0x0	**EXT_EXTIRQ5**: **0** = DISABLE **1** = ENABLE
11	0x0	**EXT_EXTIRQ4**: **0** = DISABLE **1** = ENABLE
10	0x0	**EXT_EXTIRQ3**: **0** = DISABLE **1** = ENABLE
9	0x0	**EXT_EXTIRQ2**: **0** = DISABLE **1** = ENABLE
8	0x0	**EXT_EXTIRQ1**: **0** = DISABLE **1** = ENABLE
7	0x0	**SWGEN1**: **0** = DISABLE **1** = ENABLE
6	0x0	**SWGEN0**: **0** = DISABLE **1** = ENABLE
5	0x0	**EXTERR**: **0** = DISABLE **1** = ENABLE
4	0x0	**HALT**: **0** = DISABLE **1** = ENABLE
3	0x0	**CTXSW**: **0** = DISABLE **1** = ENABLE
2	0x0	**MTHD**: **0** = DISABLE **1** = ENABLE
1	0x0	**WDTMR**: **0** = DISABLE **1** = ENABLE
0	0x0	**GPTMR**: **0** = DISABLE **1** = ENABLE

NV_PVIC_FALCON_IRQDEST_0

Offset: 0x101c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**TARGET_EXT_EXTIRQ8**: **0** = FALCON_IRQ0 **0** = HOST_NORMAL **0** = INIT **1** = FALCON_IRQ1 **1** = HOST_NONSTALL
30	0x0	**TARGET_EXT_EXTIRQ7**: **0** = FALCON_IRQ0 **0** = HOST_NORMAL **0** = INIT **1** = FALCON_IRQ1 **1** = HOST_NONSTALL
29	0x0	**TARGET_EXT_EXTIRQ6**: **0** = FALCON_IRQ0 **0** = HOST_NORMAL **0** = INIT **1** = FALCON_IRQ1 **1** = HOST_NONSTALL
28	0x0	**TARGET_EXT_EXTIRQ5**: **0** = FALCON_IRQ0 **0** = HOST_NORMAL **0** = INIT **1** = FALCON_IRQ1 **1** = HOST_NONSTALL
27	0x0	**TARGET_EXT_EXTIRQ4**: **0** = FALCON_IRQ0 **0** = HOST_NORMAL **0** = INIT **1** = FALCON_IRQ1 **1** = HOST_NONSTALL
26	0x0	**TARGET_EXT_EXTIRQ3**: **0** = FALCON_IRQ0 **0** = HOST_NORMAL **0** = INIT **1** = FALCON_IRQ1 **1** = HOST_NONSTALL
25	0x0	**TARGET_EXT_EXTIRQ2**: **0** = FALCON_IRQ0 **0** = HOST_NORMAL **0** = INIT **1** = FALCON_IRQ1 **1** = HOST_NONSTALL

Bit	Reset	Description
24	0x0	**TARGET_EXT_EXTIRQ1**: **0** = FALCON_IRQ0 **0** = HOST_NORMAL **0** = INIT **1** = FALCON_IRQ1 **1** = HOST_NONSTALL
23	0x0	**TARGET_SWGEN1**: **0** = FALCON_IRQ0 **0** = HOST_NORMAL **0** = INIT **1** = FALCON_IRQ1 **1** = HOST_NONSTALL
22	0x0	**TARGET_SWGEN0**: **0** = FALCON_IRQ0 **0** = HOST_NORMAL **0** = INIT **1** = FALCON_IRQ1 **1** = HOST_NONSTALL
21	0x0	**TARGET_EXTERR**: **0** = FALCON_IRQ0 **0** = HOST_NORMAL **0** = INIT **1** = FALCON_IRQ1 **1** = HOST_NONSTALL
20	0x0	**TARGET_HALT**: **0** = FALCON_IRQ0 **0** = HOST_NORMAL **0** = INIT **1** = FALCON_IRQ1 **1** = HOST_NONSTALL
19	0x0	**TARGET_CTXSW**: **0** = FALCON_IRQ0 **0** = HOST_NORMAL **0** = INIT **1** = FALCON_IRQ1 **1** = HOST_NONSTALL
18	0x0	**TARGET_MTHD**: **0** = FALCON_IRQ0 **0** = HOST_NORMAL **0** = INIT **1** = FALCON_IRQ1 **1** = HOST_NONSTALL
17	0x0	**TARGET_WDTMR**: **0** = FALCON_IRQ0 **0** = HOST_NORMAL **0** = INIT **1** = FALCON_IRQ1 **1** = HOST_NONSTALL
16	0x0	**TARGET_GPTMR**: **0** = FALCON_IRQ0 **0** = HOST_NORMAL **0** = INIT **1** = FALCON_IRQ1 **1** = HOST_NONSTALL

Bit	Reset	Description
15	0x0	**HOST_EXT_EXTIRQ8**: **0** = FALCON **1** = HOST
14	0x0	**HOST_EXT_EXTIRQ7**: **0** = FALCON **1** = HOST
13	0x0	**HOST_EXT_EXTIRQ6**: **0** = FALCON **1** = HOST
12	0x0	**HOST_EXT_EXTIRQ5**: **0** = FALCON **1** = HOST
11	0x0	**HOST_EXT_EXTIRQ4**: **0** = FALCON **1** = HOST
10	0x0	**HOST_EXT_EXTIRQ3**: **0** = FALCON **1** = HOST
9	0x0	**HOST_EXT_EXTIRQ2**: **0** = FALCON **1** = HOST
8	0x0	**HOST_EXT_EXTIRQ1**: **0** = FALCON **1** = HOST
7	0x0	**HOST_SWGEN1**: **0** = FALCON **0** = INIT **1** = HOST
6	0x0	**HOST_SWGEN0**: **0** = FALCON **0** = INIT **1** = HOST
5	0x0	**HOST_EXTERR**: **0** = FALCON **0** = INIT **1** = HOST
4	0x0	**HOST_HALT**: **0** = FALCON **0** = INIT **1** = HOST
3	0x0	**HOST_CTXSW**: **0** = FALCON **0** = INIT **1** = HOST
2	0x0	**HOST_MTHD**: **0** = FALCON **0** = INIT **1** = HOST

Bit	Reset	Description
1	0x0	**HOST_WDTMR**: **0** = FALCON **0** = INIT **1** = HOST
0	0x0	**HOST_GPTMR**: **0** = FALCON **0** = INIT **1** = HOST

NV_PVIC_FALCON_GPTMRINT_0

Offset: 0x1020
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VAL**: **0** = INIT

NV_PVIC_FALCON_GPTMRVAL_0

Offset: 0x1024
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VAL**: **0** = INIT

NV_PVIC_FALCON_GPTMRCTL_0

Offset: 0x1028
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**GPTMR_SRC_MODE**: **0** = ENGCLK **0** = INIT **1** = PTIMER
0	0x0	**GPTMREN**: **0** = DISABLE **0** = INIT **1** = ENABLE

NV_PVIC_FALCON_PTIMER0_0

Offset: 0x102c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VAL**: **0** = INIT

NV_PVIC_FALCON_PTIMER1_0

Offset: 0x1030
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VAL**: **0** = INIT

NV_PVIC_FALCON_WDTMRVAL_0

Offset: 0x1034
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VAL**: **0** = INIT

NV_PVIC_FALCON_WDTMRCTL_0

Offset: 0x1038
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**WDTMR_SRC_MODE**: **0** = ENGCLK **0** = INIT **1** = PTIMER
0	0x0	**WDTMREN**: **0** = DISABLE **0** = INIT **1** = ENABLE

NV_PVIC_FALCON_IRQDEST2_0

Offset: 0x103c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
16	0x0	**TARGET_DMA**: **0** = FALCON_IRQ0 **0** = HOST_NORMAL **0** = INIT **1** = FALCON_IRQ1 **1** = HOST_NONSTALL
0	0x0	**HOST_DMA**: **0** = FALCON **0** = INIT **1** = HOST

NV_PVIC_FALCON_MAILBOX0_0

Offset: 0x1040
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**: **0** = INIT

NV_PVIC_FALCON_MAILBOX1_0

Offset: 0x1044
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**: **0** = INIT

NV_PVIC_FALCON_ITFEN_0

Offset: 0x1048
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000004 (0bxxxx,xxxx,xxxx,xxxx,xxx0,xxx0,xxxx,x100)

Bit	R/W	Reset	Description
12	RO	0x0	**HIRQ_NONSTALL_EDGE**: **0** = FALSE **1** = TRUE
8	RW	0x0	**CTXSW_NACK**: **0** = FALSE **1** = TRUE
2	RW	0x1	**PRIV_POSTWR**: **0** = FALSE **1** = INIT **1** = TRUE
1	RW	0x0	**MTHDEN**: **0** = DISABLE **0** = INIT **1** = ENABLE
0	RW	0x0	**CTXEN**: **0** = DISABLE **0** = INIT **1** = ENABLE

NV_PVIC_FALCON_IDLESTATE_0

Offset: 0x104c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000XXXX (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
17:16	RW	0x0	**ENGINE_BUSY_CYA**: **0** = HW **1** = RESERVED **2** = SW_BUSY **3** = SW_IDLE
15:1	RO	X	**EXT_BUSY**
0	RO	X	**FALCON_BUSY**: **0** = FALSE **1** = TRUE

NV_PVIC_FALCON_CURCTX_0

Offset: 0x1050
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
30	0x0	**CTXVLD**: **0** = FALSE **0** = INIT **1** = TRUE
29:28	0x0	**CTXTGT**: **0** = INIT **0** = LOCAL_FB **2** = COHERENT_SYSMEM **3** = NONCOHERENT_SYSMEM
27:0	0x0	**CTXPTR**: **0** = INIT

NV_PVIC_FALCON_NXTCTX_0

Offset: 0x1054
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
30	0x0	**CTXVLD**: **0** = FALSE **0** = INIT **1** = TRUE
29:28	0x0	**CTXTGT**: **0** = INIT **0** = LOCAL_FB **2** = COHERENT_SYSMEM **3** = NONCOHERENT_SYSMEM
27:0	0x0	**CTXPTR**: **0** = INIT

NV_PVIC_FALCON_CTXACK_0

Offset: 0x1058
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
1	X	**REST_ACK**: **0** = CLEAR **1** = SET
0	X	**SAVE_ACK**: **0** = CLEAR **1** = SET

NV_PVIC_FALCON_FHSTATE_0

Offset: 0x105c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
17	X	**STALL_REQ**: **0** = FALSE **1** = TRUE
16	X	**ENGINE_FAULTED**: **0** = FALSE **1** = TRUE
15:1	X	**EXT_HALTED**
0	X	**FALCON_HALTED**: **0** = FALSE **1** = TRUE

NV_PVIC_FALCON_PRIVSTATE_0

Offset: 0x1060
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**PRIV**: **0** = DISABLE **0** = INIT **1** = ENABLE

NV_PVIC_FALCON_MTHDDATA_0

Offset: 0x1064
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

NV_PVIC_FALCON_MTHDID_0

Offset: 0x1068
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
16	RO	X	**WPEND**: **0** = DONE **1** = PENDING
15	RW	X	**PRIV**: **0** = DISABLE **1** = ENABLE
14:12	RW	X	**SUBCH**

Bit	R/W	Reset	Description
11:0	RW	X	**ID**

NV_PVIC_FALCON_MTHDWDAT_0

Offset: 0x106c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

NV_PVIC_FALCON_MTHDCOUNT_0

Offset: 0x1070
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**COUNT**: **0** = INIT

NV_PVIC_FALCON_MTHDPOP_0

Offset: 0x1074
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
0	X	**POP**: **0** = FALSE **1** = TRUE

NV_PVIC_FALCON_MTHDRAMSZ_0

Offset: 0x1078
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**RAMSZ**: **0** = INIT

NV_PVIC_FALCON_SFTRESET_0

Offset: 0x107c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
0	X	**EXT**: **0** = FALSE **1** = TRUE

NV_PVIC_FALCON_OS_0

Offset: 0x1080
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VERSION**: **0** = INIT

NV_PVIC_FALCON_RM_0

Offset: 0x1084
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CONFIG**: **0** = INIT

NV_PVIC_FALCON_SOFT_PM_0

Offset: 0x1088
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
17	0x0	**TRIGGER_START**: **0** = INIT
16	0x0	**TRIGGER_END**: **0** = INIT
5:0	0x0	**PROBE**: **0** = INIT

NV_PVIC_FALCON_SOFT_MODE_0

Offset: 0x108c
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**PROBE**: **0** = INIT

NV_PVIC_FALCON_DEBUG1_0

Offset: 0x1090
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000040 (0bxxxx,xxxx,xxxx,x000,0000,0000,0100,0000)

Bit	Reset	Description
18	0x0	**CTXSW_MODE1**: **0** = DEFAULT **0** = INIT **1** = BYPASS_IDLE_CHECKS
17	0x0	**TRACE_FORMAT**: **0** = INIT **0** = UNCOMPRESSED **1** = COMPRESSED
16	0x0	**CTXSW_MODE**: **0** = INIT
15:0	0x40	**MTHD_DRAIN_TIME**: **64** = INIT

NV_PVIC_FALCON_DEBUGINFO_0

Offset: 0x1094
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

NV_PVIC_FALCON_IBRKPT1_0

Offset: 0x1098
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b000x,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: **0** = DISABLE **0** = INIT **1** = ENABLE
30	0x0	**SKIP**: **0** = DISABLE **0** = INIT **1** = ENABLE
29	0x0	**SUPPRESS**: **0** = DISABLE **0** = INIT **1** = ENABLE
23:0	0x0	**PC**: **0** = INIT

NV_PVIC_FALCON_IBRKPT2_0

Offset: 0x109c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b000x,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: **0** = DISABLE **0** = INIT **1** = ENABLE
30	0x0	**SKIP**: **0** = DISABLE **0** = INIT **1** = ENABLE

Bit	Reset	Description
29	0x0	**SUPPRESS**: **0** = DISABLE **0** = INIT **1** = ENABLE
23:0	0x0	**PC**: **0** = INIT

NV_PVIC_FALCON_CGCTL_0

Offset: 0x10a0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**CG_OVERRIDE**: **0** = INIT

NV_PVIC_FALCON_ENGCTL_0

Offset: 0x10a4
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000X0X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
9	RO	X	**STALLACK**: **0** = FALSE **1** = TRUE
8	RO	X	**STALLREQ**: **0** = FALSE **1** = TRUE
3	RW	X	**SWITCH_CONTEXT**: **0** = FALSE **1** = TRUE
2	RW	X	**CLR_STALLREQ**: **0** = FALSE **1** = TRUE

Bit	R/W	Reset	Description
1	RW	X	**SET_STALLREQ**: **0** = FALSE **1** = TRUE
0	RW	X	**INV_CONTEXT**: **0** = FALSE **1** = TRUE

NV_PVIC_FALCON_PMM_0

Offset: 0x10a8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,000x,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**TFBIF_STALL2_SEL**: **0** = RDATQ_FULL
27:24	0x0	**TFBIF_STALL1_SEL**: **0** = RDATQ_FULL
23:20	0x0	**TFBIF_STALL0_SEL**: **0** = RDATQ_FULL **1** = RACKQ_FULL **2** = WREQQ_FULL **3** = WDATQ_FULL **4** = WACKQ_FULL **5** = MREQQ_FULL **6** = RREQ_PEND **7** = WREQ_PEND **8** = RDATQ_FULL_SC **9** = RACKQ_FULL_SC **10** = WREQQ_FULL_SC **11** = WDATQ_FULL_SC **12** = WACKQ_FULL_SC **13** = MREQQ_FULL_SC **14** = RREQ_PEND_SC **15** = WREQ_PEND_SC
19:17	0x0	**TFBIF_DSTAT_SEL**: **0** = _1KTRANSFER **1** = RREQ **2** = WREQ **3** = TWREQ **4** = _1KTRANSFER_SC **5** = RREQ_SC **6** = WREQ_SC **7** = TWREQ_SC
15:12	0x0	**FALCON_SOFTPM1_SEL**: **0** = _0

Bit	Reset	Description
11:8	0x0	**FALCON_SOFTPM0_SEL**: **0** = _0 **1** = _1 **2** = _2 **3** = _3 **4** = _4 **5** = _5 **6** = _0_SC **7** = _1_SC **8** = _2_SC **9** = _3_SC **10** = _4_SC **11** = _5_SC
7:5	0x0	**FALCON_IDLE_SEL**: **0** = WAITING **1** = ENG_IDLE **2** = MTHD_FULL **3** = WAITING_SC **4** = ENG_IDLE_SC **5** = MTHD_FULL_SC
4:0	0x0	**FALCON_STALL_SEL**: **0** = ANY **1** = CODE **2** = DMAQ **3** = DMFENCE **4** = DMWAIT **5** = IMWAIT **6** = IPND **7** = LDSTQ **8** = SB **9** = ANY_SC **10** = CODE_SC **11** = DMAQ_SC **12** = DMFENCE_SC **13** = DMWAIT_SC **14** = IMWAIT_SC **15** = IPND_SC **16** = LDSTQ_SC **17** = SB_SC

NV_PVIC_FALCON_ADDR_0

Offset: 0x10ac
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0000,0000)

Bit	Reset	Description
11:6	0x0	**MSB**: **0** = INIT

Bit	Reset	Description
5:0	0x0	**LSB**: **0** = INIT

NV_PVIC_FALCON_IBRKPT3_0

Offset: 0x10b0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b000x,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: **0** = DISABLE **0** = INIT **1** = ENABLE
30	0x0	**SKIP**: **0** = DISABLE **0** = INIT **1** = ENABLE
29	0x0	**SUPPRESS**: **0** = DISABLE **0** = INIT **1** = ENABLE
23:0	0x0	**PC**: **0** = INIT

NV_PVIC_FALCON_IBRKPT4_0

Offset: 0x10b4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b000x,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: **0** = DISABLE **0** = INIT **1** = ENABLE

Bit	Reset	Description
30	0x0	**SKIP**: **0** = DISABLE **0** = INIT **1** = ENABLE
29	0x0	**SUPPRESS**: **0** = DISABLE **0** = INIT **1** = ENABLE
23:0	0x0	**PC**: **0** = INIT

NV_PVIC_FALCON_IBRKPT5_0

Offset: 0x10b8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b000x,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: **0** = DISABLE **0** = INIT **1** = ENABLE
30	0x0	**SKIP**: **0** = DISABLE **0** = INIT **1** = ENABLE
29	0x0	**SUPPRESS**: **0** = DISABLE **0** = INIT **1** = ENABLE
23:0	0x0	**PC**: **0** = INIT

NV_PVIC_FALCON_EXCI2_0

Offset: 0x10bc
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ADDRESS**: **0** = INIT

NV_PVIC_FALCON_EXCI_0

Offset: 0x10d0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:28	X	**EXPC_HIGH**
24:20	X	**EXCAUSE**: **0** = TRAP0 **1** = TRAP1 **2** = TRAP2 **3** = TRAP3 **8** = ILL_INS **9** = INV_INS **10** = MISS_INS **11** = DHIT_INS **13** = SP_OVERFLOW **15** = BRKPT_INS **16** = DMEM_MISS_INS **17** = DMEM_DHIT_INS **18** = DMEM_PAFAULT_INS **19** = DMEM_PERMISSION_INS **20** = DMEM_FAULT_INS
19:0	X	**EXPC**

NV_PVIC_FALCON_SVEC_SPR_0

Offset: 0x10d4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000X0000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
18	X	**SIGPASS**: **0** = FALSE **1** = TRUE

NV_PVIC_FALCON_RSTAT0_0

Offset: 0x10d8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXX0X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
30	X	**HALTED**
29	X	**WAITING**
28	X	**VALID**
25:23	X	**IRQ_FLUSH**
22	X	**EXC_FLUSH**
21	X	**AFILL_FLUSH**
20	X	**HALTSTOP_FLUSH**
16	X	**NO_INSTR_STALL**
12	X	**BL_STALL**
11	X	**SP_STALL**
10	X	**FLOW_STALL**
9	X	**SB_HIT_STALL**
8	X	**SB_FULL_STALL**
3	X	**DIV_STALL**
2	X	**FENCE_STALL**
1	X	**DMA_STALL**
0	X	**MEM_STALL**

NV_PVIC_FALCON_RSTAT3_0

Offset: 0x10dc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00XXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
23	X	**LDST_XT_BLOCK**
22	X	**LDST_XT_BUSY**
21	X	**DMA_RD_BUSY**
20	X	**DMA_WR_BUSY**
19	X	**DMA_RDQ_EMPTY**
18	X	**DMA_ACKQ_EMPTY**
17	X	**DMA_FBREQ_IDLE**
15	X	**CTXSW_PEND**
14:12	X	**CTXSW_STATE**
10	X	**CSWE**
8:6	X	**CSWIE**
5	X	**SBWB_EMPTY**
4	X	**LDST_IDLE**
3	X	**SCP_IDLE**
2	X	**DMA_IDLE**
1	X	**CTXSW_IDLE**
0	X	**MTHD_IDLE**

NV_PVIC_FALCON_IRQSCMASK_0

Offset: 0x10e0
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x0001ffff (0bxxxx,xxxx,xxxx,xxx1,1111,1111,1111,1111)

Bit	Reset	Description
16	0x1	**DMA**: **0** = DISABLE **1** = ENABLE
15:8	0xff	**EXT**: **0** = DISABLE **255** = ENABLE
7	0x1	**SWGEN1**: **0** = DISABLE **1** = ENABLE
6	0x1	**SWGEN0**: **0** = DISABLE **1** = ENABLE
5	0x1	**EXTERR**: **0** = DISABLE **1** = ENABLE
4	0x1	**HALT**: **0** = DISABLE **1** = ENABLE
3	0x1	**CTXSW**: **0** = DISABLE **1** = ENABLE
2	0x1	**MTHD**: **0** = DISABLE **1** = ENABLE
1	0x1	**WDTMR**: **0** = DISABLE **1** = ENABLE
0	0x1	**GPTMR**: **0** = DISABLE **1** = ENABLE

NV_PVIC_FALCON_HSCTL_0

Offset: 0x10e4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**SP_MIN**: **0** = DISABLE **1** = ENABLE
0	0x0	**TRACEPC**: **0** = DISABLE **1** = ENABLE

NV_PVIC_FALCON_HSCTL_PRIV_LEVEL_MASK_0

Offset: 0x10e8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xfffffb8f (0b1111,1111,1111,1111,1111,1011,1000,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **524288** = RESET_FUSE1 **1048575** = ALL_SOURCES_ENABLED **1048575** = RESET_FUSE0
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x0	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0x8	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_HWCFG2_0

Offset: 0x10f4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
16	X	**SCP**: **0** = DISABLE **1** = ENABLE
14	X	**BOOT_FROM_HS**: **0** = FALSE **1** = TRUE
13	X	**RISCV_BR_PRIV_LOCKDOWN**: **0** = UNLOCK **1** = LOCK
12	X	**MEM_SCRUBBING**: **0** = DONE **1** = PENDING
11	X	**RISCV_PL3_DISABLE**: **0** = FALSE **1** = TRUE
10	X	**RISCV**: **0** = DISABLE **1** = ENABLE
9	X	**SECUREBUS**: **0** = DISABLE **1** = ENABLE
8	X	**HS**: **0** = DISABLE **1** = ENABLE
7	X	**VHR**: **0** = DISABLE **1** = ENABLE
6	X	**STRAP_FUN**: **0** = DISABLE **1** = ENABLE
5	X	**HSCODE_REVOCATION**: **0** = DISABLE **1** = ENABLE
4	X	**KMEM**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
3	X	**DBGMODE**: **0** = DISABLE **1** = ENABLE
2	X	**PKCBOOT**: **0** = DISABLE **1** = ENABLE
1	X	**BMEM**: **0** = DISABLE **1** = ENABLE
0	X	**SHA**: **0** = DISABLE **1** = ENABLE

NV_PVIC_FALCON_CPUCTL_0

Offset: 0x1100
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x0xx,xxxx)

Bit	R/W	Reset	Description
6	RW	0x0	**ALIAS_EN**: **0** = FALSE **0** = INIT **1** = TRUE
5	RO	X	**STOPPED**: **0** = FALSE **1** = TRUE
4	RO	X	**HALTED**: **0** = FALSE **1** = TRUE
3	WO	X	**HRESET**: **0** = FALSE **1** = TRUE
2	WO	X	**SRESET**: **0** = FALSE **1** = TRUE
1	WO	X	**STARTCPU**: **0** = FALSE **1** = TRUE
0	WO	X	**IINVAL**: **0** = FALSE **1** = TRUE

NV_PVIC_FALCON_BOOTVEC_0

Offset: 0x1104
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VEC**: **0** = INIT

NV_PVIC_FALCON_HWCFG_0

Offset: 0x1108
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:27	X	**DMAQUEUE_DEPTH**
26:18	X	**METHODFIFO_DEPTH**
17:9	X	**DMEM_SIZE**
8:0	X	**IMEM_SIZE**

NV_PVIC_FALCON_DMACTL_0

Offset: 0x110c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00XX000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	R/W	Reset	Description
20:16	RO	X	**DMAQ_NUM**
2	RO	X	**IMEM_SCRUBBING**: **0** = DONE **1** = PENDING
1	RO	X	**DMEM_SCRUBBING**: **0** = DONE **1** = PENDING
0	RW	0x1	**REQUIRE_CTX**: **0** = FALSE **1** = INIT **1** = TRUE

NV_PVIC_FALCON_DMATRFBASE_0

Offset: 0x1110
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**BASE**: **0** = INIT

NV_PVIC_FALCON_DMATRFMOFFS_0

Offset: 0x1114
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:0	0x0	**OFFS**: **0** = INIT

NV_PVIC_FALCON_DMATRFCMD_0

Offset: 0x1118
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0X00XXXX (0bxxxx,0xxx,xxx0,xxx0,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
27	RW	0x0	**SET_DMLVL**: **0** = FALSE **0** = INIT **1** = TRUE
26:24	RW	X	**LVL**
20	RO	0x0	**ERROR**: **0** = FALSE **1** = TRUE
16	RW	0x0	**SET_DMTAG**: **0** = FALSE **0** = INIT **1** = TRUE
14:12	RW	X	**CTXDMA**
10:8	RW	X	**SIZE**: **0** = _4B **1** = _8B **2** = _16B **3** = _32B **4** = _64B **5** = _128B **6** = _256B
6	RW	X	**NOTIFY**: **0** = FALSE **1** = TRUE
5	RW	X	**WRITE**: **0** = FALSE **1** = TRUE
4	RW	X	**IMEM**: **0** = FALSE **1** = TRUE
1	RO	X	**IDLE**: **0** = FALSE **1** = TRUE
0	RO	X	**FULL**: **0** = FALSE **1** = TRUE

NV_PVIC_FALCON_DMATRFFBOFFS_0

Offset: 0x111c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**OFFS**: **0** = INIT

NV_PVIC_FALCON_DMAPOLL_FB_0

Offset: 0x1120
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXX001X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,xxxx)

Bit	R/W	Reset	Description
31:24	RO	X	**RCOUNT**
23:16	RO	X	**WCOUNT**
5	RW	0x0	**CFG_W_FENCE**: **0** = FALSE **0** = INIT **1** = TRUE
4	RW	0x1	**CFG_R_FENCE**: **0** = FALSE **1** = INIT **1** = TRUE
1	RO	X	**DMA_ACTIVE**: **0** = FALSE **1** = TRUE
0	RO	X	**FENCE_ACTIVE**: **0** = FALSE **1** = TRUE

NV_PVIC_FALCON_DMAPOLL_CP_0

Offset: 0x1124
Read/Write: See table below
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0xXXXX002X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,xxxx)

Bit	R/W	Reset	Description
31:24	RO	X	**RCOUNT**
23:16	RO	X	**WCOUNT**
5	RW	0x1	**CFG_W_FENCE**: **0** = FALSE **1** = INIT **1** = TRUE
4	RW	0x0	**CFG_R_FENCE**: **0** = FALSE **0** = INIT **1** = TRUE
1	RO	X	**DMA_ACTIVE**: **0** = FALSE **1** = TRUE
0	RO	X	**FENCE_ACTIVE**: **0** = FALSE **1** = TRUE

NV_PVIC_FALCON_HWCFG1_0

Offset: 0x112c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x90104126 (0b1001,0x00,0001,0000,0100,0001,0010,0110)

Bit	Reset	Description
31	0x1	**IMEM_AUTOFILL**: **0** = DISABLE **1** = ENABLE **1** = INIT
30	0x0	**DMEM_APERTURES**: **0** = DISABLE **0** = INIT **1** = ENABLE
29	0x0	**PRIV_DIRECT**: **0** = FALSE **0** = INIT **1** = TRUE

Bit	Reset	Description
28	0x1	**CSB_SIZE_16M**: **0** = FALSE **1** = INIT **1** = TRUE
27	0x0	**DBG_PRIV_BUS**: **0** = DISABLE **0** = INIT **1** = ENABLE
25:21	0x0	**DMEM_TAG_WIDTH**: **0** = INIT
20:16	0x10	**TAG_WIDTH**: **16** = INIT
15:12	0x4	**DMEM_PORTS**: **4** = INIT
11:8	0x1	**IMEM_PORTS**: **1** = INIT
7:6	0x0	**CORE_REV_SUBVERSION**: **0** = INIT **0** = _0 **1** = _1 **2** = _2 **3** = _3
5:4	0x2	**SECURITY_MODEL**: **0** = NONE **2** = INIT **2** = LIGHT **3** = HEAVY
3:0	0x6	**CORE_REV**: **1** = _1_0 **2** = _2_0 **3** = _3_0 **4** = _4_0 **5** = _5_0 **6** = INIT **6** = _6_0 **7** = _7_0

NV_PVIC_FALCON_CPUCTL_ALIAS_0

Offset: 0x1130
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
1	X	**STARTCPU**: **0** = FALSE **1** = TRUE

NV_PVIC_FALCON_CG2_0

Offset: 0x1134
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0003fffe (0bxxxx,xxxx,xxxx,xx11,1111,1111,1111,111x)

Bit	Reset	Description
17	0x1	**SLCG_FBIF**: **0** = ENABLED **1** = DISABLED
16	0x1	**SLCG_FALCON_TOP**: **0** = ENABLED **1** = DISABLED
15	0x1	**SLCG_FALCON_IRQSTAT**: **0** = ENABLED **1** = DISABLED
14	0x1	**SLCG_FALCON_WDTMR**: **0** = ENABLED **1** = DISABLED
13	0x1	**SLCG_FALCON_GPTMR**: **0** = ENABLED **1** = DISABLED
12	0x1	**SLCG_FALCON_TSYNC**: **0** = ENABLED **1** = DISABLED
11	0x1	**SLCG_FALCON_LDST**: **0** = ENABLED **1** = DISABLED
10	0x1	**SLCG_FALCON_MUL**: **0** = ENABLED **1** = DISABLED
9	0x1	**SLCG_FALCON_RF**: **0** = ENABLED **1** = DISABLED
8	0x1	**SLCG_FALCON_PMB**: **0** = ENABLED **1** = DISABLED

Bit	Reset	Description
7	0x1	**SLCG_FALCON_CTXSW**: **0** = ENABLED **1** = DISABLED
6	0x1	**SLCG_FALCON_CFG**: **0** = ENABLED **1** = DISABLED
5	0x1	**SLCG_FALCON_ICD**: **0** = ENABLED **1** = DISABLED
4	0x1	**SLCG_FALCON_DIV**: **0** = ENABLED **1** = DISABLED
3	0x1	**SLCG_FALCON_PIPE**: **0** = ENABLED **1** = DISABLED
2	0x1	**SLCG_FALCON_GC6_SR_FSM**: **0** = ENABLED **1** = DISABLED
1	0x1	**SLCG_FALCON_DMA**: **0** = ENABLED **1** = DISABLED

NV_PVIC_FALCON_STACKCFG_0

Offset: 0x1138
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00XXXXXX (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**SPEXC**: **0** = DISABLE **0** = INIT **1** = ENABLE
23:0	X	**BOTTOM**: **0** = INIT

NV_PVIC_FALCON_IMCTL_0

Offset: 0x1140
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
26:24	X	**CMD**: **0** = NOP **1** = IMINV **2** = IMBLK **3** = IMTAG **4** = IMTAG_SETVLD
23:0	X	**ADDR_BLK**

NV_PVIC_FALCON_IMSTAT_0

Offset: 0x1144
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**VAL**

NV_PVIC_FALCON_TRACEIDX_0

Offset: 0x1148
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00XX0000 (0b0000,0000,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**CNT**: **0** = INIT
23:16	RO	X	**MAXIDX**

Bit	R/W	Reset	Description
7:0	RW	0x0	**IDX**: **0** = INIT

NV_PVIC_FALCON_TRACEPC_0

Offset: 0x114c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00XXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
23:0	X	**PC**

NV_PVIC_FALCON_IMFILLRNG0_0

Offset: 0x1150
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**TAG_HI**: **0** = INIT
15:0	0x0	**TAG_LO**: **0** = INIT

NV_PVIC_FALCON_IMFILLRNG1_0

Offset: 0x1154
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**TAG_HI**: **0** = INIT
15:0	0x0	**TAG_LO**: **0** = INIT

NV_PVIC_FALCON_IMFILLCTL_0

Offset: 0x1158
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
8:0	0x0	**NBLOCKS**: **0** = INIT

NV_PVIC_FALCON_IMCTL_DEBUG_0

Offset: 0x115c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
26:24	X	**CMD**: **0** = NOP **2** = IMBLK **3** = IMTAG
23:0	X	**ADDR_BLK**

NV_PVIC_FALCON_TRACEINFO_0

Offset: 0x117c
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x0000XXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
15:0	X	**COUNT**

NV_PVIC_FALCON_IMEMC_0

Offset: 0x1180
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,xx00,0000,0000,0000,0000,0000,00xx)

Bit	R/W	Reset	Description
31	RO	0x0	**SEC_LOCK**: **0** = FALSE **1** = TRUE
30	RO	0x0	**SEC_WR_VIO**: **0** = FALSE **1** = TRUE
29	RO	0x0	**SEC_ATOMIC**: **0** = FALSE **1** = TRUE
28	RW	0x0	**SECURE**: **0** = INIT
25	RW	0x0	**AINCR**: **0** = FALSE **0** = INIT **1** = TRUE
24	RW	0x0	**AINCW**: **0** = FALSE **0** = INIT **1** = TRUE
23:8	RW	0x0	**BLK**: **0** = INIT
7:2	RW	0x0	**OFFS**: **0** = INIT

NV_PVIC_FALCON_IMEMD_0

Offset: 0x1184
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

NV_PVIC_FALCON_IMEMT_0

Offset: 0x1188
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000XXXX (0bxxxx,xxxx,xxxx,xx00,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
17	RO	0x0	**FAULT_UNCORRECTED**: **0** = INIT
16	RO	0x0	**FAULT_CORRECTED**: **0** = INIT
15:0	RW	X	**TAG**

NV_PVIC_FALCON_PMB_IMEM_PRIV_LEVEL_MASK_0

Offset: 0x118c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED

Bit	Reset	Description
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_DMA_PRIV_LEVEL_MASK_0

Offset: 0x1190
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR

Bit	Reset	Description
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_BOOTVEC_PRIV_LEVEL_MASK_0

Offset: 0x1194
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_AMAP_PRIV_LEVEL_MASK_0

Offset: 0x1198
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_TRACEBUF_PRIV_LEVEL_MASK_0

Offset: 0x119c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_TMR_PRIV_LEVEL_MASK_0

Offset: 0x11a0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR

Bit	Reset	Description
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_IRQSCMASK_PRIV_LEVEL_MASK_0

Offset: 0x11a4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_DBGCTL_PRIV_LEVEL_MASK_0

Offset: 0x11a8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xfffffbff (0b1111,1111,1111,1111,1111,1011,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **524288** = RESET_FUSE1 **1048575** = ALL_SOURCES_ENABLED **1048575** = RESET_FUSE0
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x0	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **8** = RESET_FUSE1 **15** = ALL_LEVELS_ENABLED **15** = RESET_FUSE0
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_SAFETY_CTRL_PRIV_LEVEL_MASK_0

Offset: 0x11b0
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_IMEM_DUMMY_0

Offset: 0x11b4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**: **0** = INIT

NV_PVIC_FALCON_PTIMER_PRIV_LEVEL_MASK_0

Offset: 0x11bc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_DMEMC_0

This is an array of four identical register entries; the register fields below apply to each entry.
Full register list is: NV_PVIC_FALCON_DMEMC_<i>, among which <i> belongs to <0..3>.
Offset: 0x11c0,..,0x11d8
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RO	0x0	**LVLERR**: **0** = FALSE **1** = TRUE
30	RO	0x0	**MULTIHIT**: **0** = FALSE **1** = TRUE
29	RO	0x0	**MISS**: **0** = FALSE **1** = TRUE
28	RW	0x0	**VA**: **0** = FALSE **0** = INIT **1** = TRUE
27	RW	0x0	**SETLVL**: **0** = FALSE **0** = INIT **1** = TRUE
26	RW	0x0	**SETTAG**: **0** = FALSE **0** = INIT **1** = TRUE
25	RW	0x0	**AINCR**: **0** = FALSE **0** = INIT **1** = TRUE
24	RW	0x0	**AINCW**: **0** = FALSE **0** = INIT **1** = TRUE
23:0	RW	0x0	**ADDRESS**: **0** = INIT

NV_PVIC_FALCON_DMEMD_0

This is an array of four identical register entries; the register fields below apply to each entry.
Full register list is: NV_PVIC_FALCON_DMEMD_<i>, among which <i> belongs to <0..3>.
Offset: 0x11c4,..,0x11dc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

NV_PVIC_FALCON_IDLESTATE_PRIV_LEVEL_MASK_0

Offset: 0x11e0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_ENG_STATE_PRIV_LEVEL_MASK_0

Offset: 0x11e4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_CG2_PRIV_LEVEL_MASK_0

Offset: 0x11e8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR

Bit	Reset	Description
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_PM_PRIV_LEVEL_MASK_0

Offset: 0x11ec
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_SP_MIN_PRIV_LEVEL_MASK_0

Offset: 0x11f0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_EXCI_PRIV_LEVEL_MASK_0

Offset: 0x11f4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_SVEC_SPR_PRIV_LEVEL_MASK_0

Offset: 0x11f8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR

Bit	Reset	Description
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_BRKPT_PRIV_LEVEL_MASK_0

Offset: 0x11fc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **524288** = RESET_FUSE1 **1048575** = ALL_SOURCES_ENABLED **1048575** = RESET_FUSE0
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **8** = RESET_FUSE1 **15** = ALL_LEVELS_ENABLED **15** = RESET_FUSE0

Bit	Reset	Description
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **8** = RESET_FUSE1 **15** = ALL_LEVELS_ENABLED **15** = RESET_FUSE0

NV_PVIC_FALCON_ICD_CMD_0

Offset: 0x1200
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xx0x,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31:27	RW	X	**EMASK_EXT**
26	RW	X	**EMASK_IV2**: **0** = FALSE **1** = TRUE
25	RW	X	**EMASK_IV1**: **0** = FALSE **1** = TRUE
24	RW	X	**EMASK_IV0**: **0** = FALSE **1** = TRUE
23	RW	X	**EMASK_EXC_IBREAK**: **0** = FALSE **1** = TRUE
22	RW	X	**EMASK_EXC_IMHIT**: **0** = FALSE **1** = TRUE
21	RW	X	**EMASK_EXC_IMISS**: **0** = FALSE **1** = TRUE
20	RW	X	**EMASK_EXC_UNIMP**: **0** = FALSE **1** = TRUE
19	RW	X	**EMASK_TRAP3**: **0** = FALSE **1** = TRUE
18	RW	X	**EMASK_TRAP2**: **0** = FALSE **1** = TRUE

Bit	R/W	Reset	Description
17	RW	X	**EMASK_TRAP1**: **0** = FALSE **1** = TRUE
16	RW	X	**EMASK_TRAP0**: **0** = FALSE **1** = TRUE
15	RO	X	**RDVLD**: **0** = FALSE **1** = TRUE
14	RO	X	**ERROR**: **0** = FALSE **1** = TRUE
13	RO	0x0	**FAULTED**: **0** = FALSE **0** = INIT **1** = TRUE
12:8	RW	X	**IDX**: **0** = REG0 **0** = RSTAT0 **0** = WB0 **1** = REG1 **1** = RSTAT1 **1** = WB1 **2** = REG2 **2** = RSTAT2 **2** = WB2 **3** = REG3 **3** = RSTAT3 **3** = WB3 **4** = REG4 **4** = RSTAT4 **5** = REG5 **5** = RSTAT5 **6** = REG6 **7** = REG7 **8** = REG8 **9** = REG9 **10** = REG10 **11** = REG11 **12** = REG12 **13** = REG13 **14** = REG14 **15** = REG15 **16** = IV0 **17** = IV1 **19** = EV **20** = SP **21** = PC **22** = IMB **23** = DMB **24** = CSW **25** = CCR **26** = SEC **27** = CTX **28** = EXCI **30** = IMB1 **30** = SEC1 **31** = DMB1

Bit	R/W	Reset	Description
7:6	RW	X	**SZ**: **0** = B **1** = HW **2** = W
3:0	RW	X	**OPC**: **0** = STOP **1** = RUN **2** = JRUN **3** = RUNB **4** = JRUNB **5** = STEP **6** = JSTEP **7** = EMASK **8** = RREG **9** = WREG **10** = RDM **11** = WDM **12** = RCM **13** = WCM **14** = RSTAT **15** = SBU

NV_PVIC_FALCON_ICD_ADDR_0

Offset: 0x1204
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ADDR**

NV_PVIC_FALCON_ICD_WDATA_0

Offset: 0x1208
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

NV_PVIC_FALCON_ICD_RDATA_0

Offset: 0x120c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

NV_PVIC_FALCON_DMEMT_0

This is an array of four identical register entries; the register fields below apply to each entry.
Full register list is: NV_PVIC_FALCON_DMEMT_<i>, among which <i> belongs to <0..3>.
Offset: 0x1210,..,0x121c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000XXXX (0bxxxx,xxxx,xxxx,xxx0,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
16	RO	0x0	**FAULTED**: **0** = INIT
15:0	WO	X	**TAG**

NV_PVIC_FALCON_ICD_PRIV_LEVEL_MASK_0

Offset: 0x1220
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **524288** = RESET_FUSE1 **1048575** = ALL_SOURCES_ENABLED **1048575** = RESET_FUSE0
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **8** = RESET_FUSE1 **15** = ALL_LEVELS_ENABLED **15** = RESET_FUSE0
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **8** = RESET_FUSE1 **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL **15** = RESET_FUSE0

NV_PVIC_FALCON_RSTAT0_PRIV_LEVEL_MASK_0

Offset: 0x1224
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED

Bit	Reset	Description
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_CSBERR_PRIV_LEVEL_MASK_0

Offset: 0x122c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR

Bit	Reset	Description
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_SCTL_0

Offset: 0x1240
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000300X (0bxxxx,xxxx,xxxx,xxxx,x011,xx00,xx00,xxx0)

Bit	R/W	Reset	Description
14	RW	0x0	**AUTH_EN**: **0** = FALSE **1** = TRUE
13	RW	0x1	**STALLREQ_CLR_EN**: **0** = FALSE **1** = TRUE
12	RW	0x1	**RESET_LVLM_EN**: **0** = FALSE **1** = TRUE
9:8	RW	0x0	**DEBUG_PRIV_LEVEL**: **0** = INIT
5:4	RW	0x0	**LSMODE_LEVEL**: **0** = INIT
1	RO	X	**HSMODE**: **0** = FALSE **1** = TRUE
0	RW	0x0	**LSMODE**: **0** = FALSE **1** = TRUE

NV_PVIC_FALCON_CSBERRSTAT_0

Offset: 0x1244
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x40XXXXXX (0b01xx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RW	0x0	**VALID**: **0** = FALSE **0** = INIT **1** = TRUE
30	RW	0x1	**ENABLE**: **0** = FALSE **1** = INIT **1** = TRUE
23:0	RO	X	**PC**

NV_PVIC_FALCON_CSBERR_INFO_0

Offset: 0x1248
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**VAL**

NV_PVIC_FALCON_CSBERR_ADDR_0

Offset: 0x124c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**VAL**

NV_PVIC_FALCON_SCTL1_0

Offset: 0x1250
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1111)

Bit	Reset	Description
3:2	0x3	**EXTLVL_MASK**: **3** = INIT
1:0	0x3	**CSBLVL_MASK**: **3** = INIT

NV_PVIC_FALCON_DBGCTL_0

Offset: 0x1254
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,xx00,xx00,0000,0000)

Bit	Reset	Description
31:16	0x0	**CYA**: **0** = INIT **35582** = ENABLE
13:12	0x0	**TRACE_MODE**: **0** = FULL **1** = REDUCED **2** = STACK
9	0x0	**ICD_CMDWL_RDM**: **0** = DISABLE **0** = INIT **1** = ENABLE
8	0x0	**ICD_CMDWL_RREG_GPR**: **0** = DISABLE **0** = INIT **1** = ENABLE
7	0x0	**PRIVWL_IBRKPT**: **0** = DISABLE **0** = INIT **1** = ENABLE

Bit	Reset	Description
6	0x0	**ICD_CMDWL_RSTAT**: **0** = DISABLE **0** = INIT **1** = ENABLE
5	0x0	**ICD_CMDWL_RREG_SPR**: **0** = DISABLE **0** = INIT **1** = ENABLE
4	0x0	**ICD_CMDWL_EMASK**: **0** = DISABLE **0** = INIT **1** = ENABLE
3	0x0	**ICD_CMDWL_STEP**: **0** = DISABLE **0** = INIT **1** = ENABLE
2	0x0	**ICD_CMDWL_RUNB**: **0** = DISABLE **0** = INIT **1** = ENABLE
1	0x0	**ICD_CMDWL_RUN**: **0** = DISABLE **0** = INIT **1** = ENABLE
0	0x0	**ICD_CMDWL_STOP**: **0** = DISABLE **0** = INIT **1** = ENABLE

NV_PVIC_FALCON_DMEM_DUMMY_0

Offset: 0x1258
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**: **0** = INIT

NV_PVIC_FALCON_DMCTL_0

Offset: 0x1260
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
26:24	X	**CMD**: **0** = NOP **1** = DMINV **2** = DMBLK **3** = DMTAG **4** = DMTAG_SETVLD **5** = DMCLEAN **6** = DMLVL
23:0	X	**ADDR_BLK**

NV_PVIC_FALCON_DMSTAT_0

Offset: 0x1264
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**VAL**

NV_PVIC_FALCON_DMVACTL_0

Offset: 0x1268
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000001c (0b0000,0000,0000,0000,0000,0000,0001,1100)

Bit	Reset	Description
31:0	0x1c	**BOUND**: **28** = INIT

NV_PVIC_FALCON_SP_MIN_0

Offset: 0x1270
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:0	0x0	**VALUE**: **0** = INIT

NV_PVIC_FALCON_HWCFG3_0

Offset: 0x1278
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXX0XXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27:16	X	**DMEM_TOTAL_SIZE**
11:0	X	**IMEM_TOTAL_SIZE**

NV_PVIC_FALCON_IMEM_PRIV_LEVEL_MASK_0

Offset: 0x1280
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED

Bit	Reset	Description
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_DMEM_PRIV_LEVEL_MASK_0

Offset: 0x1284
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR

Bit	Reset	Description
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_CPUCTL_PRIV_LEVEL_MASK_0

Offset: 0x1288
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_EXE_PRIV_LEVEL_MASK_0

Offset: 0x128c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_IRQTMR_PRIV_LEVEL_MASK_0

Offset: 0x1290
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_MTHDCTX_PRIV_LEVEL_MASK_0

Offset: 0x1294
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR

Bit	Reset	Description
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_SCTL_PRIV_LEVEL_MASK_0

Offset: 0x1298
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffff8f (0b1111,1111,1111,1111,1111,1111,1000,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = INIT
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0x8	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = INIT **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED

Bit	Reset	Description
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_WDTMR_PRIV_LEVEL_MASK_0

Offset: 0x129c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_DMEML_0

This is an array of four identical register entries; the register fields below apply to each entry.
Full register list is: NV_PVIC_FALCON_DMEML_<i>, among which <i> belongs to <0..3>.
Offset: 0x12a0,..,0x12ac
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
2:0	X	**LVL**

NV_PVIC_FALCON_PMB_DMEM_PRIV_LEVEL_MASK_0

This is an array of four identical register entries; the register fields below apply to each entry.
Full register list is: NV_PVIC_FALCON_PMB_DMEM_PRIV_LEVEL_MASK_<i>, among which <i> belongs to <0..3>.
Offset: 0x12b0,..,0x12bc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED

Bit	Reset	Description
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_DMAINFO_FINISHED_FBRD_LOW_0

Offset: 0x12c0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**VAL**

NV_PVIC_FALCON_DMAINFO_FINISHED_FBRD_HIGH_0

Offset: 0x12c4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	X	**OBIT**
30:0	X	**VAL**

NV_PVIC_FALCON_DMAINFO_FINISHED_FBWR_LOW_0

Offset: 0x12c8
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**VAL**

NV_PVIC_FALCON_DMAINFO_FINISHED_FBWR_HIGH_0

Offset: 0x12cc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	X	**OBIT**
30:0	X	**VAL**

NV_PVIC_FALCON_DMAINFO_CURRENT_FBRD_LOW_0

Offset: 0x12d0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**VAL**

NV_PVIC_FALCON_DMAINFO_CURRENT_FBRD_HIGH_0

Offset: 0x12d4
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	X	**OBIT**
30:0	X	**VAL**

NV_PVIC_FALCON_DMAINFO_CURRENT_FBWR_LOW_0

Offset: 0x12d8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**VAL**

NV_PVIC_FALCON_DMAINFO_CURRENT_FBWR_HIGH_0

Offset: 0x12dc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	X	**OBIT**
30:0	X	**VAL**

NV_PVIC_FALCON_DMAINFO_CTL_0

Offset: 0x12e0
Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
4	RO	X	**INTR_ERR_DMAREAD**: **0** = FALSE **1** = TRUE
3	RO	X	**INTR_ERR_DMATYPE**: **0** = NORMAL **1** = TAGGED
2	RW	X	**INTR_ERR_COMPLETION**: **0** = FALSE **1** = CLR **1** = TRUE
1	WO	X	**CLR_FBWR**: **0** = FALSE **1** = TRUE
0	WO	X	**CLR_FBRD**: **0** = FALSE **1** = TRUE

NV_PVIC_FALCON_DMAINFO_ERR_TAG_HIGH_0

Offset: 0x12e4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
30:0	X	**VAL**

NV_PVIC_FALCON_DMAINFO_ERR_TAG_LOW_0

Offset: 0x12e8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**VAL**

NV_PVIC_FALCON_SAFETY_ERB_0

Offset: 0x12ec
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**: **3942517760** = CORRECTED_ERR **3942522111** = UNCORRECTED_ERR

NV_PVIC_FALCON_SAFETY_MAILBOX_0

Offset: 0x12f0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**ERRCODE**: **0** = INIT **224** = SBE_IROM **225** = DBE_IROM **226** = PARITY_IMEM_EXT **227** = DMEM_EXT **240** = SBE_IMEM_PF **241** = DBE_IMEM_PF **241** = PARITY_IMEM_PF **242** = SBE_IMEM_SCP **243** = DBE_IMEM_SCP **243** = PARITY_IMEM_SCP **244** = SBE_IMEM_DMA **245** = DBE_IMEM_DMA **245** = PARITY_IMEM_DMA **246** = SBE_IMEM_PMB **247** = DBE_IMEM_PMB **247** = PARITY_IMEM_PMB **248** = DMEM_LDST **249** = DMEM_PMB **250** = DMEM_DMA **251** = DMEM_SCP **252** = CORRECTED_SHA **253** = UNCORRECTED_SHA **254** = CORRECTED_RISCV **255** = UNCORRECTED_RISCV
31:0	0x0	**DATA**: **0** = INIT
23:0	0x0	**ERRADDR**: **0** = INIT

NV_PVIC_FALCON_SAFETY_CTRL_0

Offset: 0x12f4
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000X0022 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx10,xx10)

Bit	R/W	Reset	Description
16	RO	X	**FUSE_OPT_ECC_EN**: **0** = FALSE **1** = TRUE
9	RW	0x0	**EXC_ON_DMEM_ERR**: **0** = FALSE **0** = INIT **1** = TRUE

Bit	R/W	Reset	Description
8	RW	0x0	**HALT_ON_IMEM_ERR**: **0** = FALSE **0** = INIT **1** = TRUE
5:4	RW	0x2	**DMEM_PARITY_DIS**: **1** = TRUE **2** = INIT
1:0	RW	0x2	**IMEM_ECC_DIS**: **1** = TRUE **2** = INIT

NV_PVIC_FALCON_SAFETY_ERR_ADDR_0

Offset: 0x12f8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VALUE**: **0** = INIT

NV_PVIC_FALCON_SAFETY_ERR_ADDR_HI_0

Offset: 0x12fc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VALUE**: **0** = INIT

NV_PVIC_FALCON_COMMON_SCRATCH_GROUP_0_0

This is an array of four identical register entries; the register fields below apply to each entry. Full register list is: NV_PVIC_FALCON_COMMON_SCRATCH_GROUP_0_<i>, among which <i> belongs to <0..3>.
Offset: 0x1300,..,0x130c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VAL**: **0** = INIT

NV_PVIC_FALCON_COMMON_SCRATCH_GROUP_1_0

This is an array of four identical register entries; the register fields below apply to each entry. Full register list is: NV_PVIC_FALCON_COMMON_SCRATCH_GROUP_1_<i>, among which <i> belongs to <0..3>.
Offset: 0x1310,..,0x131c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VAL**: **0** = INIT

NV_PVIC_FALCON_COMMON_SCRATCH_GROUP_2_0

This is an array of four identical register entries; the register fields below apply to each entry. Full register list is: NV_PVIC_FALCON_COMMON_SCRATCH_GROUP_2_<i>, among which <i> belongs to <0..3>.
Offset: 0x1320,..,0x132c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VAL**: **0** = INIT

NV_PVIC_FALCON_COMMON_SCRATCH_GROUP_3_0

This is an array of four identical register entries; the register fields below apply to each entry. Full register list is: NV_PVIC_FALCON_COMMON_SCRATCH_GROUP_3_<i>, among which <i> belongs to <0..3>.
Offset: 0x1330,..,0x133c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VAL**: **0** = INIT

NV_PVIC_FALCON_PRIVSTATE_PRIV_LEVEL_MASK_0

Offset: 0x13d0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR

Bit	Reset	Description
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_FALCON_PRGNVER_0

Offset: 0x13d4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000012 (0b0000,0000,0000,0000,0000,0000,0001,0010)

Bit	Reset	Description
31:18	0x0	**BUGFIX**: **0** = DEFT
17:4	0x1	**MINOR**: **1** = DEFT
3:0	0x2	**MAJOR**: **2** = DEFT

NV_PVIC_FALCON_COMMON_SCRATCH_GROUP_0_PRIV_LEVEL_MASK_0

Offset: 0x13f0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffcc (0b1111,1111,1111,1111,1111,1111,1100,1100)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED

Bit	Reset	Description
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xc	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **12** = DEFAULT_PRIV_LEVEL **12** = LEVEL2_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xc	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **12** = DEFAULT_PRIV_LEVEL **12** = LEVEL2_ENABLED **15** = ALL_LEVELS_ENABLED

NV_PVIC_FALCON_COMMON_SCRATCH_GROUP_1_PRIV_LEVEL_MASK_0

Offset: 0x13f4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffcc (0b1111,1111,1111,1111,1111,1111,1100,1100)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR

Bit	Reset	Description
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xc	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **12** = DEFAULT_PRIV_LEVEL **12** = LEVEL2_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xc	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **12** = DEFAULT_PRIV_LEVEL **12** = LEVEL2_ENABLED **15** = ALL_LEVELS_ENABLED

NV_PVIC_FALCON_COMMON_SCRATCH_GROUP_2_PRIV_LEVEL_MASK_0

Offset: 0x13f8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffcc (0b1111,1111,1111,1111,1111,1111,1100,1100)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xc	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **12** = DEFAULT_PRIV_LEVEL **12** = LEVEL2_ENABLED **15** = ALL_LEVELS_ENABLED

Bit	Reset	Description
3:0	0xc	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **12** = DEFAULT_PRIV_LEVEL **12** = LEVEL2_ENABLED **15** = ALL_LEVELS_ENABLED

NV_PVIC_FALCON_COMMON_SCRATCH_GROUP_3_PRIV_LEVEL_MASK_0

Offset: 0x13fc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffcc (0b1111,1111,1111,1111,1111,1111,1100,1100)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xc	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **12** = DEFAULT_PRIV_LEVEL **12** = LEVEL2_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xc	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **12** = DEFAULT_PRIV_LEVEL **12** = LEVEL2_ENABLED **15** = ALL_LEVELS_ENABLED

7.3.3.3 TFBIF Registers

NV_PVIC_TFBIF_CTL_0

Offset: 0x2000
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000X0XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,xxxx,xx1x)

Bit	R/W	Reset	Description
12	RO	X	**VPR**: **0** = FALSE **1** = TRUE
11:8	RW	0x0	**SRTOVAL**: **0** = INIT
7	RW	X	**CLR_SRTOUT**: **0** = CLEAR **1** = SET
6	RO	X	**SRTOUT**: **0** = FALSE **1** = TRUE
5	RO	X	**IDLEWDERR**: **0** = FALSE **1** = TRUE
4	RO	X	**IDLE**: **0** = FALSE **1** = TRUE
3	RW	X	**RESET**: **0** = CLEAR **1** = SET
2	RW	X	**CLR_IDLEWDERR**: **0** = CLEAR **1** = SET
1	RW	0x1	**ENABLE**: **0** = FALSE **1** = INIT **1** = TRUE
0	RW	X	**CLR_BWCOUNT**: **0** = CLEAR **1** = SET

NV_PVIC_TFBIF_MCCIF_FIFOCTRL_0

Offset: 0x2004
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
8	0x0	**WCLK_OVR_MODE**: **0** = DISABLE **0** = INIT **1** = ENABLE
7	0x0	**RCLK_OVR_MODE**: **0** = DISABLE **0** = INIT **1** = ENABLE
6	0x0	**CCLK_OVERRIDE**: **0** = DISABLE **0** = INIT **1** = ENABLE
5	0x0	**RDCL_RDFAST**: **0** = DISABLE **0** = INIT **1** = ENABLE
4	0x0	**WRMC_CLLE2X**: **0** = DISABLE **0** = INIT **1** = ENABLE
3	0x0	**RDMC_RDFAST**: **0** = DISABLE **0** = INIT **1** = ENABLE
2	0x0	**WRCL_MCLE2X**: **0** = DISABLE **0** = INIT **1** = ENABLE
1	0x0	**WCLK_OVERRIDE**: **0** = DISABLE **0** = INIT **1** = ENABLE
0	0x0	**RCLK_OVERRIDE**: **0** = DISABLE **0** = INIT **1** = ENABLE

NV_PVIC_TFBIF_THROTTLE_0

Offset: 0x2008
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x80000064 (0b10xx,0000,0000,0000,xxxx,0000,0110,0100)

Bit	Reset	Description
31:30	0x2	**LEAK_SIZE**: **0** = _16B **1** = _32B **2** = _64B **3** = _128B
27:16	0x0	**LEAK_COUNT**: **0** = DISABLE
11:0	0x64	**BUCKET_SIZE**: **100** = INIT

NV_PVIC_TFBIF_DBG_STAT0_0

Offset: 0x200c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00XXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
20	X	**UNWEIGHT_ACTMON_MCB**
19	X	**UNWEIGHT_ACTMON_ACTIVE**
17	X	**WU_IDLE**
16	X	**RU_IDLE**
15	X	**CSB_IDLE**
14	X	**WMCCIF_IDLE**
13	X	**RMCCIF_IDLE**
12	X	**ENGINE_IDLE**
11	X	**STALL_MREQ**
10	X	**STALL_WREQ_PENDING**
9	X	**STALL_RREQ_PENDING**

Bit	Reset	Description
8	X	**STALL_WACKQ**
7	X	**STALL_WDATQ**
6	X	**STALL_WREQQ**
5	X	**STALL_RACKQ**
4	X	**STALL_RDATQ**
3	X	**TAGQ_ISSUED**
2	X	**WREQ_ISSUED**
1	X	**RREQ_ISSUED**
0	X	**_1K_TRANSFER**

NV_PVIC_TFBIF_DBG_STAT1_0

Offset: 0x2010
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

NV_PVIC_TFBIF_DBG_RDCOUNT_LO_0

Offset: 0x2014
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

NV_PVIC_TFBIF_DBG_RDCOUNT_HI_0

Offset: 0x2018
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

NV_PVIC_TFBIF_DBG_WRCOUNT_LO_0

Offset: 0x201c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

NV_PVIC_TFBIF_DBG_WRCOUNT_HI_0

Offset: 0x2020
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

NV_PVIC_TFBIF_DBG_R32COUNT_0

Offset: 0x2024
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

NV_PVIC_TFBIF_DBG_R64COUNT_0

Offset: 0x2028
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

NV_PVIC_TFBIF_DBG_R128COUNT_0

Offset: 0x202c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

NV_PVIC_TFBIF_MCCIF_FIFOCTRL1_0

Offset: 0x2034
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**SWR2MC_REORDER_DEPTH_LIMIT**: **0** = INIT
15:0	0x0	**SRD2MC_REORDER_DEPTH_LIMIT**: **0** = INIT

NV_PVIC_TFBIF_WRR_RDP_0

Offset: 0x2038
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**INT_WEIGHT**: **0** = INIT
15:0	0x0	**EXT_WEIGHT**: **0** = INIT

NV_PVIC_TFBIF_ATT_PRIV_LEVEL_MASK_0

Offset: 0x2040
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:12	0xfffff	**SOURCE_ENABLE**: **1048575** = ALL_SOURCES_ENABLED
11	0x1	**SOURCE_WRITE_CONTROL**: **0** = LOWERED **1** = BLOCKED

Bit	Reset	Description
10	0x1	**SOURCE_READ_CONTROL**: **0** = LOWERED **1** = BLOCKED
9	0x1	**WRITE_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
8	0x1	**READ_VIOLATION**: **0** = SOLDIER_ON **1** = REPORT_ERROR
7:4	0xf	**WRITE_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED
3:0	0xf	**READ_PROTECTION**: **0** = ALL_LEVELS_DISABLED **8** = ONLY_LEVEL3_ENABLED **15** = ALL_LEVELS_ENABLED **15** = DEFAULT_PRIV_LEVEL

NV_PVIC_TFBIF_TRANSCFG_0

Offset: 0x2044
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxx00,xx00,xx00,xx00,xx00,xx00,xx00,xx00)

Bit	Reset	Description
29:28	0x0	**ATT7_SWID**: **0** = INIT
25:24	0x0	**ATT6_SWID**: **0** = INIT
21:20	0x0	**ATT5_SWID**: **0** = INIT
17:16	0x0	**ATT4_SWID**: **0** = INIT
13:12	0x0	**ATT3_SWID**: **0** = INIT
9:8	0x0	**ATT2_SWID**: **0** = INIT
5:4	0x0	**ATT1_SWID**: **0** = INIT

Bit	Reset	Description
1:0	0x0	**ATT0_SWID**: **0** = INIT

NV_PVIC_TFBIF_REGIONCFG_0

Offset: 0x2048
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,0xxx,0xxx,0xxx,0xxx,0xxx,0xxx,0xxx)

Bit	Reset	Description
31	0x0	**T7_VPR**: **0** = INIT
27	0x0	**T6_VPR**: **0** = INIT
23	0x0	**T5_VPR**: **0** = INIT
19	0x0	**T4_VPR**: **0** = INIT
15	0x0	**T3_VPR**: **0** = INIT
11	0x0	**T2_VPR**: **0** = INIT
7	0x0	**T1_VPR**: **0** = INIT
3	0x0	**T0_VPR**: **0** = INIT

NV_PVIC_TFBIF_ACTMON_ACTIVE_MASK_0

Offset: 0x204c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000006 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0110)

Bit	Reset	Description
3	0x0	**ACTIVE**: **0** = FALSE **0** = INIT **1** = TRUE
2	0x1	**DELAYED_MC**: **0** = FALSE **1** = INIT **1** = TRUE
1	0x1	**STALLED_MC**: **0** = FALSE **1** = INIT **1** = TRUE
0	0x0	**STARVED_MC**: **0** = FALSE **0** = INIT **1** = TRUE

NV_PVIC_TFBIF_ACTMON_ACTIVE_BORPS_0

Offset: 0x2050
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000083 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1000,0011)

Bit	Reset	Description
7	0x1	**ACTIVE_OPERATION**: **0** = OR **1** = AND **1** = INIT
6	0x0	**ACTIVE_POLARITY**: **0** = INIT **0** = POSITIVE **1** = NEGATIVE
5	0x0	**DELAYED_MC_OPERATION**: **0** = INIT **0** = OR **1** = AND
4	0x0	**DELAYED_MC_POLARITY**: **0** = INIT **0** = POSITIVE **1** = NEGATIVE
3	0x0	**STALLED_MC_OPERATION**: **0** = INIT **0** = OR **1** = AND

Bit	Reset	Description
2	0x0	**STALLED_MC_POLARITY**: **0** = INIT **0** = POSITIVE **1** = NEGATIVE
1	0x1	**STARVED_MC_OPERATION**: **0** = OR **1** = AND **1** = INIT
0	0x1	**STARVED_MC_POLARITY**: **0** = POSITIVE **1** = INIT **1** = NEGATIVE

NV_PVIC_TFBIF_ACTMON_ACTIVE_WEIGHT_0

Offset: 0x2054
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000001 (0b0000,0000,0000,0000,0000,0000,0000,0001)

Bit	Reset	Description
31:0	0x1	**VAL**: **1** = INIT

NV_PVIC_TFBIF_REGIONCFG1_0

Offset: 0x2058
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxx0,0000,xxx0,0000,xxx0,0000,xxx0,0000)

Bit	Reset	Description
28:24	0x0	**T3_APERT_ID**: **0** = INIT
20:16	0x0	**T2_APERT_ID**: **0** = INIT
12:8	0x0	**T1_APERT_ID**: **0** = INIT

Bit	Reset	Description
4:0	0x0	**T0_APERT_ID**: **0** = INIT

NV_PVIC_TFBIF_REGIONCFG2_0

Offset: 0x205c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxx0,0000,xxx0,0000,xxx0,0000,xxx0,0000)

Bit	Reset	Description
28:24	0x0	**T7_APERT_ID**: **0** = INIT
20:16	0x0	**T6_APERT_ID**: **0** = INIT
12:8	0x0	**T5_APERT_ID**: **0** = INIT
4:0	0x0	**T4_APERT_ID**: **0** = INIT

NV_PVIC_TFBIF_ACTMON_MCB_MASK_0

Offset: 0x2060
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000006 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0110)

Bit	Reset	Description
3	0x0	**ACTIVE**: **0** = FALSE **0** = INIT **1** = TRUE
2	0x1	**DELAYED_MC**: **0** = FALSE **1** = INIT **1** = TRUE

Bit	Reset	Description
1	0x1	**STALLED_MC**: **0** = FALSE **1** = INIT **1** = TRUE
0	0x0	**STARVED_MC**: **0** = FALSE **0** = INIT **1** = TRUE

NV_PVIC_TFBIF_ACTMON_MCB_BORPS_0

Offset: 0x2064
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000082 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1000,0010)

Bit	Reset	Description
7	0x1	**ACTIVE_OPERATION**: **0** = OR **1** = AND **1** = INIT
6	0x0	**ACTIVE_POLARITY**: **0** = INIT **0** = POSITIVE **1** = NEGATIVE
5	0x0	**DELAYED_MC_OPERATION**: **0** = INIT **0** = OR **1** = AND
4	0x0	**DELAYED_MC_POLARITY**: **0** = INIT **0** = POSITIVE **1** = NEGATIVE
3	0x0	**STALLED_MC_OPERATION**: **0** = INIT **0** = OR **1** = AND
2	0x0	**STALLED_MC_POLARITY**: **0** = INIT **0** = POSITIVE **1** = NEGATIVE
1	0x1	**STARVED_MC_OPERATION**: **0** = OR **1** = AND **1** = INIT

Bit	Reset	Description
0	0x0	**STARVED_MC_POLARITY**: **0** = INIT **0** = POSITIVE **1** = NEGATIVE

NV_PVIC_TFBIF_ACTMON_MCB_WEIGHT_0

Offset: 0x2068
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000001 (0b0000,0000,0000,0000,0000,0000,0000,0001)

Bit	Reset	Description
31:0	0x1	**VAL**: **1** = INIT

NV_PVIC_TFBIF_THI_TRANSPROP_0

Offset: 0x2070
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
16	X	**TZ_AUTH**
14:8	X	**STREAMID1**
6:0	X	**STREAMID0**

7.4 High Definition Audio (HDA)

7.4.1 Overview

This chapter describes the industry-standard High Definition Audio (HDA) controller. This role for this controller is solely to provide a multi-channel audio path to the Display Interface, refer to the corresponding chapter for more details.

This chapter describes the SoC implementation-specific HDA registers only. For information on the industry-standard registers defined by the HDA specification, refer to the publicly available High Definition Audio Specification from Intel® Corporation (also sometimes referred to by its code-name of Azalia).

The HDA block provides an HDA-compliant interface to the embedded audio "codec" in the Display Interface. Multiple output streams are supported.

7.4.1.1 High-level Features

- Compliant to High Definition Audio Specification Revision 1.0
- Supports DP 1.4 Audio feature
- Supports HDMI 1.4 Audio feature
- Supports one audio stream sending with HDMI or DP video stream
- Supports HBR Audio stream
- Supports audio stream with HDMI 2.1 FRL
- Supports DVFS latency with maximum latency of 240µs for 8-channel use case
- Supports HDMI CTS/DP audio timestamp packet generation
- Supports loopback mode for testing

Table 7.75 Audio Format Support

Format	Interface	Sampling bits	Sampling Frequency	Number of channels supported
LPCM	DP	16/20/24	32/44.1/48 kHz	2-8
LPCM	HDMI	16/20/24	32/48/96 kHz 44.1/88.2 kHz 176.4/192 kHz	2-8

Format	Interface	Sampling bits	Sampling Frequency	Number of channels supported
Compressed Audio (NON-HBR): AC3, DTS, MPEG1, MPEG2, MP3,MPEG2/4 AAC WMA	DP/HDMI	NA	N/A	AC3: 2/5.1 DTS 5.1 MPEG2-AAC 2/5.1
Compressed Audio (NON-HBR): DD+ Compressed Audio (HBR): TrueHD, DTS-HD	HDMI	NA	384/768 kHz	DD+ 8 TrueHD, DTS-HD: 7~8

There are two register groups defined for audio: one is for the FPCI interface to the system, and the other is for audio.

The FPCI registers have 256 bytes (0x0~0xFC). It includes FPCI configuration registers, AZA Hardware debug registers, registers like OB FIFO monitor, OB FIFO watermark, etc.

Registers can be accessed by:

1. NV_HDA_APB_PRI_CFG (or NV_ADDRESS_MAP_HDA_BASE+ NV_HDA_APB_DFPCI_CFG_OFFSET) + offset (via PRI bus)
2. NV_HDA_APB_DFPCI_CFG (or NV_ADDRESS_MAP_HDA_BASE) + offset (via FPCI bus)

HDA registers have 16 Kbytes (0x0000~0x3FFF) in BAR0 and it includes all HDA registers.

HDA registers can be accessed by NV_HDACODEC_APB_BAR1_START + offset (via FPCI bus)

7.4.2 HDA Registers

7.4.2.1 FCPI Interface Registers

HDA_AXI_BAR0_SZ_0
Offset: 0x0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000004 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0100)

Bit	Reset	Description
19:0	0x4	**AXI_BAR0_SIZE**: The size of the address range associated with BARi is in 4K increments. Value of 0 signifies BARi is not used.

HDA_AXI_BAR1_SZ_0

Offset: 0x4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**AXI_BAR1_SIZE**: The size of the address range associated with BARi is in 4K increments. Value of 0 signifies BARi is not used.

HDA_AXI_BAR2_SZ_0

Offset: 0x8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**AXI_BAR2_SIZE**: The size of the address range associated with BARi is in 4K increments. Value of 0 signifies BARi is not used.

HDA_AXI_BAR3_SZ_0

Offset: 0xc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**AXI_BAR3_SIZE**: The size of the address range associated with BARi is in 4K increments. Value of 0 signifies BARi is not used.

HDA_AXI_BAR0_START_0

Offset: 0x40
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

Reset: 0x03518000 (0b0000,0011,0101,0001,1000,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:12	0x3518	**AXI_BAR0_START**: The start of AXI address space for BARi. The AXI target address is compared to start/size for each BAR to determine if the access is to that BAR.

HDA_AXI_BAR1_START_0

Offset: 0x44
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:12	0x0	**AXI_BAR1_START**: The start of AXI address space for BARi. The AXI target address is compared to start/size for each BAR to determine if the access is to that BAR.

HDA_AXI_BAR2_START_0

Offset: 0x48
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:12	0x0	**AXI_BAR2_START**: The start of AXI address space for BARi. The AXI target address is compared to start/size for each BAR to determine if the access is to that BAR.

HDA_AXI_BAR3_START_0

Offset: 0x4c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:12	0x0	**AXI_BAR3_START**: The start of AXI address space for BARi. The AXI target address is compared to start/size for each BAR to determine if the access is to that BAR.

HDA_FPCI_BAR0_0

Offset: 0x80
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00040001 (0b0000,0000,0000,0100,0000,0000,0000,xxx1)

Bit	Reset	Description
31:4	0x4000	**FPCI_BAR0_START**: The start of FPCI address space mapped into the BARi range of PCI memory space. The 40-bit FPCI address is determined by a shift left 12 of the value of this register.
0	0x1	**FPCI_BAR0_ACCESS_TYPE**: Indicates if the address region is memory mapped versus configuration or IO space. 0 = Memory mapped access (PW only) 1 = IO/config access (NPW only)

HDA_FPCI_BAR1_0

Offset: 0x84
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000001 (0b0000,0000,0000,0000,0000,0000,0000,xxx1)

Bit	Reset	Description
31:4	0x0	**FPCI_BAR1_START**: The start of FPCI address space mapped into the BARi range of PCI memory space. The 40-bit FPCI address is determined by a shift left 12 of the value of this register.
0	0x1	**FPCI_BAR1_ACCESS_TYPE**: Indicates if the address region is memory mapped versus configuration or IO space. 0 = Memory mapped access (PW only) 1 = IO/config access (NPW only)

HDA_FPCI_BAR2_0

Offset: 0x88

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000001 (0b0000,0000,0000,0000,0000,0000,0000,xxx1)

Bit	Reset	Description
31:4	0x0	**FPCI_BAR2_START**: The start of FPCI address space mapped into the BARi range of PCI memory space. The 40-bit FPCI address is determined by a shift left 12 of the value of this register.
0	0x1	**FPCI_BAR2_ACCESS_TYPE**: Indicates if the address region is memory mapped versus configuration or IO space. 0 = Memory mapped access (PW only) 1 = IO/config access (NPW only)

HDA_FPCI_BAR3_0

Offset: 0x8c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000001 (0b0000,0000,0000,0000,0000,0000,0000,xxx1)

Bit	Reset	Description
31:4	0x0	**FPCI_BAR3_START**: The start of FPCI address space mapped into the BARi range of PCI memory space. The 40-bit FPCI address is determined by a shift left 12 of the value of this register.
0	0x1	**FPCI_BAR3_ACCESS_TYPE**: Indicates if the address region is memory mapped versus configuration or IO space. 0 = Memory mapped access (PW only) 1 = IO/config access (NPW only)

HDA_MSI_BAR_SZ_0

MSI BAR SIZE

Offset: 0xc0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0x0	**MSI_BAR_SIZE**: The size of the address range associated with MSI BAR is in 4K increments. Value of 0 signifies BAR is not used.

HDA_MSI_AXI_BAR_ST_0

MSI AXI BAR START

Offset: 0xc4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:12	0x0	**MSI_AXI_BAR_START**: The start of upstream AXI address space for MSI BAR. The upstream FPCI address is compared to start/1KB range for MSI BAR to determine if the access is MSI. Bits 31:12 of MSI BAR start correspond to AXI address bits 31:12.

HDA_MSI_FPCI_BAR_ST_0

MSI FPCI BAR START

Offset: 0xc8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,xxxx)

Bit	Reset	Description
31:4	0x0	**MSI_FPCI_BAR_START**: The start of upstream FPCI address space for MSI BAR. The upstream FPCI address is compared to start/1KB range for MSI BAR to determine if the access is MSI. Bits 31:4 of MSI BAR start correspond to UFPCI address bits 39:12.

HDA_MSI_VEC0_0

MSI VECTORi, i in [0,7], RW

Offset: 0x100
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_VECTOR0**: Each vector register corresponds to 32 of the possible 256 MSI vectors. VECTOR0 corresponds to MSI vectors 31-0. Vector7 corresponds to MSI vectors 255-223. When an upstream MSI is sent, the bit corresponding to the MSI vector is set to 1 by hardware if the corresponding enable bit is 1. The bit is set to 0 if a 1 is written to its location.

HDA_MSI_VEC1_0

Offset: 0x104
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_VECTOR1**: Each vector register corresponds to 32 of the possible 256 MSI vectors. VECTOR0 corresponds to MSI vectors 31-0. Vector7 corresponds to MSI vectors 255-223. When an upstream MSI is sent, the bit corresponding to the MSI vector is set to 1 by hardware if the corresponding enable bit is 1. The bit is set to 0 if a 1 is written to its location.

HDA_MSI_VEC2_0

Offset: 0x108
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_VECTOR2**: Each vector register corresponds to 32 of the possible 256 MSI vectors. VECTOR0 corresponds to MSI vectors 31-0. Vector7 corresponds to MSI vectors 255-223. When an upstream MSI is sent, the bit corresponding to the MSI vector is set to 1 by hardware if the corresponding enable bit is 1. The bit is set to 0 if a 1 is written to its location.

HDA_MSI_VEC3_0

Offset: 0x10c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_VECTOR3**: Each vector register corresponds to 32 of the possible 256 MSI vectors. VECTOR0 corresponds to MSI vectors 31-0. Vector7 corresponds to MSI vectors 255-223. When an upstream MSI is sent, the bit corresponding to the MSI vector is set to 1 by hardware if the corresponding enable bit is 1. The bit is set to 0 if a 1 is written to its location.

HDA_MSI_VEC4_0

Offset: 0x110
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_VECTOR4**: Each vector register corresponds to 32 of the possible 256 MSI vectors. VECTOR0 corresponds to MSI vectors 31-0. Vector7 corresponds to MSI vectors 255-223. When an upstream MSI is sent, the bit corresponding to the MSI vector is set to 1 by hardware if the corresponding enable bit is 1. The bit is set to 0 if a 1 is written to its location.

HDA_MSI_VEC5_0

Offset: 0x114
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_VECTOR5**: Each vector register corresponds to 32 of the possible 256 MSI vectors. VECTOR0 corresponds to MSI vectors 31-0. Vector7 corresponds to MSI vectors 255-223. When an upstream MSI is sent, the bit corresponding to the MSI vector is set to 1 by hardware if the corresponding enable bit is 1. The bit is set to 0 if a 1 is written to its location.

HDA_MSI_VEC6_0

Offset: 0x118
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_VECTOR6**: Each vector register corresponds to 32 of the possible 256 MSI vectors. VECTOR0 corresponds to MSI vectors 31-0. Vector7 corresponds to MSI vectors 255-223. When an upstream MSI is sent, the bit corresponding to the MSI vector is set to 1 by hardware if the corresponding enable bit is 1. The bit is set to 0 if a 1 is written to its location.

HDA_MSI_VEC7_0

Offset: 0x11c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_VECTOR7**: Each vector register corresponds to 32 of the possible 256 MSI vectors. VECTOR0 corresponds to MSI vectors 31-0. Vector7 corresponds to MSI vectors 255-223. When an upstream MSI is sent, the bit corresponding to the MSI vector is set to 1 by hardware if the corresponding enable bit is 1. The bit is set to 0 if a 1 is written to its location.

HDA_MSI_EN_VEC0_0

MSI ENABLE VECTORi, i in [0,7], RW

Offset: 0x140
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_ENABLE_VECTOR0**: Each vector register corresponds to the enable bit for 32 of the possible 256 MSI vectors. ENABLE VECTOR0 corresponds to enable bits for MSI vectors 31-0. Vector7 corresponds to enable bits for MSI vectors 255-223.When an upstream MSI is sent, the bit corresponding to the MSI vector is set to 1 by hardware if the corresponding enable bit is 1.

HDA_MSI_EN_VEC1_0

Offset: 0x144
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_ENABLE_VECTOR1**: Each vector register corresponds to the enable bit for 32 of the possible 256 MSI vectors. ENABLE VECTOR0 corresponds to enable bits for MSI vectors 31-0. Vector7 corresponds to enable bits for MSI vectors 255-223. When an upstream MSI is sent, the bit corresponding to the MSI vector is set to 1 by hardware if the corresponding enable bit is 1.

HDA_MSI_EN_VEC2_0

Offset: 0x148
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_ENABLE_VECTOR2**: Each vector register corresponds to the enable bit for 32 of the possible 256 MSI vectors. ENABLE VECTOR0 corresponds to enable bits for MSI vectors 31-0. Vector7 corresponds to enable bits for MSI vectors 255-223. When an upstream MSI is sent, the bit corresponding to the MSI vector is set to 1 by hardware if the corresponding enable bit is 1.

HDA_MSI_EN_VEC3_0

Offset: 0x14c
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_ENABLE_VECTOR3**: Each vector register corresponds to the enable bit for 32 of the possible 256 MSI vectors. ENABLE VECTOR0 corresponds to enable bits for MSI vectors 31-0. Vector7 corresponds to enable bits for MSI vectors 255-223. When an upstream MSI is sent, the bit corresponding to the MSI vector is set to 1 by hardware if the corresponding enable bit is 1.

HDA_MSI_EN_VEC4_0

Offset: 0x150
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_ENABLE_VECTOR4**: Each vector register corresponds to the enable bit for 32 of the possible 256 MSI vectors. ENABLE VECTOR0 corresponds to enable bits for MSI vectors 31-0. Vector7 corresponds to enable bits for MSI vectors 255-223. When an upstream MSI is sent, the bit corresponding to the MSI vector is set to 1 by hardware if the corresponding enable bit is 1.

HDA_MSI_EN_VEC5_0

Offset: 0x154
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_ENABLE_VECTOR5**: Each vector register corresponds to the enable bit for 32 of the possible 256 MSI vectors. ENABLE VECTOR0 corresponds to enable bits for MSI vectors 31-0. Vector7 corresponds to enable bits for MSI vectors 255-223. When an upstream MSI is sent, the bit corresponding to the MSI vector is set to 1 by hardware if the corresponding enable bit is 1.

HDA_MSI_EN_VEC6_0

Offset: 0x158
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_ENABLE_VECTOR6**: Each vector register corresponds to the enable bit for 32 of the possible 256 MSI vectors. ENABLE VECTOR0 corresponds to enable bits for MSI vectors 31-0. Vector7 corresponds to enable bits for MSI vectors 255-223. When an upstream MSI is sent, the bit corresponding to the MSI vector is set to 1 by hardware if the corresponding enable bit is 1.

HDA_MSI_EN_VEC7_0

Offset: 0x15c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_ENABLE_VECTOR7**: Each vector register corresponds to the enable bit for 32 of the possible 256 MSI vectors. ENABLE VECTOR0 corresponds to enable bits for MSI vectors 31-0. Vector7 corresponds to enable bits for MSI vectors 255-223. When an upstream MSI is sent, the bit corresponding to the MSI vector is set to 1 by hardware if the corresponding enable bit is 1.

HDA_CONFIGURATION_0

Configuration

Offset: 0x180
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x800e8e40 (0b1xxx,xxxx,xxxx,111x,10xx,111x,0100,0000)

Bit	R/W	Reset	Description
31	RW	0x1	**CLKEN_OVERRIDE**: This can override the clock enable in case of malfunction.
19	RW	0x1	**PW_NO_DEVSEL_ERR_CYA**: Setting this would enable fix for when we don't detect DECERR due to no DEVSEL for DS PWs only.
18	RO	0x1	**INITIATOR_READ_IDLE**: This read-only bit provides status reads on AFI upstream. A value of 1b indicates there are no outstanding reads to initiator.
17	RO	0x1	**INITIATOR_WRITE_IDLE**: This read-only bit provides status writes on AFI upstream. A value of 1b indicates there are no outstanding writes to initiator.
15	RW	0x1	**WDATA_LEAD_CYA**: Used to en(dis)able the handling of write data ahead of requests on IPFS AXI target
14	RW	0x0	**WR_INTRLV_CYA**: Used to en(dis)able the handling of interleaved write requests on IPFS AXI target
11	RO	0x1	**TARGET_READ_IDLE**: This read-only bit provides status reads to IPFS target. A value of 1b indicates there are no outstanding reads to downstream FPCI.
10	RO	0x1	**TARGET_WRITE_IDLE**: This read-only bit provides status writes to IPFS target. A value of 1b indicates there are no outstanding writes to downstream FPCI.
9	RO	0x1	**MSI_VEC_EMPTY**: This read-only bit provides status on whether MSI Vector registers have any active bits valid or not
7	RW	0x0	**UFPCI_MSIAW**: MSI After Write ordering rule. 1 = Whenever MSI is ready assert the interrupt 0 = Default behavior, apply MSIAW ordering rule
6	RW	0x1	**UFPCI_PWPASSPW**: Input to upstream FPCI 1 = Whenever write is ready, send it 0 = Write goes only when outstanding PWs outside of new write's region are retired (default).
5	RW	0x0	**UFPCI_PASSPW**: Input to upstream FPCI. Allow upstream FPCI reads to pass writes.
4	RW	0x0	**UFPCI_PWPASSNPW**: Used for upstream FPCI. Allow upstream FPCI PWs to pass NPWs.
3	RW	0x0	**DFPCI_PWPASSNPW**: Used for downstream FPCI. Allow downstream FPCI PWs to pass NPWs.
2	RW	0x0	**DFPCI_RSPPASSPW**: Input to downstream FPCI. Allow downstream FPCI responses to pass writes

Bit	R/W	Reset	Description
1	RW	0x0	**DFPCI_PASSPW**: Input to downstream FPCI. Allow downstream FPCI reads to pass writes.
0	RW	0x0	**EN_FPCI**: When the IPFS device block is disabled, it is completely invisible on the IPFS bus, i.e., it doesn't even process IPFS configuration accesses.

HDA_FPCI_ERROR_MASKS_0

FPCI Error Masks

Offset: 0x184
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**MASK_FPCI_MASTER_ABORT**: This bit allows FPCI error to be forwarded to AXI response when FPCI error response indicates Master Abort. 1 = forward error, 0 = return AXI OKAY response (2'b0)
1	0x0	**MASK_FPCI_DATA_ERROR**: This bit allows FPCI error to be forwarded to AXI response when FPCI error response indicates Data Error. 1 = forward error, 0 = return AXI OKAY response (2'b0)
0	0x0	**MASK_FPCI_TARGET_ABORT**: This bit allows FPCI error to be forwarded to AXI response when FPCI error response. This bit also covers decode error generated when there is no DEVSEL received indicates Target Abort. 1 = forward error, 0 = return AXI OKAY response (2'b0)

HDA_INTR_MASK_0

Interrupt Masks

Offset: 0x188
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,xxxx,xxx0,xxxx,xxx0)

Bit	Reset	Description
16	0x0	**IP_INT_MASK**: IP interrupt to MPCORE gated by mask.
8	0x0	**MSI_MASK**: MSI to MPCORE gated by mask.
0	0x0	**INT_MASK**: Interrupt to MPCORE gated by mask.

HDA_INTR_CODE_0

Interrupt Control

Offset: 0x18c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4:0	0x0	**INT_CODE**: Eight interrupt codes. If the code is 0, logging of the next interrupt is enabled **0** = INT_CODE_CLEAR: Clear interrupt code **1** = INT_CODE_INI_SLVERR: Interrupt code for MPCORE AXI SLVERR response to IPFS **2** = INT_CODE_INI_DECERR: Interrupt code for MPCORE AXI DECERR response to IPFS **3** = INT_CODE_TGT_SLVERR: Interrupt code for PCIe endpoint FPCI target abort or data error response to IPFS **4** = INT_CODE_TGT_DECERR: Interrupt code for PCIE2 FPCI master abort response to IPFS **5** = INT_CODE_TGT_WRERR: Interrupt code for bufferable write to non-posted write address region **6** = RSVD1: Reserved **7** = INT_CODE_DFPCI_DECERR: Interrupt code for PCIE2 response to downstream request when downstream FPCI address does not fall in a claimable downstream region **8** = INT_CODE_AXI_DECERR: Interrupt code for IPFS response to downstream request when AXI target AXI address does not fall in any of IPFS downstream BARs **9** = INT_CODE_FPCI_TIMEOUT: Interrupt code for FPCI Timeout **10** = RSVD2: Reserved for future expansion **11** = RSVD3 **12** = RSVD4 **13** = RSVD5 **14** = RSVD6 **15** = INT_CODE_SM_FATAL_ERROR: Interrupt code for SM fatal error **16** = INT_CODE_SM_NON_FATAL_ERROR: Interrupt code for SM non-fatal error

HDA_INTR_SIGNATURE_0

Interrupt Signature

Offset: 0x190
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00x0)

Bit	Reset	Description
31:2	0x0	**INT_INFO**: For interrupt codes 1-5/7-8, it contains address bits [31:2], either in FPCI memory space or AXI space. For FPCI generated errors, the info contains FPCI address. For AXI/IPFS generated errors, the info contains AXI address.
0	0x0	**DIR**: Indicates direction of the AXI/FPCI transaction. 1 = Read /0 = Write If signature type is 6 (sideband message), this field is 1. **0** = WRITE: Interrupt due to a write transaction **1** = READ: Interrupt due to a read transaction

HDA_UPPER_FPCI_ADDR_0

Upper FPCI Address

Offset: 0x194
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**INT_INFO_UPPER**: These 8 bits are the upper byte of captured FPCI address (bits[39:32]) when interrupt code is 3, 4 or 7. These bits determine the region in the HyperTransport Address Map that was accessed.

HDA_IPFS_INTR_ENABLE_0

IPFS Interrupt Enable

Offset: 0x198
Read/Write: R/W

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xx00,xxxx,0000,0000)

Bit	Reset	Description
13	0x0	**EN_SM_NON_FATAL_ERROR**: Enable bit for interrupt code 15
12	0x0	**EN_SM_FATAL_ERROR**: Enable bit for interrupt code 14
7	0x0	**EN_FPCI_TIMEOUT**: Enable bit for interrupt code 9
6	0x0	**EN_AXI_DECERR**: Enable bit for interrupt code 8
5	0x0	**EN_DFPCI_DECERR**: Enable bit for interrupt code 7
4	0x0	**EN_TGT_WRERR**: Enable bit for interrupt code 5
3	0x0	**EN_TGT_DECERR**: Enable bit for interrupt code 4
2	0x0	**EN_TGT_SLVERR**: Enable bit for interrupt code 3
1	0x0	**EN_INI_DECERR**: Enable bit for interrupt code 2
0	0x0	**EN_INI_SLVERR**: Enable bit for interrupt code 1

HDA_UFPCI_CONFIG_0

Upstream FPCI Configuration

Offset: 0x19c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000002 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0010)

Bit	Reset	Description
4:0	0x2	**UNITID_T0C0**: Upstream FPCI Unit ID for controller 0. HyperTransport, upstream FPCI request

HDA_CFG_REVID_0

CFG_REVID register

Offset: 0x1a0
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00001004 (0bxxxx,xxxx,xxxx,00xx,xx01,00xx,xxx0,01xx)

Bit	R/W	Reset	Description
19	RO	0x0	**DEV2SM_NONISO_REQUEST_PEND**: This is to tell if there is a non ISO request pending. **0** = NO **1** = YES
18	RO	0x0	**DEV2SM_ISO_REQUEST_PEND**: This is to tell if there is an ISO request pending. **0** = NO **1** = YES
13:12	RW	0x1	**STRAP_CPU_MODE**: MCP: mode to send MSI. Can have it programmable. **0** = NB_INTEL **1** = NB_AMD **2** = AMD **3** = TMTA
11	RW	0x0	**CFG_REVID_WRITE_ENABLE**: MCP: the enable to override the revision ID. Can have it programmable **0** = CLEAR **1** = SET
10	RW	0x0	**CFG_REVID_OVERRIDE**: MCP: a way to override the current revision ID. Can have it programmable **0** = DISABLE **1** = ENABLE
4	RO	0x0	**DEV2LEG_NONCOH_REQUEST_PEND**: MCP: Tells the leg block that we have a non coherent request pending. **0** = NO **1** = YES
3	RO	0x0	**DEV2LEG_COH_REQUEST_PEND**: MCP comment: Tells the leg block that we have a coherent request pending **0** = NO **1** = YES
2	RW	0x1	**SM2DEV_FPCI_TIMEOUT_EN**: FPCI timeout enable bit for Controller **0** = DISABLE **1** = ENABLE

HDA_FPCI_TIMEOUT_0

FPCI_TIMEOUT register

Offset: 0x1a4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000f0000 (0bxxxx,xxxx,xxxx,1111,0000,0000,0000,0000)

Bit	Reset	Description
19:0	0xf0000	**SM2ALL_FPCI_TIMEOUT_THRESH**: This sets the timeout thresh value for FPCI bus. The starts counting for each queues (ISO/NISO - RD/WR) have pending request in FPCI wrapper, the count resets when the requests popped

HDA_TOM_0

Top of Memory Limit

Offset: 0x1a8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x3fff0fff (0bxx11,1111,1111,1111,xxxx,1111,1111,1111)

Bit	Reset	Description
29:16	0x3fff	**LEG2ALL_TOM2**: Top of Memory Limit 2.
11:0	0xfff	**LEG2ALL_TOM1**: Top of Memory Limit 1.

HDA_INITIATOR_ISO_PW_RESP_PENDING_0

Initiator ISO PW Response Pending

Offset: 0x1ac
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0

Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**NUM_INITIATOR_ISO_PW_RESP_PEND**: Number of pending initiator ISO PW responses

HDA_INITIATOR_NISO_PW_RESP_PENDING_0

Initiator Non-ISO PW Response Pending

Offset: 0x1b0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**NUM_INITIATOR_NISO_PW_RESP_PEND**: Number of pending initiator NISO PW responses

HDA_INTR_STATUS_0

IPFS Interrupt Status

Offset: 0x1b4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**IP_INTR_STATUS**: Status of IP interrupt
1	0x0	**MSI_INTR_STATUS**: Status of MSI interrupt
0	0x0	**IPFS_INTR_STATUS**: Status of IPFS interrupt

HDA_DFPCI_BEN_0

Downstream FPCI Byte Enables

Offset: 0x1b8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
31	0x0	**EN_DFPCI_BEN**: Enable bit for ben; when set, programmed be is sent on DFPCI bus
3:0	0x0	**DFPCI_BYTE_ENABLE_N**: Active low byte enables

HDA_CLKGATE_HYSTERESIS_0

Offset: 0x1bc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000014 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0001,0100)

Bit	Reset	Description
7:0	0x14	**CLK_DISABLE_CNT**: Number of IPFS clock cycles to wait after clock gating criteria is met to disable IPFS/FPCI clocks

HDA_MISC_0

Offset: 0x1dc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**HDA_DEVICE_DIS**: Serial ATA Interface 0 Disable 1 = HDA Interface 0 Disabled (Not seen as part of PCI space) 0 = HDA Interface 0 Enabled. **0** = ENABLE **1** = DISABLE

HDA_HDA_GSC_ID_0

GSC_ID register for Software to program the GSC_ID value that will transmit through

Offset: 0x1e0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4:0	0x0	**HDA_GSC_ID**: Default GSC_ID is 0x0

HDA_ORDERING_RULES_0

Adding new registers related to IPFS.

Offset: 0x1e4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,000x)

Bit	Reset	Description
3	0x0	**UPSTREAM_MSIAW**: Modified Upstream MSIAW ordering. 0 = MSIAW behaviour 1 = Old MSIAW behaviour
2	0x0	**UPSTREAM_RESPAW**: Modified RespAW ordering. 0 = RespAW behaviour 1 = Old RespAW behaviour
1	0x0	**UPSTREAM_RAW**: Modified RAW ordering. 0 = RAW behaviour 1 = Old RAW behaviour

HDA_A2F_UFPCI_CFG0_0

Offset: 0x1e8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000050 (0b0000,0000,xxxx,x000,0000,0000,0101,0000)

Bit	Reset	Description
31:24	0x0	**STATIC_WAIT_IDLE_CNTR**
18:16	0x0	**STATIC_UFPCI_UFA_STARVE_CNTR_PRI1**
15:12	0x0	**STATIC_UFPCI_UFA_STARVE_CNTR_PRI0**
11:10	0x0	**STATIC_UFPCI_RR_BURST_SZ_PRI1**
9:8	0x0	**STATIC_UFPCI_RR_BURST_SZ_PRI0**
7	0x0	**STATIC_WAIT_CLAMP_EN**
6	0x1	**STATIC_UFPCI_UFA_DYN_BLOCK_EN**
5	0x0	**STATIC_UFPCI_UFA_BLK_COHERENT**
4:2	0x4	**STATIC_UFPCI_BLOCK_CMD_THRESHOLD**
1	0x0	**STATIC_CYA_UFA_ARB**
0	0x0	**STATIC_CYA_BACK2BACK_UPSTREAM_BLOCK**

HDA_A2F_UFPCI_CFG1_0
Offset: 0x1ec
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**STATIC_WAIT_UNCLAMP_CNTR**

7.4.2.2 HDMI HDAC Registers

Procedure to write HDMI_HDAC_BFM_ registers (not including the FIFO registers)

1. write audio_emu2 = data
2. write audio_emu1.addr, write=1
3. write audio_emu1.addr, write=0
4. write audo_emu2.data = handshake.REQ=1
5. write audio_emu1.addr = HANDSHAKE, write=1
6. write audio_emu1.addr = HANDSHAKE, write=0
7. poll until handshake.busy = 1

8. write audo_emu2.data = handshake.REQ=0
9. write audio_emu1.addr = HANDSHAKE, write=1
10. write audio_emu1.addr = HANDSHAKE, write=0
11. poll until handshake.busy = 0

HDMI_HDAC_BFM_HANDSHAKE_0
Offset: 0x80
Byte Offset: 0x200
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	R/W	Reset	Description
1	RO	0x0	**ACK**
0	WO	0x0	**REQ**

HDMI_HDAC_BFM_CONFIG_0
Offset: 0x81
Byte Offset: 0x204
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,xxx0)

Bit	Reset	Description
7:4	0x0	**CODEC_ADDRESS**
0	0x0	**TX_ENABLE**

HDMI_HDAC_BFM_CMD_FIFO_LSB_0
Offset: 0x82
Byte Offset: 0x208
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA_31_0**

HDMI_HDAC_BFM_CMD_FIFO_MSB_0

Write to MSB triggers the 40-bit write of LSB and MSB

Offset: 0x83
Byte Offset: 0x20c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**DATA_39_32**

HDMI_HDAC_BFM_CMD_FIFO_STATE_0
Offset: 0x84
Byte Offset: 0x210
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXX000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**COUNT**
0	X	**EMPTY**

HDMI_HDAC_BFM_RESP_CNT_0

Non-zero responses are enqueued

Offset: 0x85
Byte Offset: 0x214
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0

Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**CNT**

HDMI_HDAC_BFM_RESP_DATA_0

Offset: 0x86
Byte Offset: 0x218
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

HDMI_HDAC_BFM_UNSOLICITED_RESP_CNT_0

Offset: 0x87
Byte Offset: 0x21c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**CNT**

HDMI_HDAC_BFM_UNSOLICITED_RESP_DATA_0

Offset: 0x88
Byte Offset: 0x220
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

HDMI_HDAC_BFM_STREAM0_CTRL_0

Offset: 0x89
Byte Offset: 0x224
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00010031 (0bxxxx,xxxx,000x,0001,0000,0000,x011,0001)

Bit	Reset	Description
23:22	ONE	**STRIPE**: **0** = ONE **1** = TWO **2** = FOUR
21	0x0	**ENABLE**
19:16	0x1	**ID**: HD Audio stream format structure
15	PCM	**TYPE**: **0** = PCM **1** = NONPCM
14	F_48_0_KHZ	**BASE**: **0** = F_48_0_KHZ **1** = F_44_1_KHZ
13:11	X1	**MULT**: **0** = X1 **1** = X2 **2** = X3 **3** = X4
10:8	DIV1	**DIV**: **0** = DIV1 **1** = DIV2 **2** = DIV3 **3** = DIV4 **4** = DIV5 **5** = DIV6 **6** = DIV7 **7** = DIV8
6:4	BPS_24	**BITS**: **0** = BPS_8 **1** = BPS_16 **2** = BPS_20 **3** = BPS_24 **4** = BPS_32
3:0	0x1	**CHAN**: CHAN = num_channels - 1

HDMI_HDAC_BFM_STREAM0_CHSTS_0

SPFID channel status and valid

Offset: 0x8a
Byte Offset: 0x228
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,xxxx,0000)

Bit	Reset	Description
15	0x0	**GEN_LEVEL**
14:8	0x0	**CLASS_CODE**
3	0x0	**PRE**
2	0x0	**COPY**: SPDIF convention: 0 = copyright, 1 = no copyright
1	0x0	**AUDIO**
0	0x0	**PRO**

HDMI_HDAC_BFM_STREAM0_SPDIF_VALID_0

Offset: 0x8b
Byte Offset: 0x22c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**V**
0	0x0	**VCFG**

HDMI_HDAC_BFM_STREAM0_SAMPLE_COUNT_0

Offset: 0x8c
Byte Offset: 0x230
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**VAL**

HDMI_HDAC_BFM_STREAM0_SP_COUNT_0
Offset: 0x8d
Byte Offset: 0x234
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**VAL**

HDMI_HDAC_BFM_STREAM_FIFO_HEADER_0
Offset: 0x8e
Byte Offset: 0x238
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
17	X	**SDO4_EN**
16	X	**SDO2_EN**
15	X	**LAST**
15:8	X	**BYTE_COUNT**
7:0	X	**STREAM_TAG**

HDMI_HDAC_BFM_STREAM_FIFO_DATA_0
Offset: 0x8f
Byte Offset: 0x23c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**DATA3**
23:16	X	**DATA2**
15:8	X	**DATA1**
7:0	X	**DATA0**

HDMI_HDAC_BFM_STREAM_FIFO_HEADER_STATE_0

Offset: 0x90
Byte Offset: 0x240
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXX000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**COUNT**
0	X	**EMPTY**

HDMI_HDAC_BFM_STREAM_FIFO_DATA_STATE_0

Offset: 0x91
Byte Offset: 0x244
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXX000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**COUNT**
0	X	**EMPTY**

HDMI_HDAC_BFM_STATE_0

Offset: 0x92
Byte Offset: 0x248
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0

Reset: 0x00XXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
23:20	X	**CODEC_STRM_ID**
16	X	**IDLE**
15:0	X	**FRAME_CNT**

7.5 Display Controller

7.5.1 Overview

The Display Controller is based on the NVDisplay 4.0 architecture, and has changed from prior NVIDIA SoCs. The main new features of NVDisplay 4.0 architecture are DSI output, native 4:2:0 HDMI output, and support for HDMI 2.1 Fixed Rate Link mode.

The MIPI DSI protocol is used in some applications. The DSI encoder is handled separately but integrated into the Display hierarchy. The DSI encoder is another form of output resource.

Native 4:2:0 output supports the 2head1Or feature (both heads are combined to drive a single panel over one output resource) to drive 8K HDMI panels without DSC compression. The display processing pipeline down samples 4:4:4 pixels to 4:2:0.

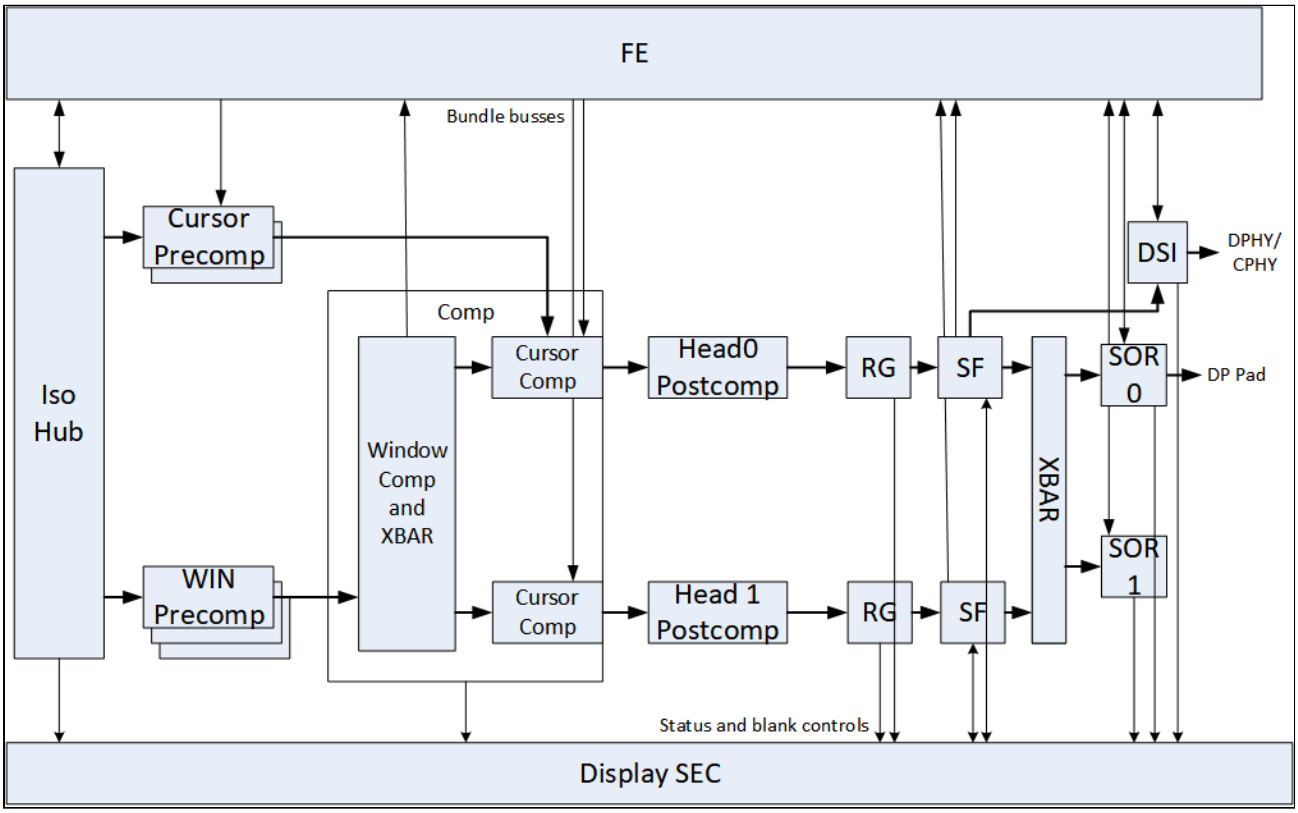
The HDMI 2.1 Fixed Rate Link mode increases the available HDMI link bandwidth by approximately 3x over HDMI 2.0. It also permits DSC compressed streams over HDMI.

The Display Controller has two heads and four windows. A window is a single surface that can be assigned to any head. Each head supports a single cursor, which can be composited on top of all windows.

There is some amount of per-window pixel processing (the precomp pipeline) and some amount of per-head pixel processing (the post-comp pipeline). There is a single shared Isochronous Memory Hub (ISO Hub) for memory access, which services all window surfaces. The output from any head can be routed to any SOR; except only head0 can be attached to DSI. There is a crossbar between each SOR to each output link.

All four windows are Full windows. Full windows contain the full HDR pixel processing functionality including a precomp scaler and tone mapping.

Figure 7.24 Top-Level Block Diagram



7.5.1.1 Standards Supported

- HDMI2.1
- DP1.4a
- eDP1.4
- HDCP1.3 / HDCP2.3
- HDA1.0

7.5.1.2 Glossary

Note that different names may end up in the same acronym when the entire TRM is put in perspective. The acronyms listed here are within the context of the Display Controller chapter.

Term	Definition
AZA	Intel Azalia Audio Controller
ASR	Aggressive Self Refresh
bpc	Bit Per Component

Term	Definition
bpp	Bit Per Pixel
BPP	Byte Per Pixel
CEC	Consumer Electronics Control
CMGR	Clock Manager
DC	Display Controller
DisplayPort	DisplayPort
DPAUX	DisplayPort Auxiliary channel
DVFS	Dynamic Voltage Frequency Scale
eDP	embedded DisplayPort (VESA)
FB	Frame Buffer
HDMI	High-Definition Multimedia Interface
HDCP	High-bandwidth Digital Content Protection
H-DA	High Definition Audio
HBR	High Bit Rate (2.7 Gbps per lane)
HBR2	High Bit Rate 2 (5.4 Gbps per lane)
HBR3	High Bit Rate 3 (8.1 Gbps per lane)
HPD	Hot Plug Detect
ISO Hub	Isochronous Memory Hub
LUT	Look-Up Table
PCLK	Pixel Clock
RBR	Reduced Bit Rate
RG	Raster Generator
SOR	Serial Output Resource (changing display-raster pixels into different protocols such as DisplayPort, HDMI)
TMDS	Transition-Minimized Differential Signaling
TSC	Time Stamps Controller
VPR	Video Protected Region
VR	Virtual Reality
WGRP	Window Group
WinTZ	Trustzone Window

7.5.1.3 Relevant Chapters in the TRM

- Address Map
- Clock Controller and Reset (CAR)
- CEC
- HDA
- Host Controller
- Memory Controller (MC)

7.5.1.4 Features

7.5.1.4.1 General

- Independent scan out paths (heads)
  - Two heads
- One cursor on each head
  - A8R8G8B8
  - A1R5G5B5
  - Cursor is enabled on a head if the head has at least one Window attached. Window need not be enabled.
  - The cursor pixels must be linearly encoded to be properly composited with Window pixels.
  - XOR blending mode is approximated in the FP16 compositor.
  - Max cursor size: 256 x 256
- Windows
  - Four windows (four full windows)
  - Windows can be dynamically assigned to any head.
- Max raster size: 32767 x 32767
  - Max active region 8192 x 8192
- Max input surface size: 32768 x 32768
  - Max fetched size: 8208 x 8208
- Max pclk: up to 1,190 MHz (this may vary between the variants, confirm from the relevant datasheet)
  - Two heads may be combined together with compression or 4:2:0 to support monitors with pclk frequencies up to 2,380 MHz (see note on previous line).
- One-Shot and Continuous display modes
  - Note that stereo is not supported in One-Shot mode.
  - DSI is not supported in One-Shot mode.
- Semaphores
  - Separate acquire and release window semaphores

    - Six release-only semaphores per head based on RG line number.
    - Support both 32-bit and 64-bit acquire and release values.
    - Supports Strict Equal, Strict Greater-or-Equal, and Circular Greater-or-Equal acquisition modes.
- Notifiers
    - CRC notifier
    - Completion notifier
- Flip control
    - minimum present interval
    - Timestamp Flip
    - Non-Tearing and Immediate flip modes
        - Immediate flip supported for RGB only
    - Generic Interlock between display channels
    - Flip Lock
- Push buffers
- Hardware clipping of windows to the viewport
- Independent channels for cursor position and window position
- Programmable viewport position
    - Overscan border
- Error checks
- Context DMAs
- Hardware-controlled mode sets
- Glitchless mode switches

7.5.1.4.2 Input Surface

Color Formats

- Indexed Color:
    - I8
- 2BPP RGB:
    - R4G4B4A4, R5G6B5, A1R5G5B5, R5G5B5A1
- 4BPP, 8bpc RGB:
    - A8R8G8B8, X8R8G8B8, A8B8G8R8, X8B8G8R8
- 4BPP, 10bpc RGB:
    - A2R10G10B10, A2B10G10R10
- 64-bit RGB:
    - R16_G16_B16_A16_NVBIAS, R16_G16_B16_A16
- FP16 RGB

    - RF16_GF16_BF16_AF16
- Packed YUV 422:
    - Y8_U8__Y8_V8_N422, U8_Y8__V8_Y8_N422
- Semi-Planar 8bpc:
    - Y8___U8V8_N444, Y8___U8V8_N422, Y8___V8U8_N420
- Semi-Planar 10bpc:
    - Y10___U10V10_N444, Y10___U10V10_N422, Y10___V10U10_N420
- Semi-Planar 12bpc:
    - Y12___U12V12_N444, Y12___U12V12_N422, Y12___V12U12_N420
- Planar 8bpc:
    - Y8___U8___V8_N444, Y8___U8___V8_N420
- No support for 10bpc or 12bpc planar
- YUV surfaces have the following restrictions:
    - Surfaces with chroma subsampling must have an even number of pixels/lines in the direction of subsampling.
    - When 90/270 rotation is enabled, all YUV surfaces must have even height.
    - When 90/270 rotation is not enabled, all YUV surfaces must have even width.

Isochronous Memory Hub

- Unified Mempool and Display fetch arbitration=1380 KB
- Fetch Metering
- Mempool Drain metering
- Horizontal and vertical-flipped scan direction
- Rotation
    - BLx8 fetch pattern used for rotation
    - Rotation is supported on all windows.
    - No more than one window using the I8 or YUV surface formats can be rotated (90/270 degrees) at any given time
- Pitch surface support
- Block Linear surface support
    - NVDisplay fetches BLx4 and uses XBAR_RAW addressing.
- VPR-capable read-only client
- 128 B read request interface
    - Response bus is 64 B
- Critical watermark to limit buffering when immediate flip is active.
    - Defined as a number of lines
- Midframe mclk switch
    - Used for DVFS

- Hardware Performance Monitor support

7.5.1.4.3 Pixel Processing

- Pipeline Depth
  - Fixed Point for some stages: 16 bpc, [-1.5,2.5] range (2 range extension bits)
  - FP16 for other stages
- Alpha blending
  - such as full alpha, per-window color key, global alpha
  - alpha blending is always done with FP16 in the linear color domain

Per Window Processing

- Chroma up-conversion for chroma-decimated YUV formats
  - Chroma overfetch
- Multiple per-window CSC matrices
  - 3 x 4 matrix multiplication with software-defined coefficients in the [-16, 16) range
  - There are several CSC matrixes at different stages of the pipe.
- Per-window LUTs (ILUT, CSC0LUT, CSC1LUT)
  - programmable matrix + variable step size LUT-based CSC from input to ICtCp; also from ICtCp to RGB with final output gamut. The scaling and tone mapping are done in ICtCp domain.
  - Input range is limited to [0,1]. Can be mirrored around 0 to support [-1,1]
  - ILUT
    - Used to convert from fixed point to FP16 linear
    - Variable step size and direct index mode
    - Maximum size: 1025 entries
    - 16 bpc
    - Programmed using the Context DMA method
  - CSC0LUT
    - Used to convert from FP16 to fixed point
    - Variable step size
    - Maximum size: 513
    - Programmed directly using methods
  - CSC1LUT
    - Used to convert from fixed point to FP16 linear
    - Variable step size
    - Maximum size: 513
    - Programmed directly using methods
- Precomposition scaling

    - Maximum line width
        - 5120 pixels in 2-tap mode
        - 2560 pixels in 5-tap mode
    - Maximum downscale factor
        - 2x in both directions in 2-tap mode
        - 4x in both directions in 5-tap mode
    - Scaler overfetch
    - Programmable 5-tap coefficients
- Tone Mapping (TMO)
    - LUT-based intensity mapping, and 3 intensity zones with configurable color correction settings
- Programmable range clamping before composition

Per Head Processing

- Post-composition LUT
    - Used to convert from FP16 linear to fixed-point gamma-encoded pixels
    - Variable step size. Also supports direct10 mode for medical imaging (DICOM)
    - Maximum size: 1025 entries
    - 14 bpc, [0,1] range. Can be mirrored around 0 to support [-1,1]
- Dither
    - 6, 8, 10 bpc output
    - Static and Dynamic Error Accumulation Dither
    - Temporal Dither
    - 2 x 2 Static and Dynamic Dither
    - Round-to-nearest (12 bpc output supported with this mode)
- Chroma Low-Pass filter (to facilitate 422 output)
- Programmable output CSC matrix
- Overscan border is added when the output viewport is smaller than the active raster region
- Programmable range clamping before output
- Display Stream Compression (DSC) v1.2a
    - RGB444/YCbCr444 input at 8/10/12bpc
    - 6bpp-32bpp output pixel depth with 1/16bpp precision
    - 1, 2, 4 slices per line
    - Max line width: 5120
    - Max pclk with DSC: same as max pclk
- 4:2:0 chroma downsampling
    - Convert from 4:4:4 to 4:2:0
    - Additional chroma overfetch added for this stage

    - Maximum line width: 5120
- Regional CRCs
    - Nine programmable regions with software-defined golden values. Used for safety features.

7.5.1.4.4 Output

- 2Head1OR
    - The DSC compressed or 4:2:0 packed output from two heads can be merged to output an 8K stream. (Note that DSC only supports 4:4:4).
    - DSC bpp must be <=16bpp in this mode.
    - Maximum raster active width in 2head1OR mode is 7680 (3840 per head)
- Raster Lock
    - Internal locking
    - Stereo phase lock
- Stall Lock
- Progressive output only
    - No support for interlaced
- Stereo
    - Frame Interleave
    - Hardware-constructed Frame packed Stereo for HDMI
- Two programmable line interrupts (registers)
- Six programmable line semaphores (methods)
    - Can also trigger interrupts
- DisplayPort 1.4a (No DP2.0 support)
    - Up to 8 channel 48 kHz Audio
    - RBR, HBR, HBR2, HBR3
    - RGB444 and YUV444 6, 8, 10, 12 bpc (6 bpc is RGB only)
    - YUV422 8, 10, 12 bpc
    - Maximum pclk over a single 4 lane HBR3 link without downspread, FEC, or DSC:
        - 1190 MHz at 16bpp (YUV 422)
        - 1,190 MHz at 18bpp
        - 1,080 MHz at 24bpp
        - 864 MHz at 30bpp
        - 720 MHz at 36bpp
    - Two Generic info frames per head (likely used for VSC and static HDR)
        - No native support for Dynamic HDR metadata
    - Forward Error Correction (FEC)
    - DSC transport
        - PPS can be used as another generic info frame when not being used for DSC

    - DisplayPort Multistream (MST)
        - Any combination of heads can be sent to any combination of SORs (provided there is enough link bandwidth to support those streams)
    - Dual-MST
        - Not supported with YUV 4:2:2 output format
        - Not supported with FEC
- eDP 1.4b
    - PSR v1
    - NLT
    - NVSR
    - Support for intermediate link rates (2.16, 2.43, 3.24, 4.32, 6.75 Gbps/lane)
- HDMI 2.0a and HDMI 1.4
    - Maximum pclk supported in non-FRL mode is as follows:

	8bpc	10bpc	12bpc
RGB / YUV 4:4:4	600 MHz	320 MHz	400 MHz
YUV 4:2:2	600 MHz	600 MHz	600 MHz
YUV 4:2:0	1,200 MHz	640 MHz	800 MHz

    - Up to 8 channel 192 kHz Audio
    - AVI, VSI, GCP, and one Generic Info frame supported
        - No support for Dynamic HDR metadata
- HDMI 2.1
    - Fixed Rate link at 3G, 6G, 8G, 10G, or 12G
    - RGB/YUV4:4:4 and YUV 4:2:0 uncompressed
    - DSC transport over FRL (RGB/YUV 4:4:4 only)
    - No 4:2:2 support in FRL mode
    - No hardware support for other HDMI 2.1 features has been added. Some features like VRR and QFT may be supported using software-only changes.
    - Maximum supported pclk: 2,376 MHz (using 2Head1Or)
- HDCP
    - HDCP 1.4 for HDMI
    - HDCP 1.3 Amendment 1.1 for DisplayPort
    - HDCP 2.3 for both DisplayPort and HDMI
- SOR Links
    - Two SOR and one total link
    - Either SOR can drive the link
    - No dual-link support
- DSI

- Head 0 only
- Four lane DSI-D
- Three trio DSI-C
- Max pclk: 500 MHz (1536 x 2456 @ 75 Hz)
- 6 bpc, 8 bpc, 10 bpc, 12 bpc RGB
- DSC (8, 10, 12, 16 bpp only)
- No audio support

7.5.1.4.5 Performance

NVD4.0 is designed to target a maximum single head display resolution of 7680 x 4320 @ 30 Hz. 8K at 60 Hz requires two heads. The maximum pixel clock is 1,190 MHz. There are a maximum of 4 window surfaces and 2 cursor surfaces fetched at any given time. Note that the total number of surfaces active at any given time is limited by the available guaranteed ISO bandwidth from FB.

Table 7.76 FB Bandwidth Requirement

Max FB bandwidth	20 GB/s
Heads	2
Windows	4

*1 GB = 1000 * 1000 * 1000 bytes

Minimum hblank

Pixel processing within the Display pipeline can introduce bubbles between each line of pixels. The output protocol may incur additional overhead on top of the pixel stream. Hblank of the raster must absorb the overhead bubbles and encoding.

Table 7.77 Minimum Hblank Requirement

Unit	Worst-case per-line gap	Reason
ISO Hub	Packed: 6 Planar/Semi-planar: 16	Unused data at the beginning and end of a line needs to be dropped. Planar Format: Worst case: 16 disp-clks SP Format: Worst case: 16 disp-clks Packed Format: Worst case: 6 disp-clks
Precomp	RGB: 34 YUV: 40	YUV chroma up-conversion: 2 cycles PXCTL state machines in PXGEN: 8 cycles Horizontal Scaler: 11 cycles Vertical Scaler: 15 cycles Extra pixels from YUVCrop: 4
Comp	6	Cursor inserts a 6-cycle bubble.

Unit	Worst-case per-line gap	Reason
Post-comp	8	Horizontal Chroma Low-Pass filter: 8
RG	0	RG requires at least 3 pixels of hblank for HFP, HSYNC, and HBP, but it does not introduce any bubbles of its own.
SF/SOR		DisplayPort min hblank can be up to 52 for single-head DSC. HDMI min hblank, no audio, no HDCP 1.3: 14 HDMI min hblank, w/audio, no HDCP1.3: 64 HDMI min hblank, w/audio, and HDCP 1.3: 106
FE	0	FE is not directly involved with the line-to-line behavior of the pipeline.

The minimum hblank refers to the size of hblank programmed in the RG methods. When 2head1Or is enabled, DD halves the RG's horizontal raster parameters. A configuration that requires N pixels of blanking requires a resolution with 2N pixels of blanking when this mode is used.

The pessimistic minimum hblank for a YUV planar or semiplanar surface is:

- MAX (SOR min hblank, (ISO Hub + precomp + comp + post-comp + RG))
- MAX (SOR min hblank, 70)

If some features are not enabled, the minimum hblank may be smaller.

Minimum vblank

To do an mclk switch during vblank, fetching must be stopped for the duration of the mclk switch or DVFS event. A complete spool up before the active region begins when fetching starts. If precomp scaling, post-comp scaling, all LUTs, DSC, 2head1Or, and mempool compression are enabled, then 22.5 scan lines are needed to complete spool up. Vblank should be greater than <fb stop time> + 22.5 lines. The 22.5 lines may be reduced if some features are not in use or if dispclk is running faster than the minimum dispclk required for the current mode.

7.5.1.4.6 Max Resolutions

The maximum pixel clock is 1,190 MHz. With the 2head1Or feature, the maximum targeted resolution is 7680 x 4320 @ 60 Hz (pclk 2,376 MHz).

7.5.1.4.7 State Management

Most method state in Display is triple buffered.

- **Assembly**: This state is set when a method is processed. The assembly value is sent on the bundle bus to the appropriate submodule. This state does not directly affect functionality. This state is needed because each new Display configuration requires multiple methods,

which are not atomic. This state permits software to *assemble* the next Display configuration over multiple methods.

- **Armed**: An update promotion bundle on the bundle bus atomically copies the assembly state for a single channel to armed. This intermediate state is needed so that software can change the Display configuration multiple times per frame. After the promotion to Armed, software is free to overwrite the Assembly state and potentially perform another Update before the next frame begins.
- **Active:** At the start of each frame, a LoadV on the pixel bus promotes the current Armed state to Active for all channels associated with that head. Active state controls the functionality of Display. All pixels that are processed before the LoadV use the Active state from the previous frame. All pixels that are processed after the LoadV use the Active state from the new frame.

Software controls the location of update in the method stream. Multiple channels may be updated at the same time using interlocked updates. During every frame, hardware generates LoadV.

7.5.1.4.8 Software Interfaces

Methods and registers control the display configuration state. Methods are triple buffered. Registers are typically single buffered and take effect immediately. There are exceptions to both rules. Methods enter the Display hardware through pushbuffers. The display itself manages the pushbuffers, rather than Host. Register programming is done through private register accesses (MMIO) through APB. Sets of methods are collected into channels, which all promote (update) at the same time. Updates on one channel do not affect other channels. The channels are defined below.

Core Channel

The Core channel controls all shared resources like OR ownership and Window ownership. It also controls all per-head state such as raster parameters and post-comp pipe settings. There is only one Core channel.

Core channel controls:

- Core Completion notifier
- Lockpins
- OR state
  - OR ownership
  - OR protocol
- Core Window state
  - Window Usage Bounds
  - Window Ownership
- Core Head state
  - Raster parameters

- Head Usage bounds
- Raster lock
- Output Scaler
- Cursor size and surface
- Output LUT, Output Color Space, Output Dynamic Range
- Dither / output color depth
- Head CRCs
- DSC settings
- 2Head1Or controls
- Chroma downsampling
- Head semaphores

Window Channel

The window channels control all of the per-window state other than window position. There is one window channel per window supported by the chip.

Window channel controls:

- Window Semaphores and Notifiers
- Composition controls (such as blend mode, depth)
- Window scaler
- Window size
- Window surface
- Input LUT
- Tone Mapping
- Surface format
- Presentation controls
- CSC0/CSC1 settings (matrix coefficients, LUT settings)

WindowImmediate Channel

The WindowImmediate channel controls the position of the window surface on the Desktop. This is a separate channel so that the window position can be changed even if the Window channel is blocked for some reason allowing a smoother user experience. The WindowImmediate channel is a pushbuffer channel. This permits software to queue up several updates simultaneously.

Cursor Channel

The Cursor channel defines the position of the cursor surface. The cursor surface itself is defined in the Core channel. Much like the Window Immediate channel, the purpose of this channel is to permit the cursor position to change even if the core channel is blocked or stalled for some reason.

The Cursor channel is the only channel not controlled by the pushbuffer. Register writes to the UDISP address range to program the Cursor channel that is a PIO channel. Writes to this range generates cursor methods within Display. No pushbuffer or FIFO is associated with this channel, so software must take care not to generate new methods when the hardware is still busy with the previous method. Display hardware reports a “Free” value to report if there is free space to generate another method.

Interlock / Fliplock

While normally the update methods on each channel are handled individually, Display provides the ability to interlock updates between (almost) any set of channels. With each update software can define the set of other channels that it wants to interlock. Display hardware waits for updates to be ready on all specified channels before permitting all of the updates to complete. The Window Immediate channels can only be interlocked with their associated Window channel. This is done to simplify the hardware implementation.

Fliplock is a special case of Interlock, which uses an internal or external lock pin. This imposes additional restrictions on the timing of the Update to ensure that flips happen on the same frame even across multiple heads. Fliplocked heads must be rasterlocked.

Register Ranges

The Display address range is 0x138EFFFF: 0x13800000 (no VGA range). All addresses in the following tables are zero-based, before adding the Display Base Address.

Table 7.78 Major Display Register Ranges

Name	Byte Address	Description
Non-VGA Display Address Range	0x000EFFFF:0x00000000	Host allocates 1 MB of address range for DISP. The first 64 K is reserved for VGA I/O accesses.
PDISP	0x0003FFFF:0x00000000	256 K
DISP Error Collator	0x00040FFF:0x00040000	4 K used for the Error Collator registers
Reserved	0x0005FFFF:0x00041000	124 K Unused for now
UDISP	0x000EFFFF:0x00060000	576 K Mostly user registers. Private state cache also defined in this range.

Table 7.79 Display Private Register Ranges

Name	Byte Address	Size	Description
NV_PDISP_FE	0x00005FFF:0x00000000	24 K 0x6000	Front End registers

Name	Byte Address	Size	Description
NV_PDISP_HEAD	0x00009FFF:0x00006000	16 K 0x4000	Head registers 2 K per head (0x800) Comp Head Post-comp Cursor Head TZ Head RG SF
Reserved	0x0000BFFF:0x0000A000		
NV_PDISP_SOR	0x0000FFFF:0x0000C000	16 K 0x4000	Per-SOR registers 2 K (0x800) per SOR
Reserved	0x00011FFF:0x00010000	8 k 0x2000	Reserved
NV_PDISP_WBOR	0x00012FFF:0x00012000	4 k 0x1000	Reserved
NV_PDISP_DSI	0x00013FFF:0x00013000	4 k 0x1000	DSI registers.
Reserved	0x000147FF:0x00014000	2 k 0x800	
NV_PDISP_HDCPRIF	0x0001487F:0x00014800	128 B	Key RAM controls
NV_PDISP_SEC	0x000148FF:0x00014880	128 B	Misc Security status/controls
NV_PDISP_UPSTREAM	0x000149FF:0x00014900	256 B	HDCP Upstream reporting
Reserved			
NV_PDISP_VGA	0x00015FFF:0x00015000	4 k 0x1000	dword aliases for all of the VGA registers
Reserved	0x00017FFF:0x00016000	8 k 0x2000	
NV_PDISP_IHUB	0x0001FFFF:0x00018000	32 K 0x8000	ISO Hub registers 512 B per window 512 B per cursor
NV_PDISP_WIN	0x0002FFFF:0x00020000	64 K 0x10000	Per-window registers 2 k per WIN (1K precomp, 1 K comp)
NV_PDISP_FE_SW	0x00030FFF:0x00030000	4 k 0x1000	Software read/write capability registers. Must be 4K aligned so that they can be mapped to DD.
Reserved	0x0003FFFF:0x00031000		
NV_PDISP_FE_EC	0x00040FFF:0x00040000	4 k 0x1000	Error Collator

Table 7.80 Display User Registers

Name	Byte Address	Size	Description
NV_UDISP_PRIVATE	0x0006FFFF:0x00060000	64 K	Assembly and Armed state cache for all private channels. (Name is confusing) 16 K: Precalc 1 K per WinPvt 1 K per WrbkPvt 1 K for Core Variables 1 K for Local variables
NV_UDISP_CORE	0x0007FFFF:0x00070000	64 K	Assembly and Armed state for the core channel.
NV_UDISP_REMAP	0x000EFFFF:0x00080000	448 K (7 64 K pages)	This is the full size of the remappable area. It should be aligned to a 256 KB boundary to make address decode and remapping easier. The non-remapped default order of channels already extends into the 0xC0000 range, so we need at least five 64 K pages
NV_UDISP_WIN0_ASY	0x000807FF:0x00080000	2 K	The assembly and armed state caches (including PUT, GET, and immediate methods) for each Window channel are mapped to a single 4K page.
NV_UDISP_WIN0_ARM	0x00080FFF:0x00080800	2 K	
(repeat 31 times for each window)			
NV_UDISP_WINIM0_ASY	0x000A07FF:0x000A0000	2 K	The assembly and armed state caches (including PUT and GET) for each Window Immediate channel are mapped to a single 4 K page.
NV_UDISP_WINIM0_ARM	0x000A0FFF:0x000A0800	2 K	
(repeat 31 times for each window)			
NV_UDISP_WRBK0_ASY	0x000C07FF:0x000C0000	2 K	The assembly and armed state caches (including PUT and GET) for each Writeback channel are mapped to a single 4 K page.
NV_UDISP_WRBK0_ARM	0x000C0FFF:0x000C0800	2 K	
(repeat 7 times for each writeback channel)			
NV_UDISP_CURS0_ASY	0x000C87FF:0x000C8000	2 K	The assembly and armed state for each cursor PIO channel is mapped into a single 4 K page.
NV_UDISP_CURS0_ARM	0x000C8FFF:0x000C8800	2 K	
(repeat 7 times for each cursor channel)			
Reserved	Reserved		Reserved
Reserved	0x000DFFFF:0x000D0000	64 K	Unused

Name	Byte Address	Size	Description
SF UDISP	0x000EFFFF:0x000E0000	64 K	This range is mapped directly to user space so that DD can control DisplayPort and HDMI Info frames. Only about 8 K of the range is used, but likely the whole page needs to be reserved so it can be mapped cleanly to user space.

7.5.1.5 Modes of Operation

7.5.1.5.1 Valid Operating Modes

Internally, the Display pipeline operates in one of three modes:

- **Sleep**: Display is idle and is not fetching any pixels or sending data to the Output Resource
- **Snooze**: The RG is actively generating a blank frame, but no pixels are being fetched from memory. LoadV is still generated every frame.
- **Awake**: Display is sending frames to the connected output resource.

There are three main controls, which affect the operating mode of Display:

- **OR Ownership**: If the head does not own an OR, it is asleep. As long as the head owns an OR, it is Awake or in Snooze.
- **Window Ownership**: If the head is not asleep and owns at least one window, the head is Awake. If the head is not asleep and does not own any windows, the head is in Snooze mode.
- **Core.Head.RunMode**: This method controls if display operates in continuous or OneShot mode. In Continuous mode, frames are generated automatically if the head is not asleep. In OneShot mode, frames are only generated one at a time as dictated by software.

A summary of the possible combinations is listed in the following table.

At least one OR owned by the head	Display Run Mode	At least one Window owned by the head	Mode
NO	---	----	Head ASLEEP, No pixels fetched
YES	Continuous	NO	Head in SNOOZE mode No pixels fetched. RG generates black frames. LoadV still generated each frame but only passed to the cursor pipeline.
YES	Continuous	YES	Head AWAKE. Pixels are being fetched and processed continuously.

At least one OR owned by the head	Display Run Mode	At least one Window owned by the head	Mode
YES	OneShot[1]	NO	Head in SNOOZE mode No pixels fetched. RG generates a single black frame and then stall at the end of vertical front porch. LoadV still generated each frame but only passed to the cursor pipeline. Software controls gate LoadV.
YES	OneShot[1]	YES	Head AWAKE Display fetches and scans out a single frame in response to each trigger event. RG stalls at the end of the vertical front porch and send Vblank pixels to the OR.

(1) Assumes that Stall Lock is also enabled.

Note: In the table above, owning a window does not require the window to be enabled. If all windows owned by a head are disabled and the Head owns an OR, the head is still awake. The compositor generates the Default Desktop Color for the active raster.

A head with DSC enabled in Snooze mode does not output a black frame. The output may be a gray frame, the DisplayPort Idle pattern, or the HDMI FRL Gap pattern.

7.5.1.5.2 OneShot mode

“OneShot mode” is the generic name for the Display state where frames are not generated automatically. Software manually releases each frame and the RG extends vblank until the next frame is released.

Two parameters control OneShot mode: 1) How the raster is stalled and 2) how the LoadV is stalled. These controls are separate to give software the flexibility to tailor the mode to their needs.

Raster Stalling

The Core.Head.SetStalllock method controls the raster stalling. Raster stalling can be enabled without LoadV blocking if desired.

Field	Description
Enable	Enables the stall lock feature
Mode	CONTINUOUS: The RG only stalls once and then switches to continuous scanout mode. This is typically used for exiting one-shot mode. ONE_SHOT: The RG stalls every frame.

Field	Description
LockPin	This defines whether an external lockpin is going to be used for the unstall event. If no lockpin is defined, the RG automatically unstalls when a LoadV is received. The LoadV indicates that a frame has been released upstream and pixels follow soon. If a lockpin is defined, the RG unstalls on the first lockpin edge after a LoadV arrives.
UnstallMode	While stalled, RG continuously repeats the last line of the vertical front porch. This is done to maintain lock with the panel. Upon an unstall, this mode selects whether RG finishes the current repeated line before unstalling or whether it jumps immediately to the start of VSYNC.
TEPolarity	Controls whether the external unstall signal is falling edge triggered or rising edge triggered. (unsupported)

After an unstall event, RG waits for NV_PDISP_RG_UNSTALL_SPOOLUP pixel clocks before starting the next frame. Then, Display time can complete spool up before scanout begins.

LoadV Stalling

In order to block the generation of new frames, LoadV is stalled. Specifically, Early LoadV is stalled. There are two mechanisms by which this can be accomplished: Using methods and using private registers.

When Core.Head.SetDisplayRate is set to OneShot, Display only releases one LoadV at a time. Software must push an Update method with the ReleaseElv flag set in order to release a LoadV for another frame. The ReleaseElv flag can be set on any channel associated with that head. Setting ReleaseElv on a Core Update releases an ELV on all active OneShot heads.

NV_PDISP_FE_ELV_BLOCK can also be used to block LoadV. The trigger NV_PDISP_FE_ELV_BLOCK_ALLOW_ONE_ELV is used to release a LoadV when this blocking mechanism is used.

When either of the LoadV blocking mechanisms are in place, RG Stall must be enabled in order to avoid underflows in the RG.

7.5.1.5.3 Passthrough Mode

All display submodules, which perform any pixel processing must support a passthrough mode, which bypasses all logic that could potentially change the value of a pixel. This includes features like scaling, composition, CSC, and the LUTs (which are also used for FP16 conversion). ISO Hub does not perform any pixel processing so it remains unaffected.

The purpose for this mode is to let Display pass an 8 bpc, 10 bpc, or 12 bpc RGB surface directly from the input to the output for software-implemented color formats. G-SYNC panels have a mode, which can update the firmware based on transmitted pixel values, so being able to pass data

unchanged is important. (Passthrough for 12 bpc content requires the use of the A16R16G10B16 input format.)

7.5.1.5.4 Power Gating

Display has a separate PD_disp power domain for all Display logic.

Full Display Power Gating

All of Display can be power gated if Display is floorswept or otherwise unused. Power Gating clamps are added on all external interfaces to the display modules. Only display logic must be present in the display partitions so that they can be power gated. If Display is dynamically power gated, software must reinitialize Display on power up. This is the same procedure as boot. There is no context save and restore hardware.

Asserting Display Reset

1. All display clocks must be running so that reset can propagate:
   a. rg*_pclk
   b. sf*_clk
   c. sor*_clk
   d. axi2apb_clk, dbb_disp_clk
   e. dispclk
   f. clk_2disp_hda_bclk (aza_bitclk)
   g. dsc_clk
   h. hubclk
   i. dsi_core_clk
   j. dsi_rx_clk
   k. dsi_pixel_clk
   l. dsi_lp_clk
   m. link*_afifo_clk
   n. link*_fast_clk
   o. xtal
2. Shut down all heads, which put all heads in the SLEEP state.
   a. Set Core.Sor.SetControl.OwnerMask to NONE.
   b. Set Core.DSI.SetControl.OwnerMask to NONE.
   c. Set Core.Wbor.SetControl.OwnerMask to NONE.
   d. Core Update.
   e. Wait for Core Update to complete.
   f. (It may be necessary to use the accelerator bits in NV_PDISP_FE_CHNCTL_<chn> to force a channel to make progress if it is stuck.)

3. If you cannot shut down a certain head or SOR even with the accelerators; force those resources to use safe clock. This step must only be done if the previous step did not shut down one of the heads.
   a. NV_PDISP_FE_CMGR_CLK_RG_FORCE_SAFE = _ENABLE
   b. NV_PDISP_FE_CMGR_CLK_SF_SAFE_CTRL = _SAFE
   c. NV_PDISP_FE_CMGR_CLK_SOR_MODE_BYPASS = _DP_SAFE
4. Deallocate all Display channels:
   a. NV_PDISP_FE_CHNCTL_<chn>_ALLOCATION = _DISCONNECT
   b. NV_PDISP_FE_CHNCTL_<chn>_ALLOCATION = _DEALLOCATE
   c. Poll for NV_PDISP_FE_CHNSTATUS_<chn>_STATE = _DEALLOC
   d. Poll for NV_PDISP_FE_CHNSTATUS_<chn>_WRITE_PENDING = _NO
5. Clear HDA Presence Detect:
   a. NV_HDACODEC_AUDIO_HDA_PRESENCE_PW*_PD = NOT_PRESENT
   b. This step should be done whenever Display resets independently of HDA (e.g. if FN0 resets but FN1 does not).
6. Clear all interrupts:
   a. Write 0xFFFFFFFF to all NV_PDISP_FE_EVT_STAT* registers.
7. Hot reset the Display DBB interface to flush out any pending transactions:
   a. MC_CLIENT_HOTRESET_CTRL_1_0_NVDISPLAY_FLUSH_ENABLE = _ENABLE
   b. Poll for MC_CLIENT_HOTRESET_STATUS_1_0_NVDISPLAY_HOTRESET_STATUS == FLUSH_DONE
8. Fence CBB and DBB:
   a. PMC_IMPL_PART_DISP_PWRDWN_REQ_CONTROL_0_PWRDWN = PWRDWN
   b. Poll PMC_IMPL_PART_DISP_PWRDWN_REQ_STATUS_0 until H1X_ACK, DBB_ACK, and CBB_ACK report PWRDWN.
9. Apply output clamps:
   a. PMC_IMPL_PART_DISP_CLAMP_CONTROL_0_CLAMP = ON
   b. Delay at least 100 ns to ensure that clamps are engaged.
10. Assert reset:
    a. The following resets should be asserted:
       i. NVDISPLAY0..CLK_RST_CONTROLLER_RST_DEV_NVDISPLAY_0.SWR_NVDISPLAY_RST
       ii. NVDISPLAY0..CLK_RST_CONTROLLER_RST_DEV_NVDISPLAY_0.SWR_NVDISPLAY_DSI_CORE_RST
11. After reset is asserted, the clocks branches mentioned in Step 1 can be stopped.
12. Before continuing to gate power, ensure that there has been at least 3.7 µs since the last ungating event.
13. SRAM Sleep and power gating enable.
    a. Write PMC_IMPL_PART_DISP_POWER_GATE_CONTROL_0 with the following settings:
       i. LOGIC_SLEEP = ON

   ii. SRAM_SLEEP = ON
   iii. SRAM_RET = OFF
   iv. INTER_PART_DELAY_EN = ENABLE
   v. START = 1
  b. Poll PMC_IMPL_PART_DISP_POWER_GATE_CONTROL_0 until START == 0.
  c. Read PMC_IMPL_PART_DISP_POWER_GATE_STATUS_0 to ensure the correct state:
   i. LOGIC_SLEEP_STS == ON
   ii. SRAM_SLEEP_STS == ON
   iii. SRAM_RET_STS == OFF

De-asserting Display Reset

1. Wait until at least 3.7 µs have passed after the last entry into the power gating state.
2. Disable SRAM Sleep and logic sleep power gating.
  a. Write PMC_IMPL_PART_DISP_POWER_GATE_CONTROL_0 with the following settings:
   i. LOGIC_SLEEP = OFF
   ii. SRAM_SLEEP = OFF
   iii. SRAM_RET = OFF
   iv. INTER_PART_DELAY_EN = ENABLE
   v. START = 1
  b. Poll PMC_IMPL_PART_DISP_POWER_GATE_CONTROL_0 until START == 0.
  c. Read PMC_IMPL_PART_DISP_POWER_GATE_STATUS_0 to ensure the correct state:
   i. LOGIC_SLEEP_STS == OFF
   ii. SRAM_SLEEP_STS == OFF
   iii. SRAM_RET_STS == OFF
3. The following clocks must be running for 2 µs before de-asserting reset.
  a. These clocks will be forcibly enabled when the system reset is asserted. If Display is reset independently of the rest of the chip, then software needs to enable these clocks manually.
  b. rg*_pclk
  c. sf*_clk
  d. sor*_clk
  e. axi2apb_clk, dbb_disp_clk
  f. dispclk
  g. clk_2disp_hda_bclk (aza_bitclk)
  h. dsc_clk
  i. hubclk
  j. dsi_core_clk
  k. dsi_rx_clk
  l. dsi_pixel_clk

   m. dsi_lp_clk
   n. link*_afifo_clk
   o. link*_fast_clk
   p. xtal
4. If necessary, re-assert the reset signal as described in Step 10 above.
   a. Some APIs that enable clocks automatically de-assert reset.
5. Wait 2 µs for reset to propagate.
6. Disable output clamps:
   a. PMC_IMPL_PART_DISP_CLAMP_CONTROL_0_CLAMP = OFF
7. De-assert reset:
   a. NVDISPLAY0..CLK_RST_CONTROLLER_RST_DEV_NVDISPLAY_0.SWR_NVDISPLAY_RST
   b. NVDISPLAY0..CLK_RST_CONTROLLER_RST_DEV_NVDISPLAY_0.SWR_NVDISPLAY_DSI_CORE_RST
8. Unfence CBB/DBB/Host Controller:
   a. PMC_IMPL_PART_DISP_PWRDWN_REQ_CONTROL_0_PWRDWN = PWRUP
   b. Poll PMC_IMPL_PART_DISP_PWRDWN_REQ_STATUS_0 until H1X_ACK, DBB_ACK, and CBB_ACK report PWRUP.
9. Take DBB out of flush mode:
   a. MC_CLIENT_HOTRESET_CTRL_1_0_NVDISPLAY_FLUSH_ENABLE = _DISABLE
10. Initialize display RAMs:
   a. NV_PDISP_FE_RAM_INIT_PUSHBUF -->TRIGGER
   b. NV_PDISP_FE_RAM_INIT_STATE_CACHE -->TRIGGER
   c. NV_PDISP_PRECOMP_WIN_RAM_INIT(*)_SCALER_RTR -->TRIGGER
   d. Poll for the trigger bits to clear before continuing
11. KFUSE read:
   a. NV_PDISP_HDCPRIF_CTRL_FUSE_VALID -->YES to trigger Display to start reading the KFUSE
   b. Optional: Poll for NV_PDISP_HDCPRIF_STATUS_KEY_READY == _YES

Impact on Power Gating to Clocks

Unused PLLs should be placed in IDDQ mode by software to save power.

7.5.2 Functional Description

The functionality of the Display Controller is described in the following sections based on its functional partition for pixel processing, Front End (FE), Isochronous Memory Hub (ISO Hub), Precomposition (Precomp), Composition (Comp), Post-composition (Post-comp), and Raster Generation (RG).

7.5.2.1 Front End Host Interface

The Display Front End receives control instructions for pixel processing from the Host Controller by managing the following:

- host interface
- registers
- interrupts
- LoadV
- startFetch,
- pushbuffers
- state cache
- error checks
- precalc

No pixel processing is performed in the FE.

The Display Front End is partitioned into five major subunits.

- Method Fetch (MF)
- Method Processor (MP)
- State Manager (SM)
- Real Time (RT)
- Arbitration Layer (AL)

7.5.2.1.1 Isochronous Memory Hub (ISO Hub)

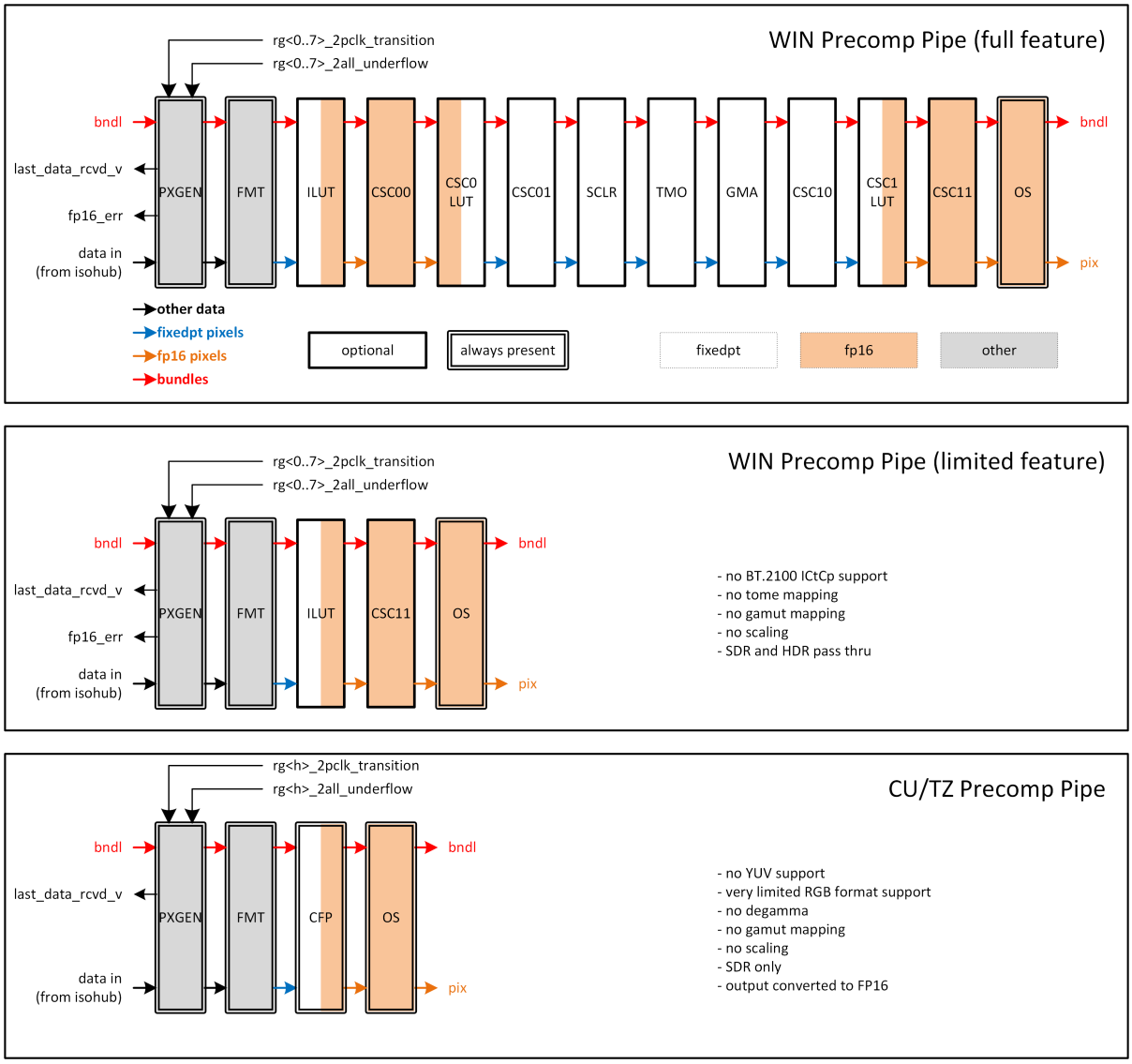
The ISO Hub is the combined interface between the ISO read path and the rest of the Display pipeline. Requests made on the display read path have a higher priority because display requires isochronous data. The purpose of the ISO Hub is to generate all of the read requests to FB for all display fetch surfaces. It must arbitrate the requests in such a way as to prevent underflows on any head. The ISO Hub contains a large shared memory pool to buffer read return data. This is used to avoid refetching of pixels for block-linear surfaces, absorbs some amount of FB latency variation, and other long latency events like DVFS. The ISO Hub does not change the values of any pixel. It needs to perform some reordering of the data (including rotation) so that the pixels can be fed into the precomp pipe in the correct order.

7.5.2.1.2 Pre-Composition Pipe (Precomp)

The Precomp pipe performs all pixel processing that must be done before composition. The Precomp formats pixels to the internal pipe color format, apply the input LUT, tone mapping, and several 3 x 4 matrix transformations, and perform scaling. A precomp pipe is tied to a specific

Window channel and is mostly controlled through that channel. Each precomp pipe operates independently of the other pipelines.

Figure 7.25 Pre-Composition Pipe

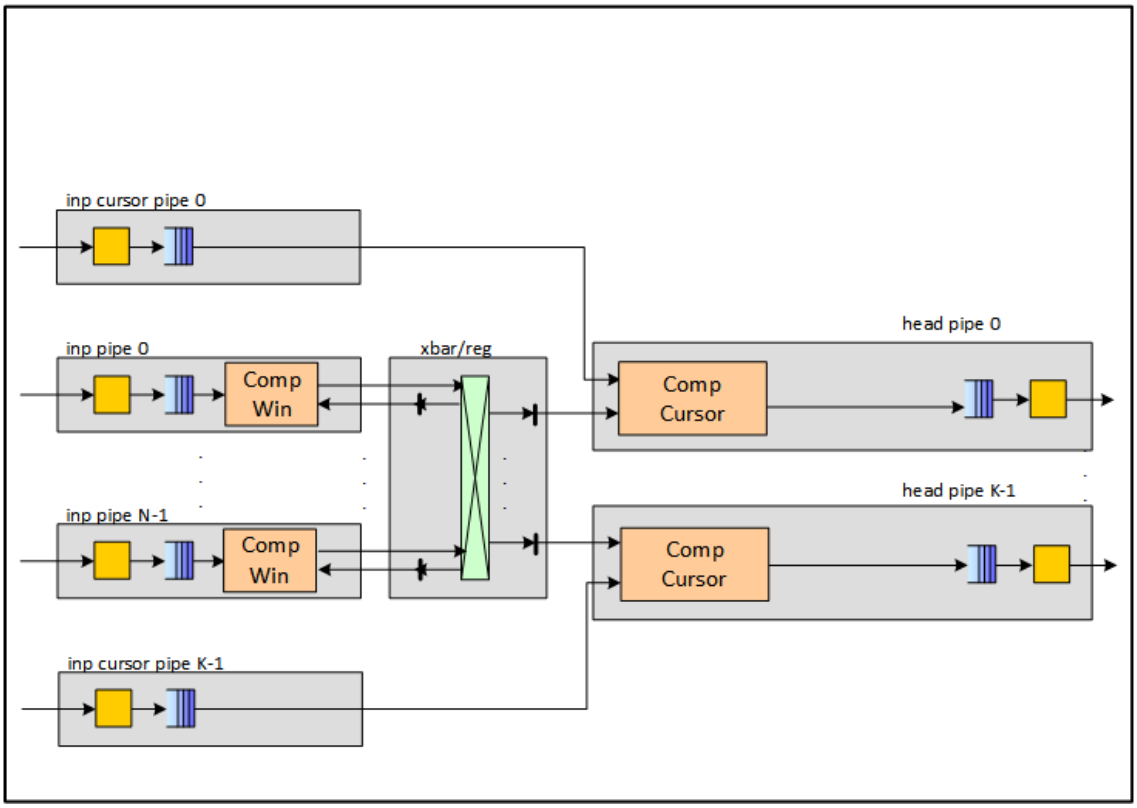


7.5.2.1.3 Composition and Crossbar

The compositor takes two input surfaces and produces one composited output surface. There is one compositor at the end of each Precomp pipe. Each of these compositors accepts pixels from the local Precomp pipe and the output of a compositor from another pipeline (or a programmable background color). Pixels are merged using alpha blending or color keying. A full cross-bar exists to

route pixels from any compositor to any other compositor. The output to the head pipe can come from any compositor. The Cursor Compositor is located in the head's pipeline after the XBAR. The cursor is always composited on top of all windows. While each window and cursor contains a single compositor, the collection of all compositors and the Crossbar is often referred to as "comp".

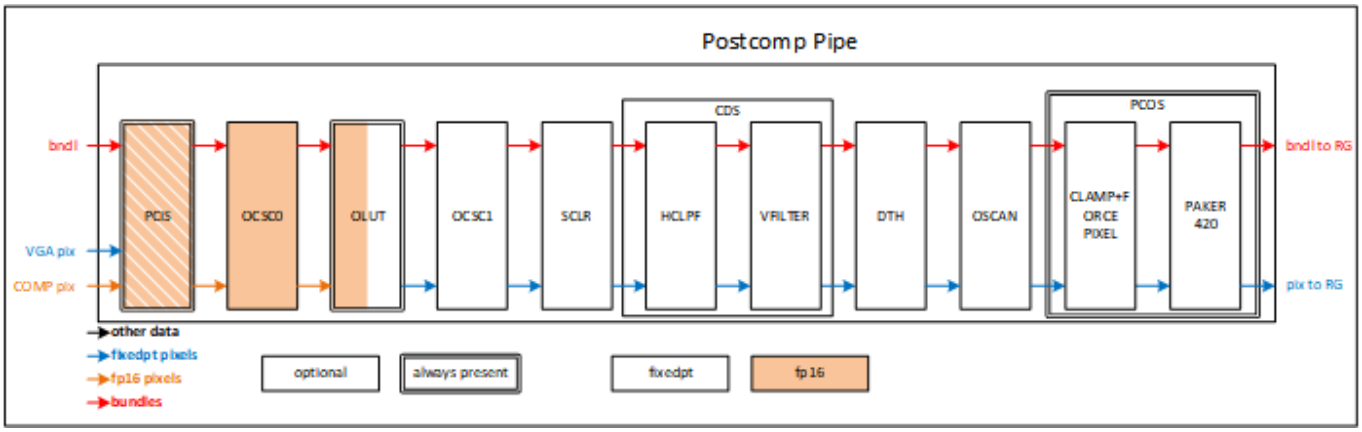
Figure 7.26 Composition Pipe



7.5.2.1.4 Post-Composition Pipe

The Post-comp pipe contains all stages of pixel processing between the Compositor XBAR and the Raster Generator. This includes the output LUT, output scaler, dither, DSC, and the output color-space converter.
There is one Post-comp pipe per head controlled by the Core Channel.

Figure 7.27 Post-Comp Pipe



PCIS: Input Stage. Handles interface with compositor and VGA

OCSC0: 3 x 4 CSC matrix. Can be used for Hue/Sat adjustment or conversion from RGB to LMS

OLUT: Output LUT. Applies the appropriate OETF/gamma for the panel

OCSC1: 3 x 4 CSC matrix. Used to convert to the final output color space

SCLR: Post-composition Scaler (Post-comp does not include a scaler)

HCLPF: Horizontal Chroma Low Pass Filter. Performs filtering on the chroma component when the output is 4:2:2 or 4:2:0

VFILTER: Vertical Chroma Low Pass Filter. Performs filtering on the chroma component when the output is 4:2:0

DTH: Dither from pipeline color depth to the panel's color depth

OSCAN: Overscan. Adds border pixels with the output Viewport is smaller than the active region

PCOS: post-comp Output Stage. Includes Regional CRC, 4:2:0 packer, and output clamping

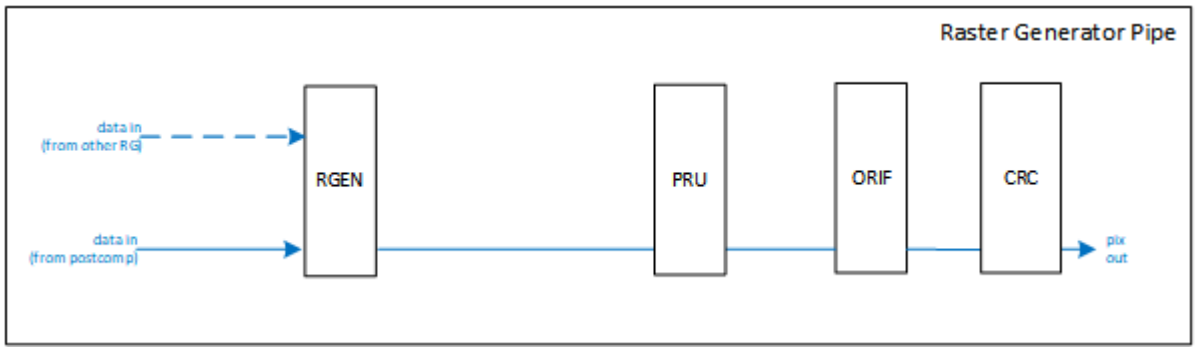
DSC: Display Stream Compression (not shown). Compresses the final pixel stream. This occurs between Post-comp and RG.

7.5.2.1.5 Raster Generator (RG)

The raster generator adds blanking, syncs, and other timing information to the raw pixels from the Post-comp pipe. The raster generator must generate one pixel per pixel clock to the output resources. The raster generator is also responsible for performing raster lock and some protocol-specific formatting.

For the 2head1Or feature, two RGs are ganged together to generate a single output stream to the SF and SOR. The RG pairing is fixed: RG1 can pass pixels to RG0.

Figure 7.28 Raster Generation Pipe



RGEN: Raster Generator. All of the raster timing (blanks, syncs) are applied here. Responsible for raster lock. For 2head1Or, RGEN for head0 can accept pixels from head1's RG.

PRU: Pixel Reorder Unit. When driving Dual-MST or Dual SST, this reorders the pixels so that the left-half pixels are sent to one stream and the right-half pixels are sent to another stream.

ORIF: Output Resource Interface. Output resource-specific formatting.

CRC: Calculates RG CRC.

7.5.2.1.6 Serial Formatter (SF)

The SF is responsible for half of the DisplayPort link layer protocol and the HDMI protocol encoding. Audio data is inserted into the blanking region at this point in the pipe. The remaining DisplayPort, TMDS, and HDMI FRL formatting are performed in the SOR. The responsibilities are split so that multiple heads can be used to drive a single SOR for DisplayPort Multistream. The SF performs some minor formatting (mainly identifying line type) when the output is DSI.

7.5.2.1.7 Serial Output Resource (SOR)

The SOR performs the remaining pixel encoding for TMDS (HDMI), HDMI FRL, and DisplayPort. HDCP is performed for both HDMI and DisplayPort in the SOR. It produces the final parallel data to the analog macro, which drives signals to the panel. The SOR is also responsible for panel power sequencing and all associated SOR macro control signals.

7.5.2.1.8 DSI

The DSI acts as an output resource similar to the SOR. It accepts pixel data from the SF and performs the encoding necessary to send the data over a DSI CPHY or DPHY link. It produces the final parallel data to be sent to the analog macro.

7.5.2.1.9 Display SEC

DISP_SEC is a centralized module that oversees the security of many of Display's subunits. DISP_SEC manages VPR policy, secure keys, upstream reporting, and the Secure Bus. Some subunits report a panic condition to DISP_SEC when they detect a condition that may potentially leak protected content. DISP_SEC can instruct the pipeline to blank the output data when this occurs.

7.5.2.1.10 CRC Collection

This mechanism is used by NVIDIA to verify display hardware functionality in testing, by comparing per-frame CRCs between RTL and a model. This is separate from, and different to, the regional CRC mechanism used for functional safety checks.

CRCs are collected in the RG, Compositor, and ORs. FE reads the subunits' CRCs over the bundle bus after they assert crc_valid. The results are written to a notifier structure in memory. The test infrastructure reads the notifier and writes out a test.crc file. This is compared against the golden CRC file.

CRC Register Interface

All units, which calculate CRCs should expose these common register fields:

Table 7.81 CRCA Register Fields

Field	Size (bits)	Description
VALID	1	Reports whether the current CRC is valid. Cleared when CRCB is read.
ERROR	1	A new CRC was generated while Valid was still high from the previous CRC. A CRC value was lost. This bit is cleared upon reading CRCA.
VPR_STATUS	1	Indicate whether the CRC in CRCB register was derived from at least one VPR pixel.

Table 7.82 CRCB Register Fields

Field	Size (bits)	Description
CRC	32	Reading the CRC clears the Valid bit in CRCA This register is protected using PLM to prevent insecure software from reading CRCs based on VPR pixels.

CRC capture begins at LoadV and ends at the end of the frame. Based on the value of Core.Head.SetControlOutputResource.CrcMode, the CRC engines capture all pixels, only active

pixels, or only nonactive pixels. In “active only” mode, the CRC calculation ends at lasth&&lastv. In the other modes, CRC calculation ends at LoadV (including the LoadV pixel for units after the RG). In OneShot mode, only the “active only” mode is permitted because the vblank period is of unpredictable size in this mode. CRCs registers can only be read with private level 3 to protect VPR information.

CRC Sequence of Events

1. Software defines a valid context DMA for the CRC notifier in SetContextDmaCrc.
2. At the start of the next frame, Comp, RG, and the ORs start calculating CRCs.
3. When CRCs are ready (after LoadV), the subunits assert crc_valid to FE.
4. FE reads CRCB and stores the value.
5. FE reads CRCA to determine if there was any error.
6. Once Comp, RG, and the OR CRCs are collected for a given frame, FE writes the entry to the CRC notifier.
7. Repeat from step 2 until the CRC notifier ContextDMA is changed.

Error Conditions

If a subunit calculates a new CRC while the valid bit is still set for the previous CRC, it should set the CRCA_ERROR bit and not update the CRC value in CRCB. If FE sees that the ERROR bit is set when reading CRCA, it should log the error and stop collecting CRCs. Since the number of CRCs collected from each subunit may not match, the CRC collection could potentially hang if it tries to continue from this point.

7.5.2.1.11 VPR

The Video Protected Region is a section of memory, which can be protected against reads or writes from various clients. When Display reads data from a VPR region, it must ensure that the data (including CRCs based on the pixel data) is never leaked to an insecure output.

7.5.2.2 Clocks

Table 7.83 Clock Summary

Clock Name	Maximum frequency (MHz)	Source	Usage
Interface clocks			
sysclk_noeg		SYSPLL	ISO Interface, NISO Interface
axi2abp_clk	408		Used for the register interface
dbb_disp_clk	533		Used for the ISO and NISO interfaces

Clock Name	Maximum frequency (MHz)	Source	Usage
Jtag_reg_clk	100	jtag_tck	SOR IOBIST
Pipeline clocks			
dispclk	1190	SPPLL, DISPPLL	Main clock for FE, precomp, comp, post-comp. The end of the ISO Hub output pipeline uses dispclk. Input of RG is on dispclk.
dsc_clk	444	dispclk/3	The DSC logic operates at three pixels per clock, so a clock that is 1/3rd the rest of the pipeline is required.
hubclk	385.7	DISPHUBPLL, SPLL	Main ISO Hub clock
rg*_pclk	1190	VPLL, xtal, PLLD	Pixel clock for the RGs. One per head.
SOR clocks			
sf*_clk	1000	sor*_clk, xtal, PLLD	Serial Formatter clock. For DisplayPort and HDMI, this uses the clock of the SOR that the SF is attached to. For DSI, this uses pclk of the head.
sor*_clk	1000	xtal, VPLLs, macro feedback clocks	SOR clock. This is typically 1/10th of the link frequency. Link pad macro generates this. For HDMI FRL, it is 1/12th the link frequency.
sor*_ifp*_clk	600	Macro feedback clocks	Used for one side of the SOR Async FIFOs connecting the SOR core logic to the “fast” clocks near the macro. Runs at 1/20th the link frequency.
link*_fast_clk	600	Macro feedback clocks	Used in the “split” modules, which implement the SOR Link XBAR muxing.
link*_macro_clk	600	Macro feedback clocks	Connected to the SLOWCLK port of the Link pad macros. Should be identical to link*_clk_fast.
SOR macro pclk reference Sor_pll_ref_clk	340	VPLLs	The pixel clock reference source for the Link pad macros. Connects to the PCLK port.
SOR macro DisplayPort reference	270	SPPLL0, SPPLL1	The DisplayPort and HDMI FRL reference clock for the Link pad macros. Connects to the DCLK ports. May be downspread for DisplayPort.
DSI clocks			
dsi_core_clk	357	DSI PHY	DSI core logic. Typically runs at I/O clk divided by 8 or 7
dsi_pixel_clk	500	DSI PHY, VPLL, PLLD	Matches the pixel clock of the head.
dsi_io_clk	2500	PLLD	High-speed I/O clock of DSI pad
Audio clocks			

Clock Name	Maximum frequency (MHz)	Source	Usage
aza_bitclk	54	SPPLL0	Main clock for the HDA codec logic and some of the AZA controller logic.
aza_2xbitclk	108	SPPLL0	Used in AZA controller
maudclk	270	SPPLL0	Used during the MAUD calculation in the HDA codec. A fast enough clock is needed to drive DDA counters to emulate the 512*fs clock.
Other			
xtal	38.4	Crystal reference clock	Used by FE to generate microsecond and millisecond reference pulses for use within Display and PMGR.

Display gates clocks to submodules, which are idle. Top-level clock enables are controlled in NV Clocks. Software must ensure that dispclk, hubclk, rg_pclk, sf_clk, sor_clk, and dsi_core_clk are enabled before programming display. The bundle bus requires some clock in order to pass bundles. Safe-clk (xtal) is fine for this purpose, but a higher clock may be desired to reduce programming time. Software must also ensure that all Display clocks are running when Display Reset is asserted. This requirement is new to NVD4.0 due to the transition to a sync-reset architecture.

7.5.2.2.1 Dispclk and Hubclk Source Switching

The dispclk, hubclk, or both may need to change when there is a change to the display bandwidth requirement. The change in bandwidth could be due to the addition/removal of a new display or changing the configuration of current display (like adding/removing windows, scaling). When dispclk and hubclk are changed while display is active, the pixel flow to raster generator must be uninterrupted. Also, the adequate pixels must stored in various buffers to avoid display from underflowing.

This section goes over current dispclk/hubclk policy and buffering requirements when these clocks switch. From the Display perspective, clocks should change dynamically without any dead cycles. While the PLLs (DISPPLL and HUBPLL) support this feature (called dynamic ramp), there are discrete ranges of frequencies where this is possible to change without any dead cycle time. The VCO operating limits determine this. When the frequency change crosses from one range to another different source or a temporary clock source is chosen while the PLL (PDIV) is changed to hit the new frequency range. After the clock is stable, we switch back to DISPPLL/HUBPLL. For every clock source switch, a dead cycle time of five clock cycles of source clock and five clock cycles of destination clock is incurred. Based on the clocking policy, there can be up to two switches per frame. The following is an example of various ranges achieved for PDIV values.

Table 7.84 DISPPLL Output Ranges

PDIV	DISPCLK RANGE
1	[806.4, 1612.8]

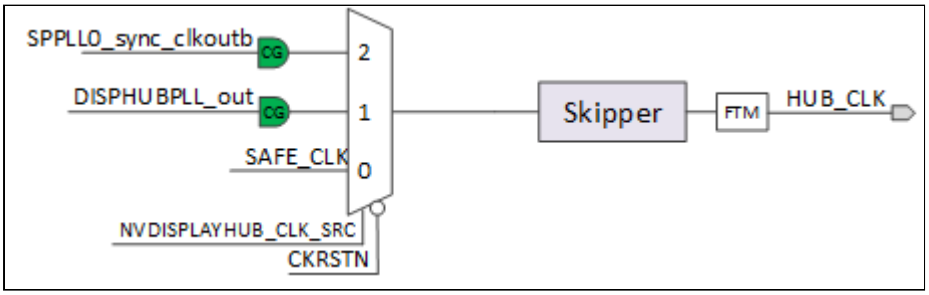
PDIV	DISPCLK RANGE
2	[403.2, 825.6]
3	[268.8, 550.4]

With HUBPLL/DISPPLL, if a fractional divider is used with dynamic ramp then the clock may undershoot by up to 5% for several microseconds before settling to the final frequency. Since hiding such large duration requires significant buffering, there are two options. One is for software always to allocate clocks 5% faster than minimum required and another is only to use a linear divider with a dynamic ramp. Software will go with the latter.

7.5.2.2.2 Hubclk

Clocking architecture does permit flexibility to use HUBPLL or a divided version of SPPLL. For the use cases of interest, we feel the level of granularity we get by generating from SPPLL is good enough and also helps save HUBPLL power. The current software policy is to have hubclk always derived from SPPLL by changing the PDIV. SPPLL is an always-running PLL at 2,700 MHz and for hubclk we would be using PDIV in [7,63]. As per PLL designers, whenever we change the PDIV value in [3,63] there will not be any dead cycles. Therefore, additional buffering due to hubclk changes will not occur.

Figure 7.29 hubclk tree



7.5.2.2.3 Dispclk

Similar to HUBCLK, SPPLL or DISPPLL can drive DISPCLK. Unlike hubclk, we cannot always generate dispclk from SPPLL for frequencies of interest. In order to simplify software programming in NVIDIA Ampere Architecture, it was decided that if dispclk<=900 then we use SPPLL with a divider (PDIV) and for other cases we will use DISPPLL. Following the same strategy as NVIDIA Ampere Architecture implies that there is clock switch per frequency change. However, for some use cases that require dispclk in the 300 MHz, 500 MHz range, the granularity that we get from SPPPLL is too coarse. It can drive the minimum voltage of the chip higher. In some cases, this was as high as 40 mV.

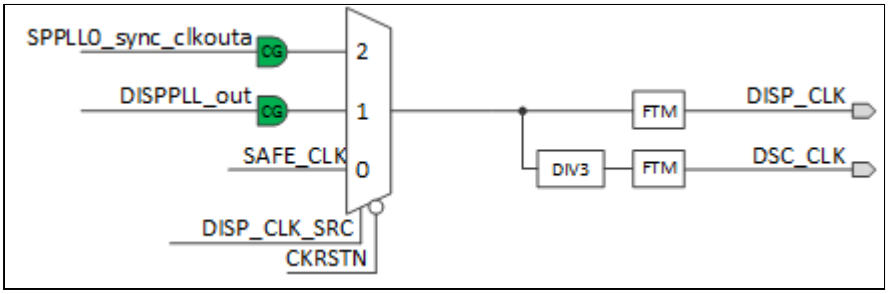
To avoid this increase in voltage, one must derive dispclk from DISPPLL. Based on the current and target dispclk frequency and taking into account PLL VCO requirements, this would imply that apart from changing the PLL settings MDIV and NDIV that permits dynamic ramp, the PDIV must be changed as well. If we change PDIV first and then program the DISPPLL, then we may run DISPCLK slower than expected frequency for a period of time (leading to underflow). If we change DISPPLL first and then program PDIV, then we may run clock much faster, violating voltage requirements.

To switch the DISPPLL PDIV, the following sequence must be followed:

1. Determine closest SPPLL linear divide frequency that is greater than or equal to the current dispclk frequency.
2. If this frequency requires a higher chip voltage, then raise the voltage.
3. Program the SPPLL divider for dispclk.
4. Switch dispclk to the output of the SPPLL.
5. Program DISPPLL to the required frequency.
6. Switch dispclk to the output of DISPPLL.
7. Reduce the voltage if possible.

This sequence has two clock switches that will likely occur within a single frame. Display requires buffering to handle the dead cycles introduced by these switches. Software uses the NVIDIA Ampere Architecture approach (use SPPLL for any frequency <= 900 MHz), but has the flexibility of using a more efficient DISPPLL approach at a later point if product requires it. Therefore, the display design has been updated to buffer for two clock switches.

Figure 7.30 Dispclk tree



7.5.2.3 Coordinate Systems

To understand the coordinate systems used by display and the associated settings, refer to the following text and diagram.

There are three coordinate systems in display:

1. Input: used to define regions inside surfaces as they are placed in memory
2. Composition (desktop): used to define how windows are composed on the screen/desktop

3. Output (or raster): for the actual output raster produced

The main processing steps that affect how to translate between these coordinate systems are:

- Scaling: input domain is not scaled. The composition domain is scaled by window scalers.
- Cropping: windows may be partially off screen (as defined in composition domain) which leads to cropped rectangles defining the visible portion. Those then have to be inverse-scaled to the input domain to find the corresponding source rectangle in the input domain.

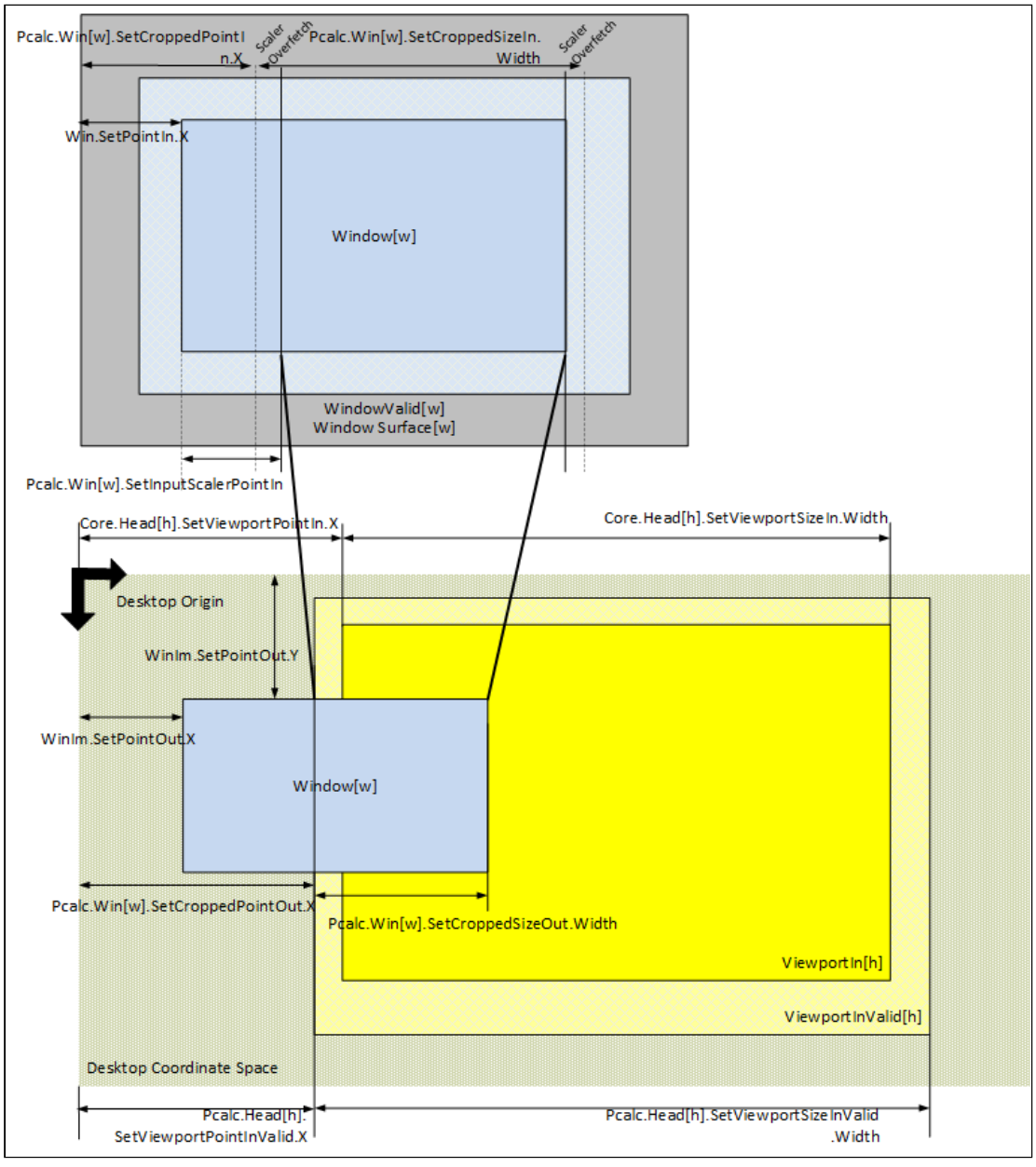
For window settings, only the input and composition domains are of interest. Since does not have an output scaler, the output domain and composition domain are actually the same.

A viewport is a rectangle in desktop coordinate space that defines the portion of the desktop that is visible on a given head. By moving the viewport position, you can pan the head across the entire desktop space.

A rectangle in desktop coordinate space defines the window size and position. The portion of the window that intersects with a viewport is visible on the corresponding Head, the rest is off-screen. Note that windows are not relative to viewports. Moving a viewport does not move the windows with it.

The diagram shows an active raster (= viewport) in yellow, and a window (blue). The window has a surface associated with it (gray).

Figure 7.31 Active Raster



The contents of the full window can be a rectangular subset of the full surface. The portion of the full window that is inside the viewport and thus visible is called the cropped window. Some relevant window settings are:

SetPointIn.X/Y:

This is the pixel/line offset into the window buffer of the top-left pixel of the full window (before cropping).

SetPointIn always defines the top-left corner of the window in the buffer, regardless of scan out direction. Display hardware calculates the address of the corner where fetching must begin.

SetSizeIn.Width/Height:

This is the size of the full window, cropping and scaling, i.e. the size of the portion of the image buffer that is needed to draw the contents of the full window.

SetPointOut.X/Y:

This is the position of the full window relative to the desktop origin. If the viewport is placed at (0,0), which is typically done, then this is the same as window position relative to the raster origin. These values can be negative.

SetSizeOut.Width/Height:

This is the size of the full window in the desktop domain after scaling, but before cropping.

For settings related to windows and window scaling, all those with "In" their names are in input domain, and all those with "Out" in their name are in desktop domain.

Most of these settings have a corresponding cropped version, which reflects the values associated with the cropped window instead of the full window. These cropped settings are the ones that hardware actually needs.

7.5.2.3.1 Window

A window is a region of the viewport that contains the contents of a surface (or multiple surfaces in the case of planar) from the FB. The window channel controls surface parameters. This includes states such as the buffer offset, input scaler parameters, and color format.

Windows can be assigned to any Head. A window channel controls each window.

There is one window channel per window supported by the class.

Window Dimensions

As a general rule, input parameters are defined in the buffer's coordinate space that is relative to the first pixel defined by the window surface context DMA. Output parameters are defined in the composition coordinate space.

Note:

- For YUV windows, the fetch rectangle must begin at an even location in the surface and have an even size. This applies to all YUV formats, and both directions, regardless of the actual chroma subsampling. So it even applies to YUV444 surfaces.

The final fetch rectangle is derived by the Precalc process. The public settings must be chosen or restricted such that the derived private settings for the fetch rectangle meet these "even" requirements for YUV windows, otherwise display does not operate correctly.

Window Ownership

The window has an associated ownership state, which can have one of the following permitted values:

- none
- Head<i>; i=0..H-1

The ownership is a double buffered value. The actual current ownership is the active state value. The new (next) ownership is the assembly state value. Not all transitions are legal. A window ownership can be changed to none, or from none at all times. It can never be changed from Head<i> to Head<k> directly for any valid i and k with i≠k. A change of ownership always has to transition through the 'none' state.

The ownership state of a window is part of the window's channel, and as such also falls under the access restrictions for that window (more later).

If additional setup is needed related to this change in ownership, for example in ISO Hub shared resource control, such changes have to be made by driver. It must be updated/activated either at the same time or in some appropriate sequence to avoid resource problems during or after the completion of the change.

7.5.3 Display Controller Registers

Refer to "Reading Register Tables" in the Introduction chapter for the register table protocol as well as recommendations for accessing registers.

NV_PDISP_FE_CMGR_CLK_SOR

This register set controls the clock that drives the SORs. The RM loads the registers with the proper required divide factors. FE enables the clocks and selects between SAFE and NORMAL modes. The status of the FE selections is reported below for debug assist. There are two clocks to each SOR: one for the logic, the other for the SOR hi-speed PLL. The hi-speed PLL source is always enabled, and always in NORMAL mode.

This is an array of 8 identical register entries; the register fields below apply to each entry. Full register list is: NV_PDISP_FE_CMGR_CLK_SOR_<i>, among which <i> belongs to <0..7>.

Offset: 0x2300,..,0x5b00
Read/Write: See table below
Parity Protection: N
Reset: 0x00X2X0X0 (0bxxxx,xx00,x000,0010,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
25:24	RW	0x0	**CLK_SOURCE**: The DP_DUAL_TMDS macro has four input clocks. This field selects which clock is used for the internal logic. When the SOR is operating in DP mode, this should be set to one of the _DPCLK settings. When the SOR is operating in TMDS mode, this should be set to one of the _PCLK settings. This field should only be changed when the SOR is asleep or the DP_MODE_BYPASS is set to DP_SAFE. Allow 200 microseconds for the plls in the analog macro to settle after changing this setting. **0** = INIT **0** = SINGLE_PCLK: Single ended pclk from the vpll **1** = DIFF_PCLK: Differential Pclk **2** = SINGLE_DPCLK: Single ended 270 MHz clock **3** = DIFF_DPCLK: Differential 270 MHz clock
23	RO	X	**STATE**: Indicates whether FE has ENABLEd or DISABLEd the clock to the SOR. Note, this state is independent of the choice of NORMAL or SAFE clock reported. **0** = DISABLE **1** = ENABLE

Bit	R/W	Reset	Description
22:18	RW	0x0	**LINK_SPEED**: Programs the link speed for DP, TMDS, FRL i.e. it selects the multiplier used by the analog macro PLL. For DP and TMDS, logic in the SOR will receive a version of this clock that is divided by 10. For FRL, logic in the SOR will receive a version of this clock that is divided by 12. This field should only be changed when MODE_BYPASS is set to DP/FRL_SAFE or when the SOR is in safe mode. It may take up to 200 microseconds for the PLLs in the analog macro to settle after this setting is changed. Changing this field will require DP/FRL link re-training. **0** = DP_1_62GHZ: Selects the 1.62 GHz link clock; PLL generates 1.62 GHz from 270 MHz (multiplier = 6) **0** = INIT **1** = DP_2_70GHZ: Selects the 2.70 GHz link clock; PLL generates 2.70 GHz from 270 MHz (multiplier = 10) **2** = DP_5_40GHZ: Selects the 5.40 GHz link clock; PLL generates 5.40 GHz from 270 MHz (multiplier = 20) **3** = DP_8_10GHZ: Selects the 8.10 GHz link clock; PLL generates 8.10 GHz from 270 MHz (multiplier = 30) **4** = EDP_2_16GHZ: Selects the 2.16 GHz link clock; PLL generates 2.16 GHz from 270 MHz (multiplier = 8). Can be used only if the eDP panel reports support for it in the DPCD registers. **5** = EDP_2_43GHZ: Selects the 2.43 GHz link clock; PLL generates 2.43 GHz from 270 MHz (multiplier = 9). Can be used only if the eDP panel reports support for it in the DPCD registers. **6** = EDP_3_24GHZ: Selects the 3.24 GHz link clock; PLL generates 3.24 GHz from 270 MHz (multiplier = 12). Can be used only if the eDP panel reports support for it in the DPCD registers. **7** = EDP_4_32GHZ: Selects the 4:32 GHz link clock; PLL generates 4:32 GHz from 270 MHz (multiplier = 16). Can be used only if the eDP panel reports support for it in the DPCD registers. **8** = TMDS_0_25_TO_0_85GHZ: Select 0.25G to 0.85G TMDS clock; PLL takes reference clock range from 25 MHz to 85 MHz (multiplier = 10). It covers 250 Mbps to 650 Mbps(ref clock 25~65 MHz. **9** = TMDS_0_50_TO_1_70GHZ: Select 0.5G to 1.7G TMDS clock; PLL takes reference clock range from 50 MHz to 170 MHz (multiplier = 10). It covers 650Mbps to 1650 Mbps (ref clock 65~165 MHz). **10** = TMDS_1_00_TO_3_40GHZ: Select 1G to 3.4G TMDS clock; PLL takes reference clock range from 100 MHz to 340 MHz (multiplier = 10). It covers 1650 Mbps to 3400 Mbps(ref clock 165~340 MHz). **11** = TMDS_3_40_TO_6_00GHZ: Select 3.4 to 6G TMDS clock; PLL takes reference clock range from 150 MHz to 300 MHz. It's for HDMI20 high-speed mode(multiplier = 20). It covers 3400 Mbps to 6000 Mbps. **12** = HDMI_FRL_3G: Select the 3G link clock; PLL generates 3 GHz from 100 MHz (multiplier = 30). **13** = HDMI_FRL_6G: Select the 6G link clock; PLL generates 6 GHz from 100 MHz (multiplier = 60). **14** = HDMI_FRL_8G: Select the 8G link clock; PLL generates 8 GHz from 100 MHz (multiplier = 80). **15** = HDMI_FRL_10G: Select the 10G link clock; PLL generates 10 GHz from 100 MHz (multiplier = 100). **16** = HDMI_FRL_12G: Select the 12G link clock; PLL generates 12 GHz from 100 MHz (multiplier = 120). **17** = EDP_6_75GHZ: Selects the 6.75 GHz link clock; PLL generates 6.75 GHz from 270 MHz (multiplier = 25). Can be used only if the eDP panel reports support for it in the DPCD registers. **18** = EDP_6_48GHZ: Selects the 6.48 GHz link clock; PLL generates 6.48 GHz from 270 MHz (multiplier = 24). Can be used only if the eDP panel reports support for it in the DPCD registers. **19** = EDP_5_94GHZ: Selects the 5.94 GHz link clock; PLL generates 5.94 GHz from 270 MHz (multiplier = 22). Can be used only if the eDP panel reports support for it in the DPCD registers.

Bit	R/W	Reset	Description
			Note for DVI and HDMI1.4/2.0, there are 4 different reference clock range defined. Software needs to select the value based on the reference clock frequency.
17:16	RW	0x2	**MODE_BYPASS**: This will bypass the clock selected by FE with a specific clock. This is required if the SOR is operating in DP mode. Changing this field will require DP link re-training. **0** = NONE: No clock bypass, use whatever clock is chosen by FE. **1** = DP_NORMAL: Use the clock generated by the analog macro. This is required when the SOR is being used in DP or FRL mode. **1** = FRL_NORMAL: Use the clock generated by the analog macro. This is required when the SOR is being used in DP or FRL mode. **2** = DP_SAFE: Use the safe_clock for the SOR. This is used in DP or FRL mode, when the clock from analog macro is not ready. **2** = FRL_SAFE: Use the safe_clock for the SOR. This is used in DP or FRL mode, when the clock from analog macro is not ready. **2** = INIT **3** = FEEDBACK: When the SOR is awake, use the clock that is generated by the analog macro. This is the same clock that is used with DP_NORMAL, but the CMGR can switch this clock to safe_clock when it needs to. This is used in TMDS mode.
15:12	RO	X	**HEAD**: Indicates whether the SOR is connected to some head or neither. HEAD is only valid if MODE==NORMAL, otherwise it may contain stale information. **0** = _0 **1** = _1 **2** = _2 **3** = _3 **4** = _4 **5** = _5 **6** = _6 **7** = _7 **15** = NONE
7:6	RO	X	**MODE**: Indicates whether FE has selected the NORMAL clock (as controlled by the DIV field) or the SAFE clock. Note that at any time the software expects to change error has occurred and should be reported. **1** = NORMAL **2** = SAFE

NV_PDISP_FE_SW_SYS_CAP

Offset: 0x30000
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**WBOR7_EXISTS**: **0** = INIT **0** = NO **1** = YES
31	0x0	**WBOR_EXISTS_7**: **0** = INIT **0** = NO **1** = YES
30	0x0	**WBOR6_EXISTS**: **0** = INIT **0** = NO **1** = YES
30	0x0	**WBOR_EXISTS_6**: **0** = INIT **0** = NO **1** = YES
29	0x0	**WBOR5_EXISTS**: **0** = INIT **0** = NO **1** = YES
29	0x0	**WBOR_EXISTS_5**: **0** = INIT **0** = NO **1** = YES
28	0x0	**WBOR4_EXISTS**: **0** = INIT **0** = NO **1** = YES
28	0x0	**WBOR_EXISTS_4**: **0** = INIT **0** = NO **1** = YES
27	0x0	**WBOR3_EXISTS**: **0** = INIT **0** = NO **1** = YES
27	0x0	**WBOR_EXISTS_3**: **0** = INIT **0** = NO **1** = YES
26	0x0	**WBOR2_EXISTS**: **0** = INIT **0** = NO **1** = YES
26	0x0	**WBOR_EXISTS_2**: **0** = INIT **0** = NO **1** = YES

Bit	Reset	Description
25	0x0	**WBOR1_EXISTS**: **0** = INIT **0** = NO **1** = YES
25	0x0	**WBOR_EXISTS_1**: **0** = INIT **0** = NO **1** = YES
24	0x0	**WBOR0_EXISTS**: **0** = INIT **0** = NO **1** = YES
24	0x0	**WBOR_EXISTS_0**: **0** = INIT **0** = NO **1** = YES
23	0x0	**DSI3_EXISTS**: **0** = INIT **0** = NO **1** = YES
23	0x0	**DSI_EXISTS_3**: **0** = INIT **0** = NO **1** = YES
22	0x0	**DSI2_EXISTS**: **0** = INIT **0** = NO **1** = YES
22	0x0	**DSI_EXISTS_2**: **0** = INIT **0** = NO **1** = YES
21	0x0	**DSI1_EXISTS**: **0** = INIT **0** = NO **1** = YES
21	0x0	**DSI_EXISTS_1**: **0** = INIT **0** = NO **1** = YES
20	0x0	**DSI0_EXISTS**: **0** = INIT **0** = NO **1** = YES
20	0x0	**DSI_EXISTS_0**: **0** = INIT **0** = NO **1** = YES

Bit	Reset	Description
15	0x0	**SOR7_EXISTS**: **0** = INIT **0** = NO **1** = YES
15	0x0	**SOR_EXISTS_7**: **0** = INIT **0** = NO **1** = YES
14	0x0	**SOR6_EXISTS**: **0** = INIT **0** = NO **1** = YES
14	0x0	**SOR_EXISTS_6**: **0** = INIT **0** = NO **1** = YES
13	0x0	**SOR5_EXISTS**: **0** = INIT **0** = NO **1** = YES
13	0x0	**SOR_EXISTS_5**: **0** = INIT **0** = NO **1** = YES
12	0x0	**SOR4_EXISTS**: **0** = INIT **0** = NO **1** = YES
12	0x0	**SOR_EXISTS_4**: **0** = INIT **0** = NO **1** = YES
11	0x0	**SOR3_EXISTS**: **0** = INIT **0** = NO **1** = YES
11	0x0	**SOR_EXISTS_3**: **0** = INIT **0** = NO **1** = YES
10	0x0	**SOR2_EXISTS**: **0** = INIT **0** = NO **1** = YES
10	0x0	**SOR_EXISTS_2**: **0** = INIT **0** = NO **1** = YES

Bit	Reset	Description
9	0x0	**SOR1_EXISTS**: **0** = INIT **0** = NO **1** = YES
9	0x0	**SOR_EXISTS_1**: **0** = INIT **0** = NO **1** = YES
8	0x0	**SOR0_EXISTS**: **0** = INIT **0** = NO **1** = YES
8	0x0	**SOR_EXISTS_0**: **0** = INIT **0** = NO **1** = YES
7	0x0	**HEAD7_EXISTS**: **0** = INIT **0** = NO **1** = YES
7	0x0	**HEAD_EXISTS_7**: **0** = INIT **0** = NO **1** = YES
6	0x0	**HEAD6_EXISTS**: **0** = INIT **0** = NO **1** = YES
6	0x0	**HEAD_EXISTS_6**: **0** = INIT **0** = NO **1** = YES
5	0x0	**HEAD5_EXISTS**: **0** = INIT **0** = NO **1** = YES
5	0x0	**HEAD_EXISTS_5**: **0** = INIT **0** = NO **1** = YES
4	0x0	**HEAD4_EXISTS**: **0** = INIT **0** = NO **1** = YES
4	0x0	**HEAD_EXISTS_4**: **0** = INIT **0** = NO **1** = YES

Bit	Reset	Description
3	0x0	**HEAD3_EXISTS**: **0** = INIT **0** = NO **1** = YES
3	0x0	**HEAD_EXISTS_3**: **0** = INIT **0** = NO **1** = YES
2	0x0	**HEAD2_EXISTS**: **0** = INIT **0** = NO **1** = YES
2	0x0	**HEAD_EXISTS_2**: **0** = INIT **0** = NO **1** = YES
1	0x0	**HEAD1_EXISTS**: **0** = INIT **0** = NO **1** = YES
1	0x0	**HEAD_EXISTS_1**: **0** = INIT **0** = NO **1** = YES
0	0x0	**HEAD0_EXISTS**: **0** = INIT **0** = NO **1** = YES
0	0x0	**HEAD_EXISTS_0**: **0** = INIT **0** = NO **1** = YES

7.6 Consumer Electronics Control (CEC)

7.6.1 Overview

The HDMI Consumer Electronics Control (CEC) block supports CEC standard communication over any an HDMI connection attached to an Orin system.

CEC supports remote control of HDMI devices attached to the Orin Series SoC, and also allows other HDMI devices to control Orin Series SoC functions. Refer to the CEC appendix of the HDMI specification for details of the CEC communications link.

7.6.2 Functional Description

CEC is an APB target module. It is located at address NV_ADDRESS_MAP_CEC_BASE, accessible in PIO mode using AON fabric. As part of AON, the CEC APB interface uses v3.0 and supports PSLVERR.

The CEC consists of hardware state machines that send and receive bytes over the CEC wire. It has a simple host interface in which one byte can be sent and received at a time. CEC supports hardware buffering for 65 blocks (65* 25msec = 1625msec) of 10 bit each. Note that, including the RX_REGISTER plus the 64 block FIFO, the CEC buffer can accommodate 65 blocks.

There are two wake options that trigger an SC7 exit on a CEC message:

- CEC external wake event, such as the wake signal from the Pad is routed directly to PMC Wake engine.
  - In PMC, a wake can be configured to raise an interrupt to SPE without waking up the rest of the SoC.
  - In this option, SPE cluster can be put in low-power state (including power-gated state). Whenever the external wake event triggers, PMC sends an interrupt to SPE-AVIC, as well as to SPE-PM. SPE-PM brings SPE out of low-power state and it can then service the wake event.
- CEC internal interrupt without enabling wake event.
  - CEC interrupt is raised whenever there is a valid start bit and logical address check passes. This interrupt is sent to SPE-AVIC and LIC. SPE software can further process the incoming messages.

The CEC has interrupts for interrupt-driven input and output, and also permits polling control.

7.6.3 Programming Guidelines

7.6.3.1 Initialization

7.6.3.1.1 Clock and Reset

The CEC uses a fixed clock source: the APB bus clock–32.768kHz clock, which is always running. It also has a clock enable and reset in the CAR block.

- To remove CEC reset, use register CLK_RST_CONTROLLER_RST_DEV_CEC_0_SWR_CEC_RST=DISABLE.
- To enable the CEC clock, use register CLK_RST_CONTROLLER_CLK_OUT_ENB_CEC_0_CLK_ENB_CEC=ENABLE

During initialization, software updates the RX logical address in the following sequence:

1. Initialization (physical and logical address expected to be updated at this point).
2. Disable TX_RX_MODE, poll to check that the value got reflected.
3. Update logical address in the RX register.
4. Enable TX_RX_MODE and poll.

Exceptions for the preceding guidelines are while writing to the TX data registers and reading from the RX data registers. In these cases, there do not disable TX_RX_MODE.

7.6.3.1.2 Interrupt

The CEC interrupt is in INTR_CTLR_COMMON_SLICE5_GISR_0, BIT 2. The interrupt is sent to SPE-AVIC (AVIC slice 2, interrupt number 17).

Enable CEC INT_MASK for TX_* and RX_* interrupts. TX_REGISTER_EMPTY, RX_REGISTER_FULL, and RX_BUFFER_FULL are the key interrupts; most others are errors. Refer to the Interrupt Controller chapter for more information.

7.6.3.1.3 Pin mux

The CEC is available on the HDMI_CEC pin. The following is the CEC pin mux configuration:

- PM = 0 (primary CEC function)
- PUPD = NORMAL
- TRISTATE = NORMAL
- E_INPUT = ENABLE
- OD = ENABLE

7.6.4 CEC Registers

7.6.4.1 Registers

Refer to "Reading Register Tables" in the Introduction chapter for the register table protocol and recommendations for accessing registers. For more information on Consumer Electronics Control (CEC), refer to the CEC appendix of the HDMI specification.

CEC_SW_CONTROL_0

Software controlled mode is not supported. Leave disabled.

Offset: 0x0
Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x000000XX (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RW	DISABLE	**MODE**: **0** = DISABLE **1** = ENABLE
4	RO	X	**FILTERED_RX_DATA_PIN**
0	RO	X	**RAW_INPUT_DATA_PIN**

CEC_HW_CONTROL_0

Follow the steps below to reset the CEC engine.

1. Set the CEC_HW_CONTROL_TX_RX_MODE to DISABLE
2. Poll for the CEC_HW_CONTROL_TX_RX_MODE to read DISABLE
3. Set all the interrupt enable bits in INT_MASK to DISABLE
4. Set all the wake enable bits in WAKE_MASK to DISABLE
5. Wait one second (one sec is the maximum CEC bus timeout period from the specification)
6. Set the CEC_HW_CONTROL_TX_RX_MODE to ENABLE.
7. Poll for the CEC_HW_CONTROL_TX_RX_MODE to read ENABLE and perform restart.

Other devices on the bus may have been successfully communicating, so there may be a few spurious false start bits and other receive bus anomalies, before normal operations are achieved.

The CEC_HW_CONTROL register contains the control bits and fields for operating and configuring the hardware CEC engine.

Note: Once the block is enabled, software must not attempt to change these configuration parameters without first disabling the block.

The specification permits a single physical entity to claim logical addresses reserved for different functions (e.g., "Tuner x" and "Playback device y"). For a PC, it is possible to be a multi-function device, so hardware may need to be able to respond to two (or more) logical addresses (as many combinations as make sense). The next fields are used to configure the logical addresses hardware responds to. Of course, as per the specification, hardware always responds to the broadcast address. The field is a 15 bit field where each bit position is associated with one of the logical addresses. A '1' in any bit position, causes the hardware to respond to the associated logical address (i.e., capture data blocks/frames addressed to the address and properly ACK/NAK said blocks). To keep things simple, the mapping of bit position to logical address is direct, i.e., bit N maps to logical address N.

Offset: 0x4
Read/Write: R/W

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b000x,xxx0,xxxx,xx00,0000,0000,0000,0000)

Bit	Reset	Description
31	DISABLE	**TX_RX_MODE**: **0** = DISABLE **1** = ENABLE This bit enables or disables the operation of the hardware CEC engine. If the TX_RX_MODE is changed to DISABLE, then hardware engine returns to IDLE state irrespective of what it is doing and drives a one onto CEC bus. Note that if the CEC_SW_CONTROL_MODE is ENABLED, it automatically disables the hardware CEC engine operation.
30	DISABLE	**FAST_SIM_MODE**: **0** = DISABLE **1** = ENABLE The bit is used to speed up RTL simulation for verification purposes. Software sets it to DISABLE.
29	DISABLE	**AUTO_CLR_TX_EMPTY_INTR**: **0** = DISABLE **1** = ENABLE
24	BLOCK	**TX_NAK_MODE**: **0** = BLOCK **1** = FRAME The specification provides multiple options for what the initiator does if an intermediate block of a frame is NAKd, i.e., does the initiator cease transmitting immediately, or complete the rest of the frame. This bit permits the transmit state machine to operate in either mode. The choices are BLOCK, which means transmission ceases at the first NAKd block, or FRAME, which means that transmission always continues to the end of the frame.
17	DISABLE	**RX_TIMING_3_EN**: **0** = DISABLE **1** = ENABLE
16	BLOCK	**RX_NAK_MODE**: **0** = BLOCK **1** = FRAME The specification provides multiple options for what the initiator does if an intermediate block of a frame is NAKd, i.e., does the initiator cease transmitting immediately, or complete the rest of the frame. The specification says that a follower may NAK a frame at any time, but there is concern that some initiators might not recover from such an early NAKd frame. This bit permits the receive state machine to operate in either mode. The choices are BLOCK, which means that a frame is NAKd as soon as a RX_REGISTER_OVERRUN is detected; or FRAME, which means that the hardware keeps track of any RX_REGISTER_OVERRUNs that occur during frame reception, and NAK only during the last block.
15	DISABLE	**RX_SNOOP**: **0** = DISABLE **1** = ENABLE When enabled, the hardware intercepts and forwards to software all traffic on the CEC bus. It continues to ACK/NAK only those addresses that are assigned to it.
14:0	0x0	**RX_LOGICAL_ADDRS**: All logical addresses to be matched. One bit per address.

CEC_INPUT_FILTER_0

The CEC_INPUT_FILTER register is used to configure the hardware filtering that is required to deglitch the incoming receive data line. As data arrives into the chip, it is pushed into a 1-bit wide FIFO that is a maximum of 64 entries deep. The actual used depth of the FIFO is set using the CEC_INPUT_FILTER_FIFO_LENGTH field. The used length is equal to CEC_INPUT_FILTER_FIFO_LENGTH+1. A datum is pushed into the FIFO once per uS tick. The FIFO bits are reset to '1' (the idle condition of the CEC bus). The filtered output of the FIFO is computed roughly as follows:

```
filterMsk[63:0] = (1 << CEC_INPUT_FILTER_FIFO_LENGTH + 1) - 1;

if (reset || (CEC_INPUT_FILTER_MODE == CEC_INPUT_FILTER_MODE_DISABLE))
    fifo[63:0] = 0xffffffffffffffff;
else if (clock tick time)
    fifo[63:0] = (fifo[63:0] << 1) | rawInputDataPin;

if (reset || (CEC_INPUT_FILTER_MODE == CEC_INPUT_FILTER_MODE_DISABLE))
    filteredRxDataPin = 1;
else if ((clock tick time) && ((fifo & filterMsk) == filterMsk))
    filteredRxDataPin = 1;
else if ((clock tick time) && ((~fifo & filterMsk) == filterMsk))
    filteredRxDataPin = 0;
else
    filteredRxDataPin = filteredRxDataPin;
```

This logic makes sure that the filteredRxDataPin output only transitions to 0 or 1 when all the examined bits in the FIFO are also 0 or 1, thus the filtered data line will not transition as long as there are glitches in the FIFO. The length of 64 provides a maximum operational length of 64 uS.

Also note that every state transition on the filtered receive data line is reportable to software via the interrupt register described later.

Offset: 0x8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
31	DISABLE	**MODE**: **0** = DISABLE **1** = ENABLE
5:0	0x0	**FIFO_LENGTH**

CEC_SPARE_0

Spare register for future use.

Offset: 0xc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**SPARES**

CEC_TX_REGISTER_0

The CEC_HW_TX_REGISTER register is used by software to provide the hardware with the data to be transmitted on the bus. Note that hardware 'blindly' transmits what it is given, e.g., it does not check to be sure that a proper legal initiator address has been provided, it does not check to be sure that the maximum frame length of 16 is not violated, etc. These higher level protocol checks are the domain of the software. To facilitate easier software programming, and smoother operation, hardware makes its own working copy of the fields in this register, thus quickly freeing up the register for software to write the next block. (Basically, it is a one-deep FIFO with fullness reported via the TX_REGISTER_EMPTY bit).

RETRY FRAME | LAST_FRAM_ SENT_BY_US | WAIT TIME

```
1            | YES                  | TIMING2_BUS_IDLE_TIME_RETRY_FRAME
1            | NO                   | TIMING2_BUS_IDLE_TIME_RETRY_FRAME
0            | NO                   | TIMING2_BUS_IDLE_TIME_NEW_FRAME
0            | YES                  | TIMING2_BUS_IDLE_TIME_ADDITIONAL_FRAME
```

If a frame is sent immediately following the previous frame, hardware waits for "TIMING2_BUS_IDLE_TIME_ADDITIONAL_FRAME," however, if the bus is used elsewhere before that time elapses, the hardware resets its wait time counter and waits for "TIMING2_BUS_IDLE_TIME_NEW_FRAME."

Offset: 0x10
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00010000 (0b0xxx,xxxx,xxxx,xx01,xxx0,xxx0,0000,0000)

Bit	R/W	Reset	Description
31	RO	UNLOCKED	**WR_LOCK**: Gets set/locked once Tx data register is written and reset/unlocked once the data is transferred to core clock Check for TX_EMPTY_INT and WR_LOCK before writing into TX_REGISTER **1** = LOCKED **0** = UNLOCKED
17	RW	DISABLE	**RETRY_FRAME**: **0** = DISABLE **1** = ENABLE Indicates if the current frame is a retry frame or not. Based on this value, hardware chooses appropriate bus idle time programmed in TX_TIMING2 register and waits for the bus to be idle before it can attempt to send start bit. This bit is only meaningful when TX_GENERATE_START_BIT is set.
16	RW	ENABLE	**GENERATE_START_BIT**: **0** = DISABLE **1** = ENABLE Indicates to the hardware to precede the transmission of this block of data with a start bit.
12	RW	DIRECT	**ADDRESS_MODE**: **0** = DIRECT **1** = BROADCAST This bit indicates to the hardware whether this particular block is directly addressed or broadcast addressed (which in turn dictates how hardware is to interpret the ACK/NAK for the block).
8	RW	0x0	**EOM** The EOM bit read from the bus.
7:0	RW	0x0	**DATA** The eight bits of address/data to be transmitted on the bus. The data is always transmitted MSB (bit 7) first to facilitate smoother EOM. This is the "end of message" bit for this block of data. This bit is set at the last block of the frame.

CEC_RX_REGISTER_0

The CEC_HW_RX_REGISTER register is used by hardware to buffer data to the software. When a block of data has been received from the bus, it is stored here for software (and when the buffer is fully filled, RX_REGISTER_FULL bit is set). The hardware has a 64deep of the receive register in order to give software as much time as possible to empty it.

Offset: 0x14
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000XXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
9	X	**ACK** The ACK/NAK bit as read from the bus. In some cases (broadcast block), even though our device may have ACKd the frame, some other device on the bus may NAK the frame, and software must know this.
8	X	**EOM** The EOM bit read from the bus.
7:0	X	**DATA** The eight bits of data that were read from the bus.

CEC_RX_TIMING_0_0

Note: Configure all the timing registers before enabling the hardware CEC engine.

Offset: 0x18
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x8b9a7180 (0b1000,1011,1001,1010,0111,0001,1000,0000)

Bit	Reset	Description
31:24	0x8b	**RX_START_BIT_MIN_DURATION** nom 4.3 mS (4300*0.032768 = 141 cycles) = 4.302 mS The minimum total duration of the start bit from the first detected hi->lo transition to the hi->lo transition that begins the first address bit. If the start bit duration underruns this time, the start bit is considered invalid, and is ignored. As the syncer can use 30 usec in the signal by shrinking it, programming two cycle less, i.e., 141 - 2 = 139
23:16	0x9a	**RX_START_BIT_MAX_DURATION** nom 4.7 mS, (4700*0.032768 = 154 cycles) = 4.699 mS The maximum total duration of the start bit from the first detected hi->lo transition to the hi->lo transition that begins the first address bit. If the start bit duration overruns this time, the start bit is considered invalid, and is ignored.
15:8	0x71	**RX_START_BIT_MIN_LO_TIME** nom 3.5 mS, (3500*0.032768 = 114 cycles) = 3.479 mS The minimum time the received bus can remain low, and still be considered the beginning of a proper start bit. If the start bit lo time underruns this time, the start bit is considered invalid, and is ignored. As the syncer can use 30 usec in the signal by shrinking it, programming one cycle less, i.e., 114 - 1 = 113
7:0	0x80	**RX_START_BIT_MAX_LO_TIME** nom 3.9 mS, (3900*0.032768 = 128 cycles) = 3.906 mS The maximum time the received bus can remain low and still be considered the beginning of a proper start bit. If the start bit lo time overruns this time, the start bit is considered invalid, and is ignored.

CEC_RX_TIMING_1_0

Note: Configure all the timing registers before enabling the hardware CEC engine.

Offset: 0x1c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x415a2238 (0b0100,0001,0101,1010,0010,0010,0011,1000)

Bit	Reset	Description
31:24	0x41	**RX_DATA_BIT_MIN_DURATION** nom 2.05 mS, (2050*0.032768 = 67 cycles) = 2.045 mS The minimum total duration of the start bit from the first detected hi->lo transition to the hi->lo transition that begins the first address bit. If the start bit duration underruns this time, the start bit is considered invalid, and is ignored. As the syncer can use 30 usec in the signal by shrinking it, programming two cycle less, i.e., 67 - 2 = 65
23:16	0x5a	**RX_DATA_BIT_MAX_DURATION** nom 2.75 mS, (2750*0.032768 = 90 cycles) = 2.746 mS The maximum total duration of the start bit from the first detected hi->lo transition to the hi->lo transition that begins the first address bit. If the start bit duration overruns this time, the start bit is considered invalid, and is ignored.
15:8	0x22	**RX_DATA_BIT_SAMPLE_TIME** nom 1.05 mS, (1050*0.032768 = 34 cycles) = 1.037 mS The time to sample that data bit.
7:0	0x38	**RX_DATA_BIT_MAX_LO_TIME** nom 1.7 mS, (1700*0.032768 = 56 cycles) = 1.709 mS The maximum time the received bus can remain low, and still be considered the beginning of a proper start bit. If the start bit lo time overruns this time, the start bit is considered invalid, and is ignored.

CEC_RX_TIMING_2_0

Note: Configure all the timing registers before enabling the hardware CEC engine.

Offset: 0x20
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000003e (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0011,1110)

Bit	Reset	Description
7:0	0x3e	**RX_END_OF_BLOCK_TIME** nom 1.9 mS, (1900*0.032768 = 62 cycles) = 1.89 mS The time to wait after the start of the final ACK/NAK phase of frame transmission before returning to the idle state. (Needed since the last ACK/NAK bit is not followed by another hi->lo transition.)

CEC_TX_TIMING_0_0

The next set of registers configures the CEC logic for hardware assisted operation. To permit flexible configuration (to support possible semi-compliant devices) the bit timing control and check values are made programmable (rather than directly using the specification values). While the block itself works on a 32K clock, in order to save register space, these timing numbers are specified in units of 30.5 uS.

First, the registers for the various timing intervals.

Note: Configure all the timing registers before enabling the hardware CEC engine.

Note: For details of exact values and tolerances, refer to the CEC appendix of the HDMI specification.

Offset: 0x24
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x76089379 (0b0111,0110,xxxx,1000,1001,0011,0111,1001)

Bit	Reset	Description
31:24	0x76	**TX_BUS_ERROR_LO_TIME** nom 3.6 mS (3600*0.032768 = 118 cycles) = 3.601 mS Time to drive bus lo when bus error must be signaled on the bus.
19:16	0x8	**TX_BUS_XITION_TIME** nom 250 uS, (250*0.032768 = 8 cycles) = 244 uS Time to wait for bus to settle after transitioning output.
15:8	0x93	**TX_START_BIT_DURATION** nom 4.5 mS, (4500*0.032768 = 147 cycles) = 4.486 mS The total duration of the start bit.
7:0	0x79	**TX_START_BIT_LO_TIME** nom 3.7 mS, (3700*0.032768 = 121 cycles) = 3.692 mS How long to hold the bus lo during the start bit.

CEC_TX_TIMING_1_0

Note: Configure all the timing registers before enabling the hardware CEC engine.

Offset: 0x28
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x224f1431 (0b0010,0010,0100,1111,0001,0100,0011,0001)

Bit	Reset	Description
31:24	0x22	**TX_ACK_NAK_BIT_SAMPLE_TIME** nom 1.05 mS, (1050*0.032768 = 34 cycles) = 1.037 mS The time to sample that ACK/NAK bit.
23:16	0x4f	**TX_DATA_BIT_DURATION** nom 2.4 mS, (2400*0.032768 = 79 cycles) = 2.41 mS The total duration of a data or ACK/NAK bit.
15:8	0x14	**TX_HI_DATA_BIT_LO_TIME** nom 0.6 mS, (600*0.032768 = 20 cycles) = 0.61 mS How long to hold the bus lo when transmitting a '1'.
7:0	0x31	**TX_LO_DATA_BIT_LO_TIME** nom 1.5 mS, (1500*0.032768 = 49 cycles) = 1.495 mS How long to hold the bus lo when transmitting a '0'.

CEC_TX_TIMING_2_0

CEC_HW_TX_TIMING2 contains the bus idle time values for the various scenarios indicated in the specification (section 9.1). Software indicates to the hardware the amount of delay in units of data bit periods, as defined by the CEC_HW_TX_TIMING1_TX_DATA_BIT_DURATION field.

Offset: 0x2c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000357 (0bxxxx,xxxx,xxxx,xxxx,xxxx,0011,0101,0111)

Bit	Reset	Description
11:8	0x3	**BUS_IDLE_TIME_RETRY_FRAME** The amount of time the bus needs to be idle before we can try resending the same frame The suggested value is >=3.
7:4	0x5	**BUS_IDLE_TIME_NEW_FRAME** The amount of time the bus needs to be idle before we can send a new frame if we were not the initiator for the previous frame. The suggested value is >=5.

Bit	Reset	Description
3:0	0x7	**BUS_IDLE_TIME_ADDITIONAL_FRAME** The amount of time the bus needs to be idle before we can send a new frame immediately after sending a frame. The suggested value is >=7.

CEC_INT_STAT_0

The INT_STAT register contains the individual interrupt status bits for the CEC hardware unit. As usual, the register INT_MASK contains the corresponding enable bit for each interrupt/status bit in the interrupt register. The interrupt enable determines whether the interrupt propagates to the PMU as an interrupt, but the interrupt status bit itself is always set if the described condition occurs. Software writes a '1' to any interrupt/status bit in order to clear it.

Note: Since the hardware state machine also looks at the state of the interrupt bits when determining what action(s) to take, it is important for software to operate in roughly the following order when processing interrupts from this source:

1. Read the interrupt register to determine what caused the interrupt, what needs doing
2. Perform the operations required, e.g., reload Tx register, read Rx register, etc.
3. Finally, clear the corresponding interrupt bits.

Offset: 0x30
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
18	X	**RX_PACKET_FILTER**: **0** = INACTIVE **1** = ACTIVE Hardware Rx engine filters unwanted and unintended packets. Once wanted intended packet received, this bit is asserted.
17	X	**RX_BUFFER_AFULL**: **0** = INACTIVE **1** = ACTIVE Hardware Rx engine writes Rx data to the RX_BUFFER. Once RX_BUFFER holds AFULL_THRESHOLD locations, this bit is asserted.
16	X	**RX_BUFFER_OVERRUN**: **0** = INACTIVE **1** = ACTIVE Hardware Rx engine writes Rx data to the RX_BUFFER. Once RX_BUFFER is fully filled and then another Rx data is received (which gets dropped), this bit is asserted.

Bit	Reset	Description
15	X	**RX_BUFFER_FULL**: **0** = INACTIVE **1** = ACTIVE Hardware Rx engine writes Rx data to the RX_BUFFER. Once RX_BUFFER is fully filled, this bit is asserted.
14	X	**FILTERED_RX_DATA_PIN_TRANSITION_L2H**: **0** = INACTIVE **1** = ACTIVE In direct software bus drive mode, software must be interrupted whenever the state of the received data line transitions. This interrupt asserted on a 0->1 transition.
13	X	**FILTERED_RX_DATA_PIN_TRANSITION_H2L**: **0** = INACTIVE **1** = ACTIVE In direct software bus drive mode, software must be interrupted whenever the state of the received data line transitions. This interrupt asserted on a 1->0 transition.
12	X	**RX_BUS_ERROR_DETECTED**: **0** = INACTIVE **1** = ACTIVE The receiver has detected the specific anomaly called a bus error and then signaled a bus error on the bus, per the specification.
11	X	**RX_BUS_ANOMALY_DETECTED**: **0** = INACTIVE **1** = ACTIVE The receiver detected some anomaly (including bus error) on the bus. Bus error has a specific definition in the specification, but there are other kinds of anomalies that can occur. All anomalies and bus errors are reported here.
10	X	**RX_START_BIT_DETECTED**: **0** = INACTIVE **1** = ACTIVE Whenever the receive engine detects a start bit, it sets this bit. The primary purpose of this bit is for debug.
9	X	**RX_REGISTER_OVERRUN**: **0** = INACTIVE **1** = ACTIVE Software failed to read the RX_REGISTER before hardware had another block of data ready to transfer. Hardware NAKs the block/frame, and the initiator must retry later. Software must flush the partially accumulated frame.
8	X	**RX_REGISTER_FULL**: **0** = INACTIVE **1** = ACTIVE The hardware has assembled a complete block from the bus and placed it into the RX_REGISTER. Software reads the block ASAP to ensure the RX_REGISTER is empty before hardware has the next block ready to load. When hardware Rx engine writes Rx data to the RX_REGISTER, this bit is asserted.
5	X	**TX_FRAME_TRANSMITTED**: **0** = INACTIVE **1** = ACTIVE This is asserted at the end of last data bit, i.e., ACK bit of the last block is sent. This is an indication that last block is sent but does not necessarily mean the transmission was successful. Software still looks for other interrupts to check for any errors.

Bit	Reset	Description
4	X	**TX_BUS_ANOMALY_DETECTED**: **0** = INACTIVE **1** = ACTIVE Sometime during block transmission, hardware detected anomalous behavior on the bus (e.g., bus remained low after transmitter had signaled hi). When such an anomaly is detected, the transmitter ceases operation, releases the bus hi, and flushes any pending data in the TX_TRANSMIT register.
3	X	**TX_ARBITRATION_FAILED**: **0** = INACTIVE **1** = ACTIVE Hardware sets this bit if during the address block phase of transmitting a frame it failed to win arbitration. In this case, transmit (of course) ceases, and the hardware flushes any pending data in the TX_REGISTER. Software must retry the frame from scratch (after the appropriate signal free time, i.e., hardware does not automatically retry).
2	X	**TX_FRAME_OR_BLOCK_NAKD**: **0** = INACTIVE **1** = ACTIVE Hardware sets this bit if any block/frame is NAKd. If software sees this bit set, it knows it must resend the frame later.
1	X	**TX_REGISTER_UNDERRUN**: **0** = INACTIVE **1** = ACTIVE The transmit register was empty at the time the transmit hardware needed to have the next block ready to send. This is an error condition. The transmitter has stopped transmitting, recovery requires resending the entire frame again from scratch.
0	X	**TX_REGISTER_EMPTY**: **0** = INACTIVE **1** = ACTIVE The transmit register is empty, so software is free to write the next block of the frame into the register. For multi-block frames, software must provide the next block before the current block is sent, otherwise a TX_REGISTER_UNDERRUN error occurs. The moment Tx engine reads from TX_REGISTER register and stores a local copy, this interrupt is asserted.

CEC_INT_MASK_0

Mask register for interrupts. When a field is set to ENABLE, a field with a value of one in the INT_STATUS register results in the interrupt line being asserted.

Offset: 0x34
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,x000,0000,0000,xx00,0000)

Bit	Reset	Description
18	DISABLE	**RX_PACKET_FILTER**: **0** = DISABLE **1** = ENABLE
17	DISABLE	**RX_BUFFER_AFULL**: **0** = DISABLE **1** = ENABLE
16	DISABLE	**RX_BUFFER_OVERRUN**: **0** = DISABLE **1** = ENABLE
15	DISABLE	**RX_BUFFER_FULL**: **0** = DISABLE **1** = ENABLE
14	DISABLE	**FILTERED_RX_DATA_PIN_TRANSITION_L2H**: **0** = DISABLE **1** = ENABLE
13	DISABLE	**FILTERED_RX_DATA_PIN_TRANSITION_H2L**: **0** = DISABLE **1** = ENABLE
12	DISABLE	**RX_BUS_ERROR_DETECTED**: **0** = DISABLE **1** = ENABLE
11	DISABLE	**RX_BUS_ANOMALY_DETECTED**: **0** = DISABLE **1** = ENABLE
10	DISABLE	**RX_START_BIT_DETECTED**: **0** = DISABLE **1** = ENABLE
9	DISABLE	**RX_REGISTER_OVERRUN**: **0** = DISABLE **1** = ENABLE
8	DISABLE	**RX_REGISTER_FULL**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**TX_FRAME_TRANSMITTED**: **0** = DISABLE **1** = ENABLE
4	DISABLE	**TX_BUS_ANOMALY_DETECTED**: **0** = DISABLE **1** = ENABLE
3	DISABLE	**TX_ARBITRATION_FAILED**: **0** = DISABLE **1** = ENABLE
2	DISABLE	**TX_FRAME_OR_BLOCK_NAKD**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
1	DISABLE	**TX_REGISTER_UNDERRUN**: **0** = DISABLE **1** = ENABLE
0	DISABLE	**TX_REGISTER_EMPTY**: **0** = DISABLE **1** = ENABLE

CEC_HW_DEBUG_RX_0

Offset: 0x38
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
27	X	**RXDATABIT_SAMPLE_TIMER**
26	X	**LOGICADDR_MATCH**
25	X	**FORCELOOUT**
24:21	X	**STATE**
20:17	X	**RXBIT_COUNT**
16:0	X	**DURATION_COUNT**

CEC_HW_DEBUG_TX_0

Offset: 0x3c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
26	X	**TXDATABIT_SAMPLE_TIMER**
25	X	**FORCELOOUT**
24:21	X	**STATE**
20:17	X	**TXBIT_COUNT**
16:0	X	**DURATION_COUNT**

CEC_RX_BUFFER_AFULL_CFG_0

Mask register for wakes. When a field is set to ENABLE, a field with a value of one in the WAKE_STATUS register results in the wake line being asserted.

Offset: 0x4c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000040 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x100,0000)

Bit	Reset	Description
6:0	0x40	**THRESHOLD**

CEC_RX_BUFFER_STAT_0

Offset: 0x50
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000,0000)

Bit	Reset	Description
6:0	0x0	**STAT**

CEC_RX_TIMING_3_0

Note: Configure all the timing registers before enabling the hardware CEC engine.

Offset: 0x54
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x382a1a0d (0b0011,1000,0010,1010,0001,1010,0000,1101)

Bit	Reset	Description
31:24	0x38	**RX_LOGIC0_BIT_MAX_LO_TIME** nom 1.9 mS, (1900*0.032768 = 84 cycles) = 1.9 mS The time to wait after the start of the final ACK/NAK phase of frame transmission before returning to the idle state. (Needed since the last ACK/NAK bit is not followed by another hi->lo transition.)

Bit	Reset	Description
23:16	0x2a	**RX_LOGIC0_BIT_MIN_LO_TIME**
15:8	0x1a	**RX_LOGIC1_BIT_MAX_LO_TIME**
7:0	0xd	**RX_LOGIC1_BIT_MIN_LO_TIME**

7.7 Audio Processing Engine (APE)

7.7.1 Overview

The Audio Processing Engine (APE) in the SoC can perform audio processing with minimal supervision from the main CPUs if required. It can also split audio processing tasks with the main CPUs, in which case the APE is generally used for more latency critical processes.

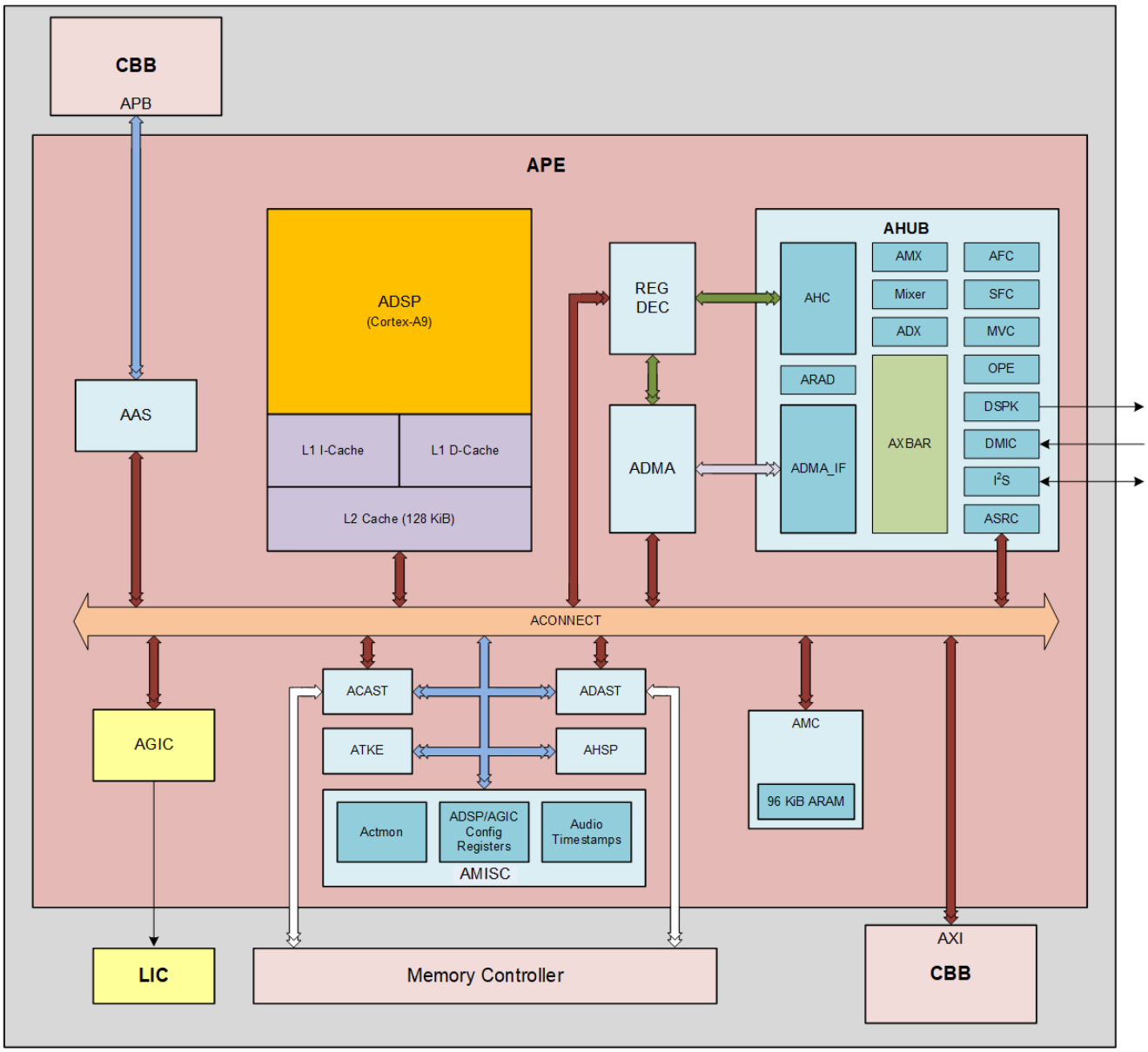
The APE contains an Audio DSP (ADSP), an internal Audio RAM (ARAM) with a dedicated Audio Memory Controller (AMC), an Audio Hub (AHUB) equipped with multiple hardware accelerators for audio signal pre-processing and post-processing, and an Audio DMA engine (ADMA). As shown in its block diagram below, the APE comprises these major functional blocks:

- ADSP based on an Arm Cortex®-A9 version r4p1 processor with L1 and L2 Caches
- AHUB containing the audio hardware accelerators, as well as all the required audio I/O interfaces
  - Audio Client Interface
  - Inter-IC Sound Interfaces (I^2S)
  - Digital Microphone Interfaces
  - Digital Speaker Interfaces
  - Mixers
  - Audio Multiplexers
  - Audio De-Multiplexers
  - Sampling Frequency Converters
  - Audio Flow Controllers
  - Output Processing Engines
  - Master Volume Controllers
  - Arbitrary Sample Rate Converters
  - Audio Sampling Rate Detectors
  - Audio Direct Memory Access Interfaces
- ADMA responsible for moving data between AHUB and DRAM/ARAM
- REGDEC serving as the register interface for the ADSP and CBB to access registers in AHUB or ADMA

- ACONNECT functioning as an AXI-based backbone directing requests from different Masters to the targeted Slaves in APE
- AMC providing the local memory controls for the local memory residing in APE
- AGIC providing all audio-related Interrupt controls as the name declares
- AMISC containing the configuration registers, decryption key slot registers, and timestamp
- ACAST providing address translation and the memory interface from ADSP to DRAM
- ADAST providing address translation and the memory interface from ADMA to DRAM
- AAS converting APB transactions from the CCPLEX via CBB to AXI transactions in APE
- AHSP consisting of synchronization registers for mailbox communications between ADSP and CCPLEX
- ATKE as the time keeper of the APE, and consisting of general-purpose timers and a watchdog timer

Note that the SPDIF interface is not supported in the SoC, and references to controls for it below should be treated as reserved.

Figure 7.32 APE Block Diagram



7.7.1.1 Standard and Compatibility

- I^2S (Inter-IC Sound) Interface
- AMBA APB Protocol
- AMBA AXI Protocol

7.7.1.2 List of Reference

This chapter makes the implicit use of the following document available from Arm® based on the assumption that readers are familiar with the Arm Architecture and have access to the documents for reference. Refer to the Arm website to download the document.

- *Arm Cortex®-A9 version r4p1*
  http://infocenter.arm.com/help/topic/com.arm.doc.100486_0401_10_en/cortex_a9_mpcore_trm_100486_0401_10_en.pdf

7.7.1.3 Glossary

Note that different names may end up in the same acronym when the entire TRM is put in perspective. The acronyms listed here are within the context of the Audio Processing Engine chapter.

Term	Definition
AAS	APB/AXI Shim
ACAST	Audio CPU AST
ACIF	Audio Client Interface
ADAST	Audio Data AST
ADMA	Audio Direct Memory Access
ADSP	Audio Digital Signal Processor
ADX	Audio De-Multiplexer
AFC	Audio Flow Controller
AGIC	Audio Generic Interrupt Controller
AHC	Audio Hub Configuration
AHUB	Audio Hub
AHSP	Audio Hardware Synchronization Primitives
AMC	Audio Memory Controller
AMISC	Audio Miscellaneous
AMX	Audio Multiplexer
ARAD	Audio Sampling Rate Detector
ASRC	Arbitrary Sample Rate Converter
AST	Address Space Translation

Term	Definition
ATKE	Audio Time Keeping Element
AXBAR	Audio Crossbar
CIF	Client Interface
CTI	Cross Trigger Interface
DMIC	Digital Microphone
DSM	Delta Sigma Modulation/Modulator
I2S	Inter-IC Sound Controller
LIC	Legacy Interrupt Controller
MBDRC	Multiband Dynamic Range Compression
Mixer	Device for merging input audio signals to produce a combined audio output
MVC	Master Volume Control
OPE	Output Processing Engine
PCM	Pulse Code Modulation/Modulator
PDM	Pulse Density Modulation/Modulator
PEQ	Parametric Equalizer
SFC	Sampling Frequency Converter
SPI	Shared Peripheral Interrupts
TSC	Timestamp System Counter

7.7.1.4 Relevant Chapters in the TRM

- Address Map
- Control BackBone (CBB)
- Clock Controller and Reset (CAR)
- CPU Complex (CCPLEX)
- Hardware Synchronization Primitives
- High Definition Audio (HDA)
- Interrupt Controllers
- Introduction
- Memory SubSystem (MSS)
- Multi-Purpose I/O Pins and Pin Multiplexing (PinMux)
- Timers

7.7.1.5 Features

The following is highlight of the APE features.

- Arbitrary Sample Rate Conversion
- Audio Flow Control
- Audio RAM (96 KiB) by Audio Direct Memory Access
- Audio Multiplexer
- Audio De-Multiplexer
- Mixer
- Audio Output Processing
- Sampling Frequency Conversion
- Master Volume Control
- Digital Speaker (x 2)
- Digital MIC (x 4)
- Inter-IC Sound (I^2S) Interface (x 6)
- Advanced Peripheral Bus (APB) for fast and directed Control/Data flow in APE and the SoC
- Advanced eXensible Interface (AXI) for fast and directed Control/Data flow in APE and the SoC
- Interrupts to CCPLEX (via IRQ and FIQ)
- Debug Port using CoreSight™ interface to ADSP
- Multiple Clock domains
  - MC_CLK
  - APE_CLK
  - ADSP_CLK
  - AHUB_CLK
- Multiple Resets
  - APE_reset
  - ADSP_reset
- Memory Controller Interface
- Interface for MMIO Requests
- TSC interface

7.7.2 Functional Descriptions

The functional descriptions of the APE are detailed in the following sections based on the functional blocks.

7.7.2.1 Audio Digital Signal Processor

The Audio Digital Signal Processor (ADSP) is an Arm Cortex®-A9 version r4p1 processor, configured with an L1 I-Cache and an L1 D-Cache each of 32 KiB, and a 128 KiB L2 I/D shared Cache, plus the Arm NEON Advanced SIMD extensions. Its main functions are:

- Algorithm-based Audio Processing
- Audio Decoding and Encoding, e.g., MP3
- Managing and Scheduling the programming of registers in Hardware Accelerators
- Managing Interrupts originated from Hardware Accelerators with Low Latency

7.7.2.2 Audio Hub

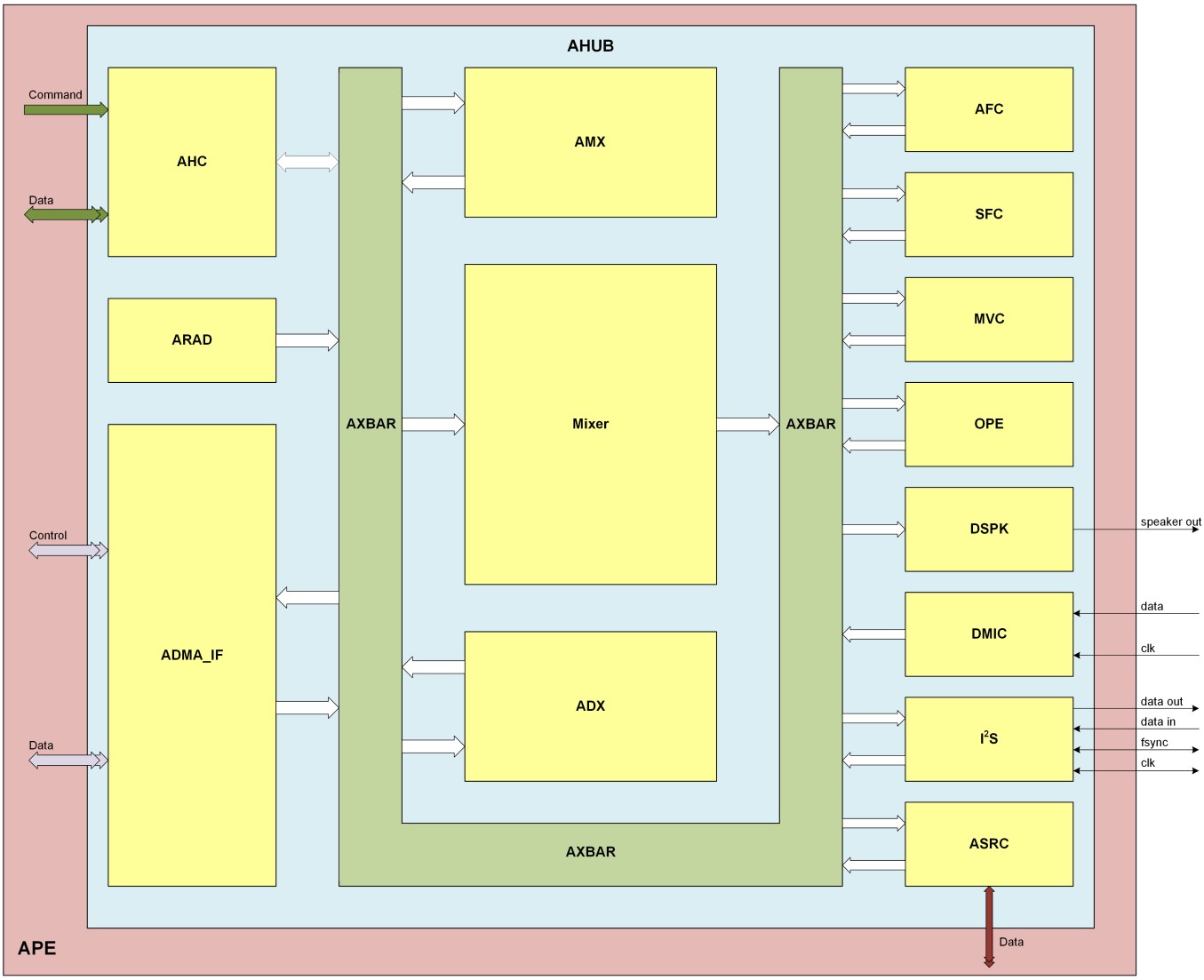
The Audio Hub (AHUB) comprises a collection of hardware accelerators for audio pre-processing and post-processing and a programmable full crossbar for routing audio data across these accelerators in time and in parallel. The main purpose of AHUB is to off load most repetitive tasks that do not require central decisions from the ADSP.

The AHUB also supports multiple interfaces to different audio devices in the host system, e.g., cellular baseband, audio codecs, Bluetooth modules, A/V receivers, etc. The AHUB provides six Inter-IC Sound (I^2S) Interfaces together with two Digital Speakers (DSPK) and four Digital Microphone (DMIC) interfaces. Audio signals routed to the HDMI interface are not going through APE, however, but delivered by the HDA interface. Refer to the High Definition Audio (HDA) chapter in this TRM for more details.

Besides supporting the multiple interfaces, the AHUB is also capable of handling all necessary protocols and signal quality requirements of these audio devices. The AHUB consists of the following key modules as depicted in the diagram below.

- Inter-IC Sound Controller (I^2S)
- Digital MIC Controller (DMIC)
- Mixer
- Audio Multiplexer (AMX)
- Audio De-Multiplexer (ADX)
- Sampling Frequency Converter (SFC)
- Audio Flow Controller (AFC)
- Output Processing Engine (OPE)
- Master Volume Control (MVC)
- Audio Direct Memory Access Interface (ADMA I/F)
- Audio Radio Detection (ARAD)
- Arbitrary Sample Rate Conversion (ASRC)

Figure 7.33 AHUB Block Diagram



The proprietary interface Audio Client Interface (ACIF) is employed to route audio samples through these accelerators and hence forms the fabric of AHUB. A switch called Audio Crossbar (AXBAR) is used to configure/modify the audio routing path between these accelerators. The accelerator modules and the audio routing are both configured via an AHUB Configuration (AHC) unit.

7.7.2.2.1 Audio Client Interface

Audio streams are routed through the AHUB by interconnecting various modules using the ACIF. In the register programming, a receive ACIF is referred to as RxCIF, and a transmit ACIF is referred to as TxCIF.

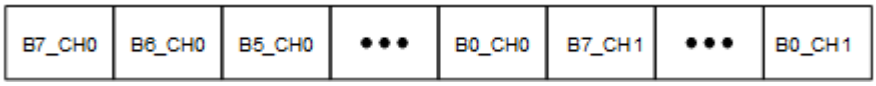
ACIF Protocol

Before a session starts, both Tx and Rx clients should know the frame length by their register configurations. The frame length is determined by the number of bits in each sample and the number of channels that the audio data carries. In the case of stereo audio data with 16 bit, a frame is 32 bits long.

Whenever Tx has data available to transmit to the Rx, it raises the FSYNC signal. Simultaneously the first bit of the data is put out on the data bus. The Tx should hold this FSYNC active for at least one clock but until the Rx client is ready. Similarly, it should also hold the data bus to the first bit of the frame until the Rx client is ready. When the Rx asserts ready, the Tx should lower the FSYNC at the end of that clock cycle. From the next clock cycle, Tx should transmit the rest of the frame (from the second bit onwards), one bit for each bit clock.

The figure below shows the transmission ordering of two-channel 8-bit audio data. Note that the bit ordering is each channel is big-endian.

Figure 7.34 Bit Ordering



Frame Length

The frame length is a product of the bit width and the number of channels in an audio stream. The register fields of the Tx and Rx clients should be programmed accordingly before the transmission starts to avoid incorrect frame parsing.

Clocking

Since all sessions are AHUB run under an AHUB clock, the clock should be fast enough to transmit audio stream data in all sessions. If a session is used for sending 8 kHz, stereo 16-bit audio stream, the AHUB clock should be at least 256 kHz. While a faster clock is better for data transmission, it is not necessarily good for power consumption. For the typical audio application like MP3 music playing, using the clock directly from the crystal without going through PLLs is recommended. Depending on the applications, users can program the clock registers to satisfy the requirement of clock frequency. The following table shows the various minimum clock frequencies for AHUB.

Table 7.85 AHUB Clock Frequencies

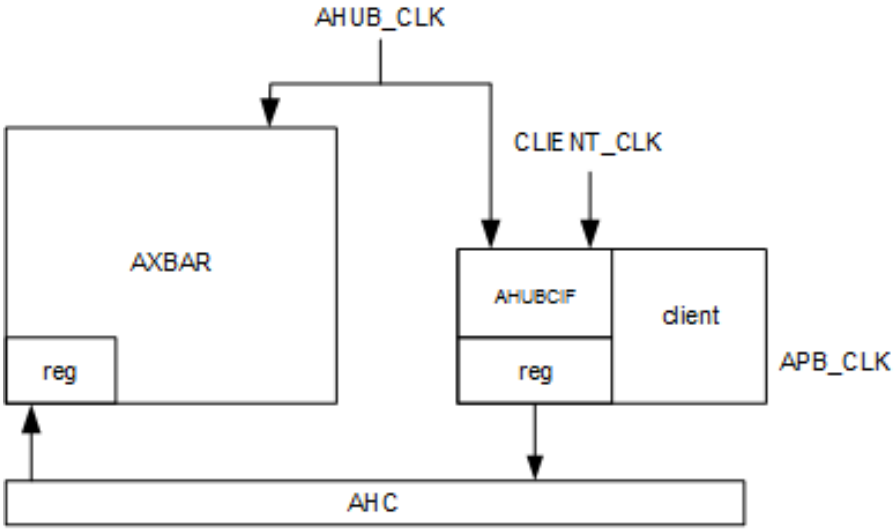
This table shows the minimum clock frequencies required for various operating modes. Higher frequencies may also be used.

Clock Frequency	Description
64 kHz	Slowest clock to carry 8-bit mono voice data in 8 kHz sampling

Clock Frequency	Description
1.4112 MHz	Clock to carry 16-bit stereo audio data in 44.1 kHz sampling
1.536 MHz	Clock to carry 16-bit stereo audio data in 48 kHz sampling
4.608 MHz	Clock to carry 24-bit stereo audio data in 96 kHz sampling or 16-bit six-channel audio data in 48 kHz sampling
12 MHz	Minimum crystal frequency
18.432 MHz	Clock to carry 24-bit eight-channel audio data in 96 kHz sampling
36.864 MHz	Clock to carry 24-bit eight-channel audio data in 192 kHz sampling (HDMI maximum)

While AHUB runs under one clock, AHUB_CLK, its clients have their own clocks. AHUBCIF is the clock boundary in the clients. The following figure shows the clocking scheme around AHUB.

Figure 7.35 Clock Domains



Packing/Unpacking Data

This feature is available only for the ADMA Interface. 32-bit packets coming from ADMA can be unpacked to 8- or 16-bit words before being transmitted to the AXBAR. The reverse path has the capability of packing 8- or 16-bit words in 32-bit packets.

The caveat is that the packing can only be done for incoming words of 8- or 16-bits, but not 24 bits, as a single 24-bit word cannot be placed (packed) in two different packets. So AXBAR_BITS in both cases can only be for 8- or 16-bits.

7.7.2.2.2 I²S Controller

The Inter-IC Sound (I²S) controller implements full-duplex and bidirectional and single direction point-to-point serial interfaces. It can interface with I²S-compatible devices.

I²S Interaction with External Device

The I²S controller can operate both as master and slave. It supports the following data transfer modes:

- I²S mode
- Left Justified Mode (LJM)
- Right Justified Mode (RJM)
- DSP mode, as defined in the Philips inter-IC-sound (I²S) bus specification
- PCM mode with short (one-bit-clock wide) and long-FSYNC (two bit-clocks wide)
- Network (Telephony) mode with independent slot selection for both Tx and Rx
- TDM mode with flexibility in number of slots with up to 16 slots.
- Capability to drive-out a High-Z outside the prescribed slot for transmission

The I²S controller can transmit and receive word lengths of 8, 16, 24, and 32. It supports u-Law and A-Law compression/decompression. The I²S controller can control the flow of traffic from another I²S controller operating on an independent bit clock (with a ppm difference compared with its own bit clock).

Transmission/Reception Data Formats

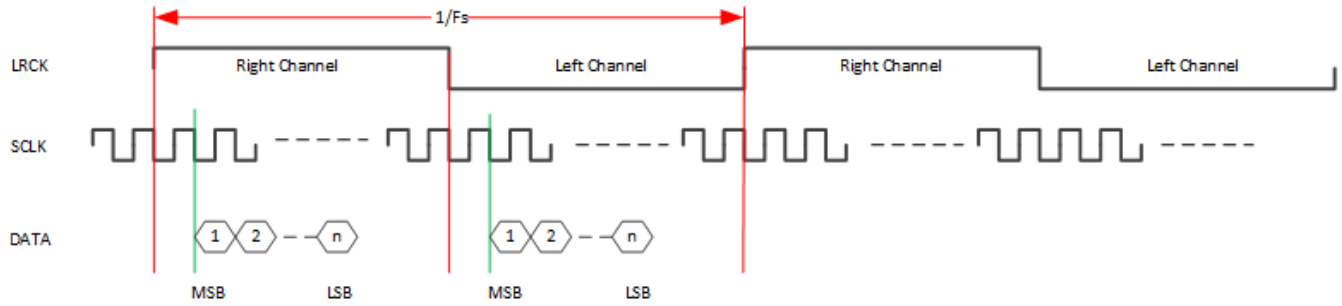
LRCK Modes

This subsection illustrates three LRCK modes: Basic I²S mode, Right Justified mode, and Left Justified mode.

Basic I²S Mode:

In Basic I²S mode, data starts one SCLK after the LRCK edge (Offset = 1).

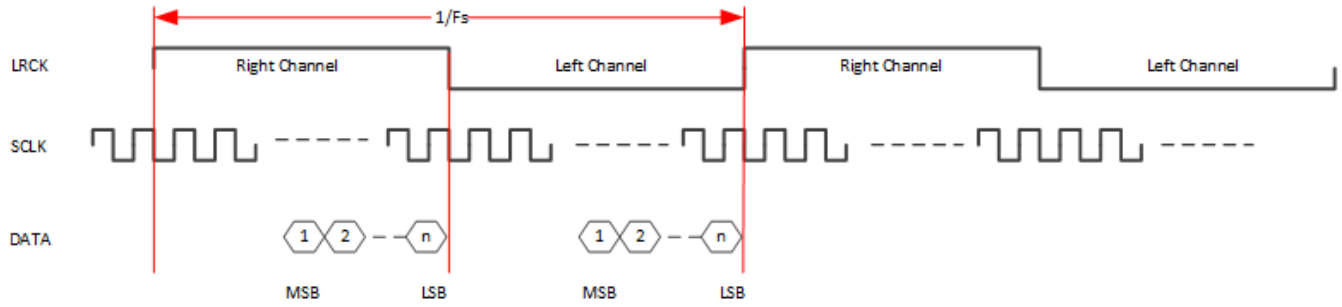
Figure 7.36 Basic I²S Mode Illustrated



Right Justified (RJ) Mode:

In RJ mode, data starts (r/2 – n) SCLKs after the LRCK edge (Offset = r/2 – n, where r = number of SCLKs per LRCK, n = number of bits/sample).

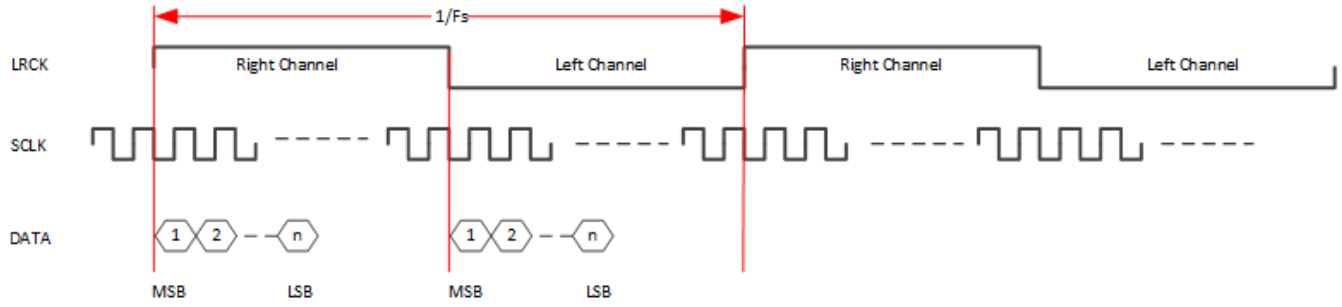
Figure 7.37 RJ Mode Illustrated



Left Justified (LJ) Mode:

In LJ mode, data starts 0 SCLKs after the LRCK edge (Offset = 0).

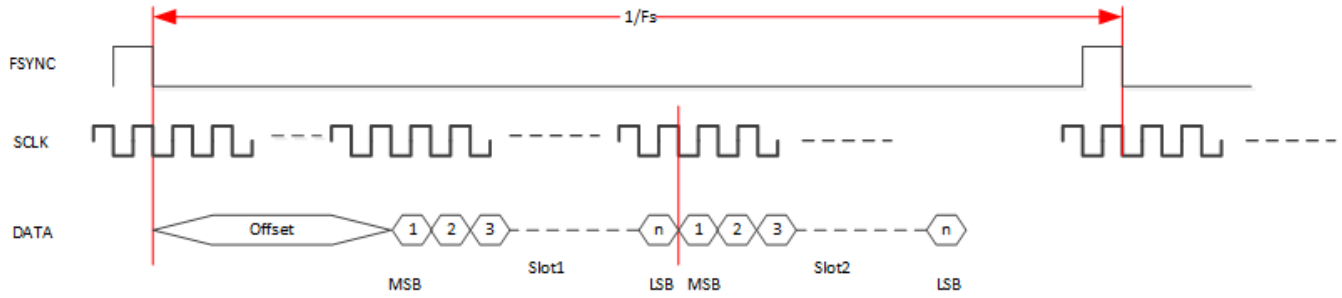
Figure 7.38 LJ Mode Illustrated



FSYNC Modes

The width of the FSYNC, the Offset value, number of slots and number of SCLKs per 1/Fs are all configurable. The slots are always contiguous.

Figure 7.39 FSYNC Mode Illustrated



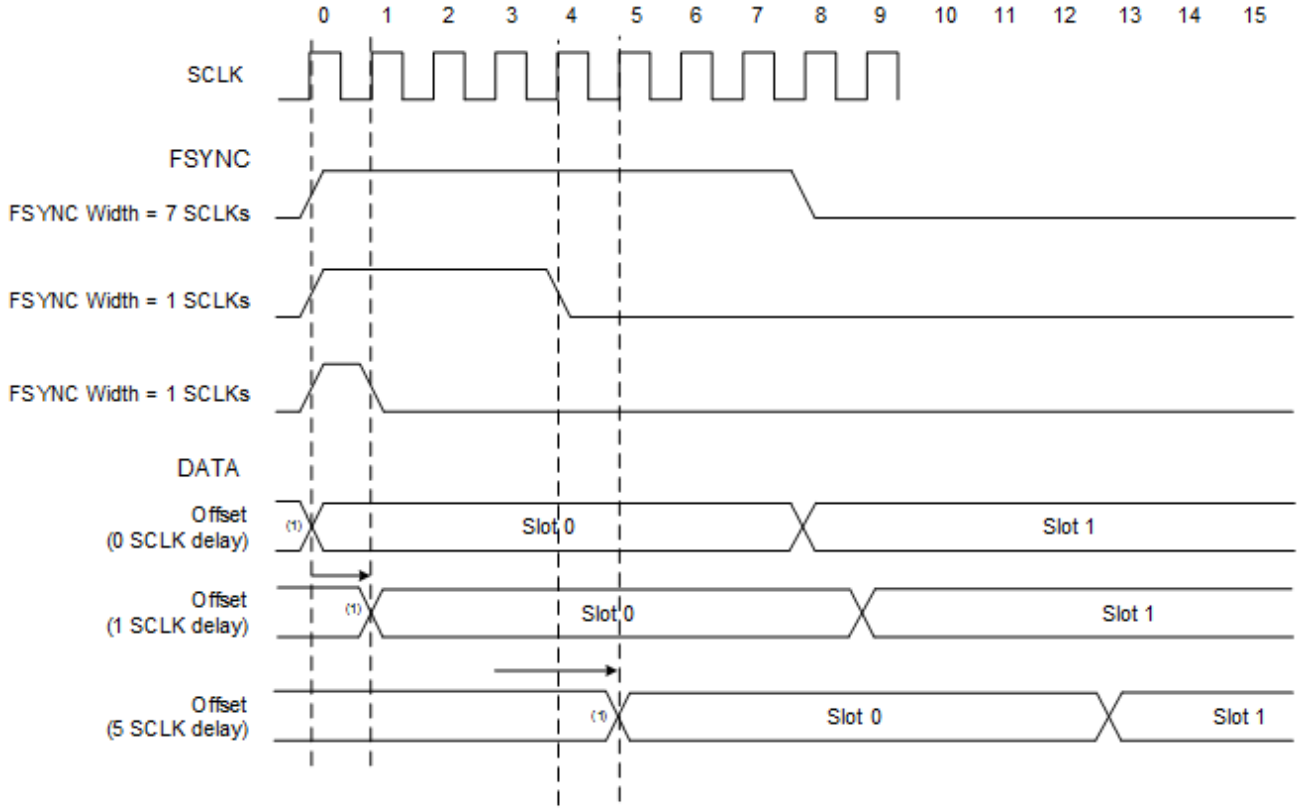
The width of the FSYNC, the Offset value, number of slots (n), and number of SCLKs per 1/Fs are configurable. But the slots are always be contiguous. Besides these, there is also the concept of "Slot Enables" which is used to specify which slots within one FSYNC actually contain data. The table below shows register fields corresponding to the parameters mentioned above.

Table 7.86 FSYNC Register Fields and Their Corresponding Parameters

Parameter Name	Register Field Name
FSYNC width	I2S_CTRL_0_FSYNC_WIDTH
Offset	I2S_AXBAR_RX_CTRL_0_DATA_OFFSET I2S_AXBAR_TX_CTRL_0_DATA_OFFSET
Number of slots	I2S_SLOT_CTRL_0_TOTAL_SLOTS
Number of sclks per 1/Fs	I2S_TIMING_CHANNEL_0_BIT_CNT
Slot Enables	I2S_AXBAR_RX_SLOT_CTRL_0_SLOT_ENABLES

The following set of diagrams shows the timing relationship between FSYNC, SCLK, and DATA lines for some sample parameter value sets.

Figure 7.40 Relationship between SCLK, FSYNC, and DATA for Different FSYNC Widths and Offset Values



1. Last bit of the last slot of the previous frame. No gap is allowed between this bit and the first bit of slot 0.

Sometimes CODEC/MODEM manufacturers specify a unique set of values for these parameters and refer to it using proprietary names like PCM mode, NW mode, etc. The following diagram shows the relationship between SCLK, FSYNC, and DATA lines for these modes that the audio hardware team has encountered in the past.

These names are proprietary and are not standardized. So it is possible that two CODEC manufacturers have DSP mode, but they each specify completely different set of values for the FSYNC mode parameters. Hence, these following diagrams need to be taken as examples that can help you understand what set of parameter values result in what kind of timing diagrams, that is instead of literally taken as *the* set of parameter values/timing diagrams for the various mode names listed for them.

Always refer to the CODEC/MODEM/external chip datasheet to find out what kind of relationship is expected between SCLK, FSYNC, and DATA lines in the I^2S and accordingly use the following examples as a knowledge base and figure out the correct parameter configuration.

Figure 7.41 Proprietary FSYNC Modes – TDM Mode

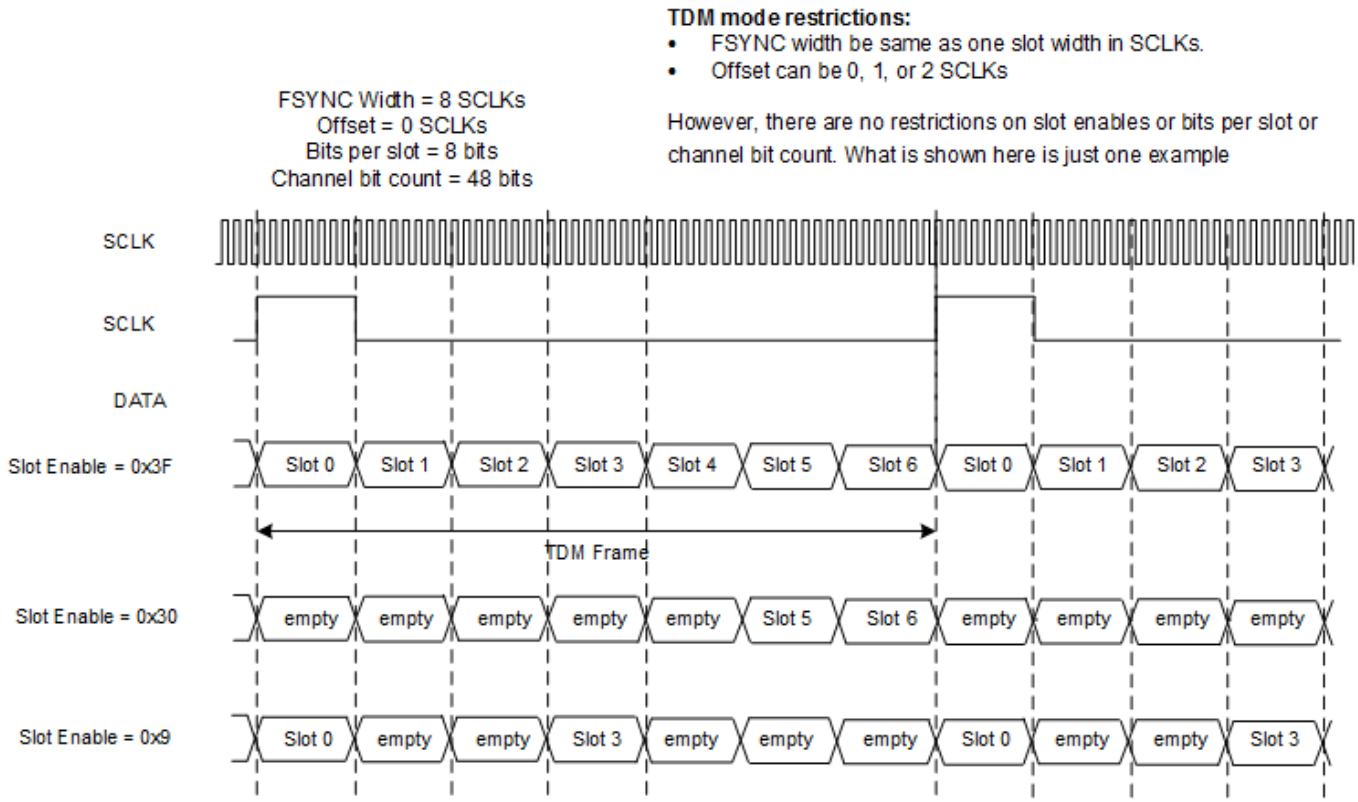


Figure 7.42 Proprietary FSYNC Modes – NW Mode

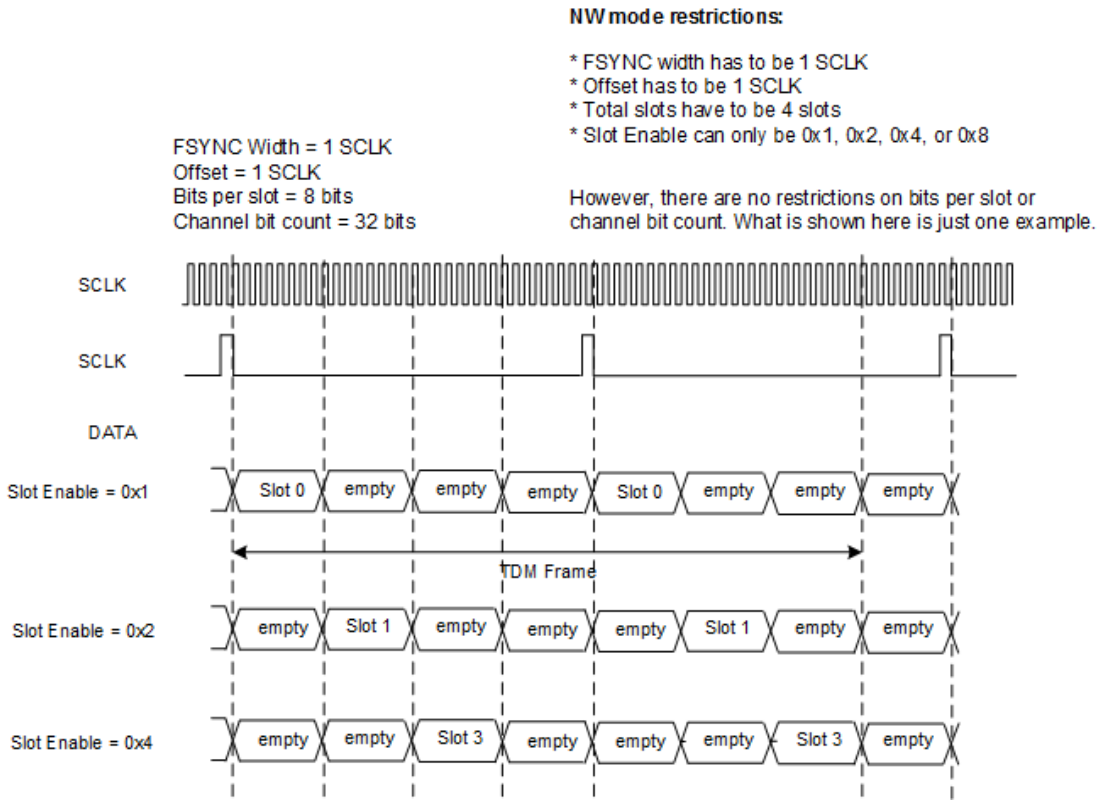


Figure 7.43 Proprietary FSYNC Modes – DSP Mode

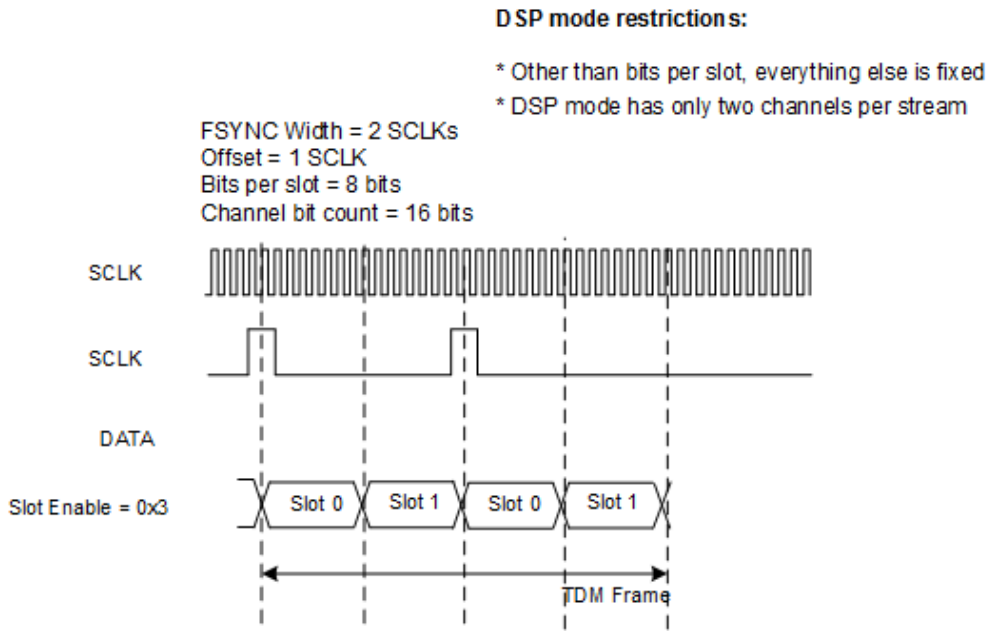
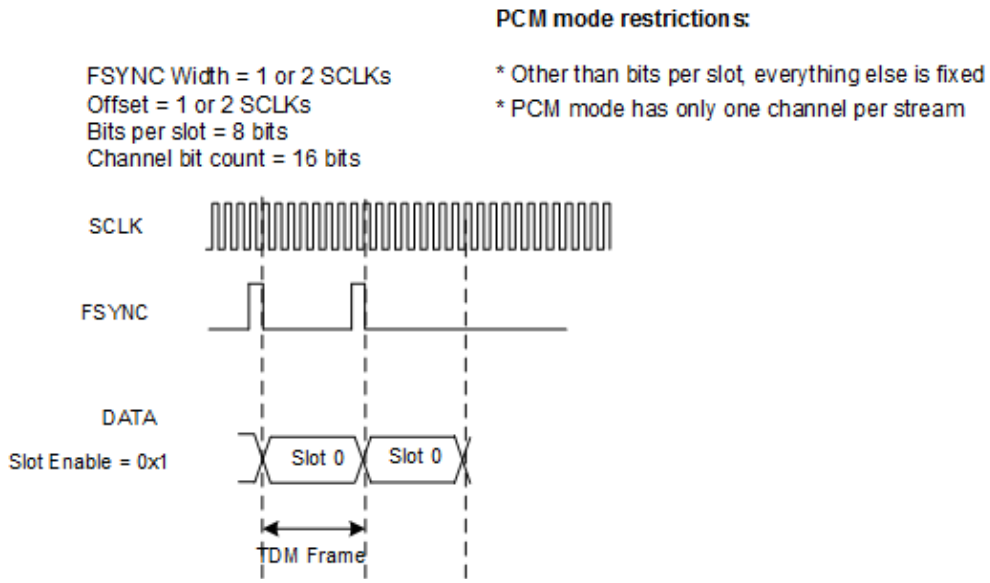


Figure 7.44 Proprietary FSYNC Modes – PCM Mode



7.7.2.2.3 Digital MIC Controller

The Digital MIC (DMIC) Controller is used to interface with Pulse Density Modulation (PDM) input devices. The DMIC controller implements a converter to convert PDM signals to PCM (Pulse Code Modulation) signals. From the signal flow perspective, the DMIC can be viewed as a PDM receiver.

DMIC Features

- Sample rate support: 8 to 48 kHz
- Input PCM bit width: 16 to 24 bits
- Oversampling Ratio: 64, 128, 256
- 24-bit fixed output bit width

7.7.2.2.4 Mixer

The Mixer supports mixing up to ten 7.1 channel audio input streams and five outputs (each of which can be a mix of any combination of the 10 input streams). Examples include mixing system tones with voice calls, mixing two audio playbacks to create a transition effect, mixing system announcements with audio playback.

Mixer Features

- Support of mixing up to 10 input streams of 7.1 channel audio each
- Support of five outputs each of which can be a mix of any combination of 10 input streams
- Time ramp-up/ramp-down volume control provided for each stream

- Fixed gain for each stream is also available
- A 32-bit sample counter is provided for each input stream to count the number of samples consumed
- A peak meter for each input stream is available. It can give peak values for non-overlapping frames of samples or can do continuous reset-on-read peak metering

7.7.2.2.5 Audio Multiplexer Block

The Audio Multiplexer (AMX) block can multiplex up to four input streams of up to 16 channels each and generate an output stream of up to 16 channels

A byte RAM helps to form an output frame by any combination of bytes from the four input frames

Two modes for data synchronization between input frames are available. This feature only starts the data transfer. Once data transfer starts (if at least one frame is sent out), the AMX works in “Wait for all mode” (if auto disable mode is NOT enabled)

- Waiting for All mode:
  At the beginning, wait for all enabled input streams to have data before forming the very first frame.
- Waiting for Any mode
  Start whenever data is available in any one of the enabled input streams.

In either mode, once the first output frame is sent out, AMX always waits for all active streams to have data available before forming and sending subsequent frames.

The AMX supports the Auto disable and Auto enable feature of IDLE streams.

7.7.2.2.6 Audio Demultiplexer Block

The Audio Demultiplexer (ADX) block takes an input stream with up to 16 channels and demultiplexes it into four output streams of up to 16 channels each.

A byte RAM helps to form output frames by any combination of bytes from the input frame. Its design is identical to that of the byte RAM in the AMX except that the data flow direction is reversed.

The ADX provides an automatic stream activation/deactivation mechanism.

7.7.2.2.7 Sampling Frequency Converter

The Sampling Frequency Converter (SFC) converts the sampling frequency of the input signal from one frequency to another.

The SFC is an AXBAR client and hence interacts with other AHUB modules using the Audio Client Interface (ACIF) protocol.

SFC Features

- Support of sampling frequency conversion of streams of up to two channels (stereo)
- Very low latency with the maximum latency of < 125 µs
- Support of the following frequency conversions

Table 7.87 Supported Frequency Conversions (in kHz)

FsIn/ FsOut	8	11.025	16	22.05	24	32	44.1	48	88.2	96	176.4	192
8	Bypass	x	x	x	x	x	x	x	x	x		
11.025	x	Bypass	x	x	x	x	x	x	x	x		
16	x	x	Bypass	x	x	x	x	x	x	x	x	x
22.05	x	x	x	Bypass	x	x	x	x	x	x	x	x
24	x	x	x	x	Bypass	x	x	x	x	x	x	x
32	x	x	x	x	x	Bypass	x	x	x	x	x	x
44.1	x	x	x	x	x	x	Bypass	x	x	x	x	x
48	x	x	x	x	x	x	x	Bypass	x	x	x	x
88.2	x	x	x	x	x	x	x	x	Bypass	x	x	x
96	x	x	x	x	x	x	x	x	x	Bypass	x	x
176.4			x	x	x	x	x	x	x	x	Bypass	x
192			x	x	x	x	x	x	x	x	x	Bypass

7.7.2.2.8 Audio Flow Controller

The Audio Flow Controller (AFC) can control the flow of traffic between two I^2S interfaces running at different clocks that are up to 100 ppm apart.

The AFC implements high fidelity interpolation and decimation algorithms that compensate for clock differences.

It can control traffic flow anywhere in the audio route, even to the inputs of AMX blocks.

The AFC can handle burst traffic from SFCs.

7.7.2.2.9 Output Processing Engine

The Output Processing Engine (OPE) has scalable number of BiQuad stages to support stereo, 5p1, and 7p1 channels meeting Ultra-Low Power (ULP) audio requirements.

7.7.2.2.10 Master Volume Control

The Master Volume Control (MVC) provides gain or attenuation to a digital signal path. It can be used in input or output digital signal path for per-stream volume control as well as master volume control.

The MVC block has one input and one output. The input digital stream can be mono- or multi-channel (up to 7.1 channels) stream. The output digital stream has the same format as the input stream. Therefore, the sample rate, number of channels, number of bits per sample in the output stream are the same as those in the input stream. An independent mute/unmute control is also included in the MVC block. In addition, whenever the gain or mute setting is changed, the gain applied in digital volume control block is ramped up or down to the new setting. The parameters of the ramp, such as the duration and ramp-curve are programmable. In addition to linear ramp this design also provides hardware acceleration for the Windows audio curve type fade. The windows audio curve type fade is typically implemented using a programmable processor.

The MVC block also has a built-in peak meter detector. Peak meter detector is used as a feedback to the user. It is located after volume application and provides the peak output data for each channel. The duration used for the peak meter calculation is programmable.

MVC Features

- Programmable range and steps for volume control
- Programmable Curve Ramp for volume control
- Independent mute/unmute controls
- Default gain values
- Peak meter detector

7.7.2.2.11 Digital Speaker Controller

The Digital Speaker Controller (DSPK) converts the multi-bit Pulse Code Modulation (PCM) audio input to oversampled 1-bit Pulse Density Modulation (PDM) output. From the signal flow perspective, the DSPK can be viewed as a PDM transmitter that up-samples the input to the desired sampling rate by interpolation then converts the oversampled PCM input to the desired 1-bit output via Delta Sigma Modulation (DSM).

DSPK Features

- Sample rate support: 8 to 48 kHz
- Input PCM bit-width: 16 to 24 bits
- Oversampling Ratio: 32, 64, 128, 256
- Passband frequency response: <= 0.5 dB peak-to-peak in 10 Hz to 20 kHz range
- THD+N: <= -80 dB @ -10 dBFS
- Dynamic Range: >= 105 dB

7.7.2.2.12 Arbitrary Sample Rate Converter

Audio systems may require high fidelity sample rate conversion because of the plurality of the audio sources. The sample rate may not be known at the stream set-up time, or is potentially time varying. In addition, ratio between input and output sample rate can be any arbitrary number and the input and output clocks could be derived from asynchronous clocks. For this, an Arbitrary Asynchronous Sample Rate Converter (HQ-AASRC) is required. AASRC consists of two parts. First part is the High Quality Arbitrary Sample Rate Converter (HQ-ASRC) or ASRC module. Arbitrary Sample Rate Converter (ASRC) can convert the input samples at input sample rate (fs_in) to output samples at output sample rate (fs_out). This design can handle over a wide-range of sampling-rate ratios from 1:24 to 24:1. Second part is the Ratio-Estimator module. This block estimates the ratio fs_in/fs_out. Ratio-Estimator can be implemented in hardware using the clocks to derive the sampling rate ratios or it can be implemented in software using the input and output buffer fullness.

7.7.2.2.13 ASRC Features

- Support of input sample rates in the range 8 to 192 kHz
- Support of output sample rates in the range 8 to 192 kHz
- Support of any sampling ratio from 1:24 to 24:1
- Provision of THD+N (0 dBFS @1 kHz, 24-bit samples), i.e., 140 dB or better
- 0.25 dB peak-to-peak Passband Ripple (DC to 0.4535 Fs)
- Linear Phase Response
- Support of input formats up to 16, 24, and 32 bits
- Support of up to six streams up to 16 total channels
- Configurable channel assignment
  Examples: 1x8 channel + 2x2 channel, 6x2 channel, 1x8 channel + 1x4 channel, etc.
- Worst-case ASRC input to output processing latency to be under ~10 mS in I/O to I/O transfers
  (This mainly involves buffering delay for ASRC operations.)

7.7.2.2.14 Audio Ratio Detector

The Audio Ratio Detector (ARAD) has several lanes, each of which has the capacity to find the frequency/time period ratio between any two of the I/O clocks in APE. This has many applications including sample rate conversion without software configuration of the ratio, ppm compensation, etc. While determining the ratio between two clocks, ARAD smooths out one-OFF abnormalities in the clocks so that the ratio value isn't affected by spurious changes in the input clocks.

7.7.2.2.15 Audio Direct Memory Access Interface

The Audio Direct Memory Access (ADMA) interface is a gateway in the AHUB for facilitating DMA transfers between memory and all of its clients. Basically it acts as a bridge to convert 32-bit full duplex AXI protocol to 1-bit DVS protocol with sufficient buffering for supporting seamless data transfers.

ADMA Interface Features

ADMAIF is the interface between ADMA and AHUB. Each ADMA channel that sends/receives data to/from AHUB must interface through an ADMAIF channel. ADMA channel sending data to AHUB pairs with an ADMAIF Tx channel. ADMA channel receiving data from AHUB pairs with an ADMAIF Rx channel. Buffer size is configurable for each ADMAIF channel.

- 4 KiB of buffering per direction for seamless DMA transfers
- Dynamic or software configurable buffer allocation based on bandwidth and latency needs of each channel
- A maximum of 20 channels can be configured per direction for audio samples transfer between memory and AHUB clients
- 32-bit full duplex AXI slave interface to communicate with ADMA for DMA transfers
- Up to 16 words of burst size
- APB slave interface for register programming
- 1-bit ACIF interface per channel to communicate with other AHUB clients through AXBAR
- Samples transfer through both DMA and register access (PIO mode)
- Support of packing (8 or 16 bits to 32 bits) in upstream direction and unpacking (32 bits to 8 or 16 bits) in downstream direction

7.7.2.3 ADMA

The ADMA caters to the DMA needs of APE clients. It performs audio samples transfer among system memory, internal memory/ARAM, and AHUB through multiple unidirectional channels. A unidirectional channel is a point-to-point virtual connection for facilitating blocks of data movement with minimal CPU intervention. ADMA is plugged into the APE with the interfaces AXI interconnect for datapath, AHOST/REGDEC for programming interface, and AHUB for DMA requests originating from its internal clients.

7.7.2.3.1 ADMA Features

- Flow controlled (Memory ↔ AHUB or AHUB ↔ AHUB) and non-flow controlled (Memory ↔ Memory) DMA transfers
  DMA requests based on FIFO status of the requesters in case of flow controlled transfers

- Two AXI Master interfaces
  (One of them dedicated to the transferring of data to and from AHUB and the other one is hooked up to ACONNECT for the transfers to and from ARAM/DRAM)
- Per channel pre-mask Interrupt for indicating transfer completion to the AGIC
  AGIC routes the post-mask Interrupt to ADSP/CPU
- 28-bit word-aligned transfer size that can be configured up to 1 GiB
- 30-bit word-aligned source and target addresses
- Page or 4 KiB boundary alignment
- Per channel configurable burst size up to 16 words for all transfers
- Auto initialization of transfer size, source and target addresses on completion of programmed words transfer in continuous mode
- Scatter/Gather through linked list for accessing non-contiguous memory regions
  Software may use this feature in two scenarios:
- When requiring streaming of audio samples spread across the memory
- When requiring less intervention from CPU

  CPU need not attend the end of Interrupts transfer from ADMA on a regular basis. It can occasionally write a bunch of descriptors into memory and attend other high priority tasks.

  Descriptor is a set of four registers (SOURCE_ADDR, TARGET_ADDR, TRANSFER_COUNT and NEXT_DESCRIPTOR_ADDR) located in consecutive 32-bit entries of DRAM/ARAM starting with SOURCE_ADDR. ADMA fetches next descriptor (SOURCE_ADDR, TARGET_ADDR, TRANSFER_COUNT and NEXT_DESCRIPTOR_ADDR) pointed to by the "NEXT_DESCRIPTOR_ADDR" of current descriptor for the next transfer while current transfer is in progress and these values are loaded at the end of the current transfer.

  Initially, it is loaded by software while setting up the channel and then automatically loaded by hardware prior to end of each block transfer to enable seamless streaming. ADMA keeps fetching the descriptor from ARAM/DRAM pointed to by "NEXT_DESCRIPTOR_ADDR" until either software disables the channel or "NEXT_DESCRIPTOR_ADDR" register value of current descriptor is "0".
- Configurable address wrapping window in multiples of burst size for source and target in all modes
- Up to eight buffers in all three transfer modes (linked list, once and continuous modes)
  - Source and target addresses initialized on completion of all programmed buffers transfer, then Interrupt is generated on completion of every buffer transfer in continuous and linked list modes
  - Interrupt generated on every buffer completion and channel would be disabled on completion of all buffers in once mode
- Up to eight outstanding reads and writes

  - ADMA allowing a maximum of eight reads to source memory and that many writes to the target memory to be outstanding at any given point of time
  - ADMA issuing read/write requests to ARAM/DRAM by mapping channel number to the ARID/AWID of AXI interface so that requests from different channels can pass each other and requests from same channel are handled in order or on first-come-first-serve basis en route to the target elements
- Weighted Round Robin arbitration scheme for selecting read requests to AHUB and Memory
  - Each DMA channel programmed to a weight/request count so that it gets the grant for issuing that many requests back-to-back.
    (This is applicable for all flow-controlled (AHUB) and non-flow-controlled transfers (ARAM/DRAM).)
  - Separate WRR arbiters used for AHUB and Memory Reads to enable un-interrupted full duplex transfer
- Configurable triggers for initiating DMA transfer
  - Triggers for each channel mapped to end of channel Interrupt of all other DMA channels so that software can choose one of them to instruct the channel to initiate the transfer upon assertion of that particular Interrupt
- Cut-through data transfers
- ADMA converting the returned data for the read requests issued to source element into write requests to the target element and forwarding them on AXI write channels in cut-through mode
- 32 channels where each channel can be configured for DMA transfer in any of four possible directions
  - MEMORY-to-AHUB
  - AHUB-to-MEMORY
  - AHUB-to-AHUB
  - MEMORY-to-MEMORY
- Threshold-based flow-control for VI/I^2C triggered DMA transfer
- ADMA Channel Virtualization
  - Virtualization of ADMA channel registers in to four 64K pages
  - Separation of ADMA Config registers into another 64K pages
- APR region and ADAST support
- Preventing secured data from writing into non-APR region
  - Unsecure ADMA channel (APR disabled) blocking the writing of secure data (APR set) to memory
- One ADMA error Interrupt per OS page
- Timestamp capture support
  - 32-bit Timestamp (TSC) captured in timestamp buffer in memory or in register
    - Timestamp capture at DMA transfer done
    - 32-bit wraps around about every 2 minutes

    - Per-channel timestamp buffer and register
- Programmable outstanding transactions cap per channel
- Burst size from 1 to 16 words (VI/I^{2}C). 1, 2, 4, 8, and 16 for other transfers. Transaction type based on Req Sel

7.7.2.3.2 ADMA Descriptor Format

The ADMA Descriptor is as follows:

Table 7.88 ADMA Descriptor Format

32-bit Offset	Name	Description
0	source_addr	Address of the Source Buffer must be 4-byte aligned. Source Buffer could be in DRAM or ARAM.
1	target_addr	Address of the Target Buffer must be 4-byte aligned. Target Buffer could be in DRAM or ARAM.
2	transfer_count	Value in Transfer_count must be a multiple of 4 (4 bytes).
3	next_descriptor_addr	Address of the next descriptor must be 4-byte aligned. Next descriptor could be in DRAM or ARAM. (Address == 0) indicates last Descriptor in the linked list.

The address fields, like all the addresses in APE, are only 32 bits wide. This limit applies to both ADSP and ADMA. To access more than 32 bits, there are two options: either use the APE AST's (ACAST or ADAST) to re-map to more than 32 bits, or use SMMU to map to higher than 4G space.

7.7.2.4 REGDEC

REGDEC provides the host interface for ADMA and AHUB in the APE. REGDEC provides only a direct register access path from ADSP/CPU to ADMA/AHUB. Command DMA and command synchronization features in AHOST are not supported by REGDEC.

REGDEC receives register requests targeted to ADMA and AHUB, decodes the requests, and forwards the requests to the corresponding REGDEC clients.

7.7.2.4.1 REGDEC Features

- AXI3 interface to ACONNECT
- Host controller client interface to AHUB and ADMA
- Direct Register Access for AHUB and ADMA
- Frequency: 100 MHz

7.7.2.5 ACONNECT

ACONNECT provides the backbone fabric for APE. Both register (control) and memory accesses are routed through ACONNECT. ACONNECT is implemented using Arm IP NIC-400. ACONNECT supports the AXI3 and APB bus protocols.

7.7.2.6 Audio RAM and Audio Memory Controller

Audio RAM (ARAM) provides a low latency local ARAM to ADSP processor and other modules. Sixteen (16) 32-bit EVP registers provide a vector table at reset.

7.7.2.6.1 Audio Memory Controller Features

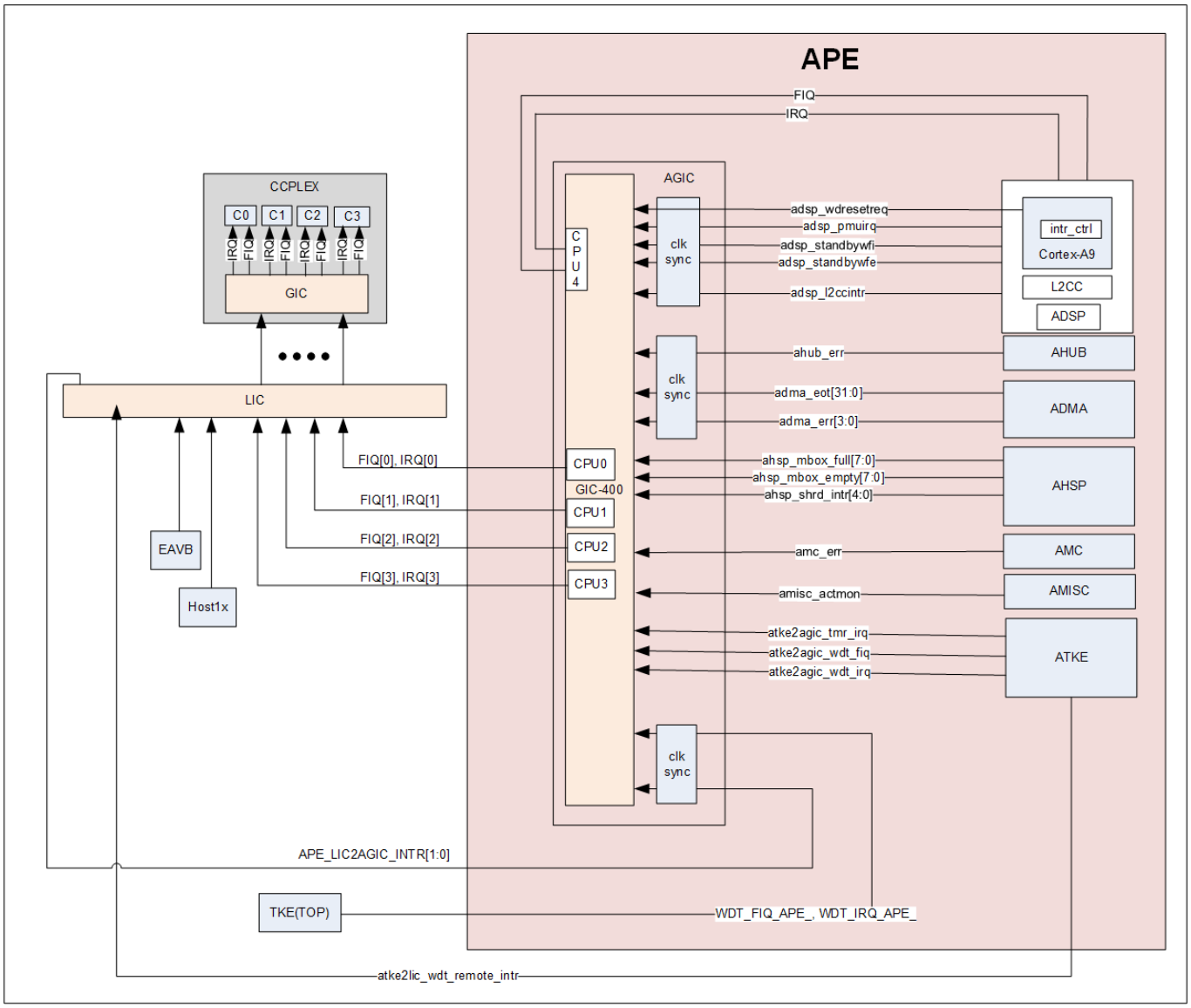
- AXI3 bus interface to ACONNECT
- 64-bit Read and Write data channels
- 16 32-bit EVP registers
- EVP Registers can be locked by sticky bit
- 96 KiB ARAM

7.7.2.7 APE Interrupt Controller

The APE Interrupt Controller (AGIC) serves as the Interrupt controller in the APE. It aggregates the Interrupt lines from all the APE modules and distributes them to ADSP and the system.

AGIC is an implementation of the Arm GIC-400 Generic Interrupt Controller, revision r0p1. Details of the GIC-400, including the technical reference manual, are available from Arm and may be downloaded from their website, http://arm.com.

Figure 7.45 AGIC Functional Diagram



7.7.2.7.1 Shared Peripheral Interrupts

There are two Interrupt lines (FIQ/IRQ) per channel. Interrupts that are mapped to FIQ should use group 0. Interrupts that are mapped to IRQ should use group 1. Group 0 is recommended for higher priority Interrupts.

Table 7.89 Interrupts Signals to GIC

Signal	ID	I/O	Edge/Level	Description
(not relevant)	31:0	I	Level	Software-Generated Interrupts (SGI's)

Signal	ID	I/O	Edge/Level	Description
adma_eot[31:0]	63:32	I	Level	ADMA Channel End of Transfer Interrupt
ahsp_mbox_full[7:0]	71:64	I	Level	AHSP Mailbox Full Interrupt
ahsp_mbox_empty[7:0]	79:72	I	Level	AHSP Mailbox Empty Interrupt
ashp_shrd_intr[4:0]	84:80	I	Level	AHSP Shared Interrupt
adsp_pmuirq	85	I	Level	ADSP/PTM Performance Monitoring Unit Interrupt
adsp_wdresetreq	86	I	Edge	ADSP Watchdog Timer Reset Request
adsp_l2ccintr	87	I	Level	ADSP L2 Cache Controller Interrupt
ahub_err	88	I	Level	AHUB Error Interrupt
amc_err	89	I	Level	AMC Error Interrupt
adma_err[3:0]	93:90	I	Level	ADMA Error Interrupts
adsp_standbywfi	94	I	Level	ADSP Standby WFI. ADSP in idle mode. Waiting for Interrupt
adsp_standbywfe	95	I	Level	ADSP Standby WFE. ADSP in idle mode. Waiting for Event
adsp_ctiirq	96	I	Level	Interrupt request by Cortex-A9 debug trigger interface
ape_actmon	97	I	Level	Activity monitoring on ADSP. This Interrupt is from AMISC
Reserved	99,98	I	Level	Reserved
Lic2ape_int[1:0]	101,100	I	Level	LIC to APE Interrupts
Reserved	102,103	I	Level	Reserved
i2c_<1,3,8>_int	104,105,106	I	Level	I^2C <1,3,8> Interrupt
shsp2ape_db	107	I	Level	System Door Bell Interrupt
top_wdt_fiq	108	I	Level	Top WDT Fiq Interrupt
top_wdt_irq	109	I	Level	Top WDT Irq Interrupt
atke_tmr_irq[3:0]	113:110	I	Level	ATKE Timer Irq Interrupt
atke_wdt_fiq	114	I	Level	ATKE WDT Fiq Interrupt
atke_wdt_irq	115	I	Level	ATKE WDT Irq Interrupt
atke_wdt_error	116	I	Level	ATKE WDT error Interrupt
Reserved	127:117	I	Level	Reserved

Notice that ID assignment of SPI starts from 32. Asynchronous Interrupt signals are synchronized in AGIC before sending to GIC-400. Signal adsp_wdresetreq is edge-triggered. The remaining APE Interrupts are level-triggered.

7.7.2.7.2 AGIC Features

- 32-bit AXI-4 compliant slave interface to ACONNECT
- Secure access to AGIC registers

7.7.2.7.3 Programming Sequence

Identifying the Supported Interrupts

This section discusses how to discover which Interrupts are supported.

1. Read GICD_TYPER. The ITLinesNumber field identifies the number of implemented GICD_ISENABLERns.
2. Write to GICD_CTLR to disable forwarding of Interrupts from distributor to the CPU interfaces
3. For each implemented GICD_ISENABLERn, starting with GICD_ISENABLER0:
   - Write (GICD_ISENABLERn = 0xffffffff).
   - Read (GICD_ISENABLERn). Bits that read as 1 correspond to supported Interrupt IDs.

Discovering Permanently Enabled Interrupts

1. Write (GICD_ICENABLERn = 0xffffffff) to disable all Interrupts that can be disabled.
2. Read (GICD_ICENABLERn).
   Bits that read as 1 correspond to Interrupts that are permanently enabled.
3. Write (GICD_ISENABLERn[i] = 1),
   where i corresponds to Interrupts that must be re-enabled.

Routing an SPI Interrupt to CPU/ADSP through FIQ using Group 1

1. Write (GICD_ICENABLERn = 0xffffffff) to disable all Interrupts that can be disabled.
2. Write (GICD_CTLR = 0x2) to enable Grp1 Interrupts.
3. Write to Interrupt Configuration Registers, GICD_ICFGRn, to configure Interrupt type (level or edge).
4. Write (GICD_IGROUPR[i] = 1) to set the intended Interrupt to Group1,
   where i corresponds to the intended Interrupt position.
5. Write to Interrupt Priority Registers, GICD_IPRIORITYRn, to assign Interrupt priority if necessary.
6. Write to GICD Priority Masked Interrupt, GICD_PMR, to set the Interrupt mask level.
7. Write to Interrupt Processor Target Registers, GICD_ITARGETSRn, to direct the Interrupt to CPU.

8. Write to Interrupt Set-Enable Register, GICD_ISENABLERn, to enable the Interrupts.
9. Enable CPU Interface Control Register (GICC_CTLR) of the destined CPU.
   AGIC is based on generic Interrupt controller Arm IP GIC400. It serves as the APE Interrupt controller, which detects, manages, and distributes Interrupts. The GIC-400 complies to AMBA AXI4 protocol and Version 2 of the Arm GIC Architecture Specification. The GIC-400 implements the GICv2 Security Extension. GIC does not support virtual extension. The feature is implemented in GIC-400 but is not used or verified. Cortex-A9 doesn't support virtualization architecture.
   - FIQEn <= 1.

Routing an SPI Interrupt to CPU/ADSP through FIQ using Group 0

1. Write (GICD_ICENABLERn = 0xffffffff) to disable all Interrupts that can be disabled.
2. Write (GICD_CTLR = 0x1) to enable Grp0 Interrupts.
3. Write to Interrupt Configuration Registers, GICD_ICFGRn, to configure Interrupt type (level or edge).
4. Write (GICD_IGROUPR[i] = 1) to set the intended Interrupt to Group0,
   where i corresponds to the intended Interrupt position.
5. Write to Interrupt Priority Registers, GICD_IPRIORITYRn, to assign Interrupt priority by if necessary.
6. Write to GICD Priority Masked Interrupt, GICD_PMR, to set the Interrupt mask level.
7. Write to Interrupt Processor Target Registers, GICD_ITARGETSRn, to direct the Interrupt to CPU.
8. Write to Interrupt Set-Enable Register, GICD_ISENABLERn, to enable the Interrupts .
9. Enable CPU Interface Control Register (GICC_CTLR) of the destined CPU.
   See Arm GIC Architecture Specification version 2, Table 2-2 and 2-3 for IRQ and FIQ behavior.
   - FIQEn <= 1.
   - EnableGrp0 <= 1. Enable Group 0 Interrupts using the FIQ signal. Group1 Interrupts always uses IRQ signal.

7.7.2.8 Audio Miscellaneous (AMISC)

This module contains ADSP configuration registers, timestamp registers for EAVB synchronization, key slots, ACTMON and debug registers.

7.7.2.8.1 AMISC Features

- Secure ADSP Configuration Registers
- Four 128-bit key slots
- EAVB Timestamp Registers
- APB4 bus interface to ACONNECT
- TSC terminal node

- 32-bit data width

7.7.2.9 ACAST and ADAST

There are two instances of AST: ACAST and ADAST. AST is responsible for looking up 40-bit address and transaction attributes based on the 32-bit address.

7.7.2.9.1 ACAST

DRAM accesses originates from ADSP go through ACAST. There are eight AST regions in ACAST. ACAST looks up transaction attributes using address.

7.7.2.9.2 ADAST

DRAM accesses from ADMA and ASRC go through ADAST. There are eight AST regions in ADAST and is looked up based on APR enable and region ID associated with the transactions.

7.7.2.10 APB-AXI Shim

APB-AXI Shim (AAS) is an APB2AXI shim which converts APB requests into AXI requests. The CPU accesses APE registers through the AAS over the APB bus. The AAS converts the requests to AXI and sends to ACONNECT which forwards the requests to the destination slaves.

7.7.2.10.1 AAS Features

- APB4
  - Support of AxProt[2:0]
  - PSLVERR
- Maximum of one outstanding APB request
- Support of request
  - INCR 1: 32-bit Read/Write

7.7.2.11 APE HSP

Hardware Sync Primitives (HSP) captures shared mailboxes and shared semaphores from the APE 1.0 AMISC module. AHSP is configured as follows:

- Eight Shared Mailboxes
- Four Shared Semaphores
- Five Interrupts

AHSP is an APB slave connected to ACONNECT. It uses 320 KiB of address space.

Refer to the Hardware Synchronization Primitives (HSP) chapter for more information about HSP modules.

7.7.2.12 APE TKE

The TKE module shares with Cortex®-R5 clusters. APE TKE (ATKE) is configured as four timers and one WDT. Although ADSP has its own internal timers, they are dependent on ADSP frequency, which can change due to DFS. When ADSP frequency changes, timer intervals are scaled accordingly which is a software burden.

ATKE provides timers that run on clocks independent of ADSP frequency.

ATKE watchdog timer interrupts the CPU when it expires. Unlike the top TKE watchdog timer, the ATKE watchdog timer does not directly reset the system. The watchdog timer state is not preserved during APE reset. Software ISR that handles watchdog timer Interrupt should save any WDT state needed.

ATKE Interrupts (four timer local interrupts, WDT fiq/irq/error Interrupts) are sent to AGIC which can be routed to CPU or ADSP.

Refer to the Timers chapter of this document for more information about TKE modules.

7.7.3 Programming Guidelines

This section describes the programming guidelines of APE.

7.7.3.1 System Level Programming

1. Starting a use case:
   Enable modules from the tail end of the audio routing towards the source of the audio routing. For example, when the audio samples are fetched from memory, mixed with system tones in the Mixer, and then played at the I^2S, enable I^2S first, followed by the Mixer, and then the ADMA Interface and ADMA.
2. Ending a use case:
   Disable modules from the module nearest to the source towards the destination of the audio routing. In the use case in the example above, disable ADMA first, followed by ADMA Interface, Mixer, and finally the I^2S.
   a. The best practice to end a use case is to disable the module using <module>_ENABLE register.
      Wait for (<module>_STATUS_0.ENABLE_STATUS == 0),
      then move on to disable the next module

b. Alternatively, instead of polling for (<module>_STATUS_0.ENABLE_STATUS == 0), software can choose to use the TX_DONE Interrupt of every module. See the "Using Interrupts" subsection below for more details.
c. Besides the above options a. and b., software can, for practical reasons, wait 20 mS after getting a DMA TRANSFER_DONE Interrupt (as DMA is the source in most audio routings), then use the method described in option a. When an I^2S/DMIC is the source of audio in AHUB, then software can wait 20 mS after checking for (I^2S/DMIC ENABLE_STATUS == 0) then use the method described in option a.

7.7.3.2 Using Interrupts

All AHUB modules have RX_DONE and TX_DONE Interrupts that can be used when ending a use case. The procedure of using these Interrupts is:

1. Write (<module>_XBAR_RX_INT_MASK = 0) and (<module>_XBAR_TX_INT_MASK = 0) to unmask the Interrupt .
2. All the TX_DONE, RX_DONE Interrupts are logically OR'ed and sent to the GIC as the ahub_err Interrupt. Hence the Interrupt service routine for ahub_err Interrupts needs to handle the TX_DONE/RX_DONE Interrupts
   a. Read (AHUB_INTR_STATUS_0_0) and (AHUB_INTR_STATUS_1_0) to check module(s) caused the ahub_err to be asserted.
   b. In the identified module(s), Read (<module>_INT_STATUS) to find out which of the module's Interrupts was fired.

7.7.3.3 Programming Module RAMs

Modules like SFC and Mixer have internal RAMs to hold filter coefficients, configurations, etc. The following are the programming guidelines for writing into and reading from the RAMs:

- Writing to a single RAM location:
  a. Write (<module>_AHUBRAMCTL_<module>_CTRL_x.RAM_ADDR = <address of the RAM entry to be written to>).
  b. Write ((<module>_AHUBRAMCTL_<module>_DATA_x) = <data to be written>).
  c. Write (<module>_AHUBRAMCTL_<module>_CTRL_x.RW = 1).
- Writing sequentially to contiguous RAM locations:
  a. Write (<module>_AHUBRAMCTL_<module>_CTRL_x.SEQ_ACCESS_EN = 1) to enable automatic address advance (increment).
  b. Write (<module>_AHUBRAMCTL_<module>_CTRL_x.RAM_ADDR = <address of the RAM entry to be written to>).
  c. Write (<module>_AHUBRAMCTL_<module>_CTRL_x.ADDR_INIT_EN = 1).
  d. Write ((<module>_AHUBRAMCTL_<module>_DATA_x) = <data to be written>).
  e. Write (<module>_AHUBRAMCTL_<module>_CTRL_x.RW = 1).

f. Repeat d. and e. for the every word that needs to be written to the RAM.

- Reading from a single RAM location:
  a. Write (<module>_AHUBRAMCTL_<module>_CTRL_x.RAM_ADDR = <address of the RAM entry to be read from>).
  b. Write (<module>_AHUBRAMCTL_<module>_CTRL_x,RW = 0).
  c. Read (<module>_AHUBRAMCTL_<module>_DATA_x) to get the data).
- Reading sequentially from contiguous RAM locations:
  a. Write (<module>_AHUBRAMCTL_<module>_CTRL_x.SEQ_ACCESS_EN = 1) to enable automatic address advance (increment).
  b. Write (<module>_AHUBRAMCTL_<module>_CTRL_x.RAM_ADDR = <address of the RAM entry to be read from>).
  c. Write (<module>_AHUBRAMCTL_<module>_CTRL_x.ADDR_INIT_EN = 1).
  d. Write (<module>_AHUBRAMCTL_<module>_CTRL_x.RW = 0).
  e. Read (<module>_AHUBRAMCTL_<module>_DATA_x) to get the data.
  f. Repeat steps 4. and 5. for every word that needs to be read from the RAM.

7.7.3.4 AHUB Sub-modules Soft Reset Sequence

AHUB sub-modules support software reset. Soft reset resets FSM and state registers and flush the ACIF FIFOs. However, the effect of software reset on the sub-module Interrupt status register is not consistent across all the sub-modules. Some modules clear the Interrupt status register on soft reset but others do not. The following sequence can be used to ensure a consistent behavior.

AHUB sub-module soft reset programming sequence:

- Disable all Interrupts of a sub-module.
- Handle and clear all pending Interrupts.
- Issue soft reset to the sub-module.
- Check for soft reset to get cleared (for this it needs client clock).
- Enable the module.

7.7.3.5 Boot Sequence

7.7.3.5.1 Cold Boot Sequence

The following steps are taken before bring ADSP out of reset. (**Note:** Unpower-gate of APE is not necessary because APE is not power-gated by default):

1. CPU programs CAR with desired frequency.
2. CPU enables clocks to APE through CAR module (Note: ADSP clocks must also be enabled before releasing APE resets.)

- Write (CLK_RST_CONTROLLER_CLK_OUT_ENB_APE_SET_0.SET_CLK_ENB_APE = 1) to enable APE clock.
- Write (CLK_RST_CONTROLLER_CLK_OUT_ENB_V_SET_0.SET_CLK_ENB_APB2APE = 1) to enable APB2APE clock.
- Write (CLK_RST_CONTROLLER_CLK_OUT_ENB_ADSP_0.SET_CLK_ENB_ADSPNEON = 1) to enable ADSPNEON clock.
- Write (CLK_RST_CONTROLLER_CLK_OUT_ENB_ADSP_0.SET_CLK_ENB_ADSP = 1) to enable ADSP Clock.

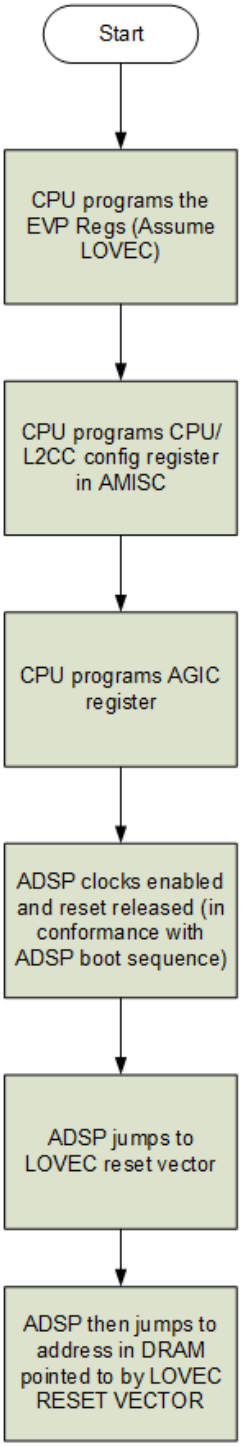
3. Re-enable APE MC clients
   - Write (MC_CLIENT_HOTRESET_CTRL_1.APE_FLUSH_ENABLE = 0) to re-enable APE MC clients.
4. CPU programs CAR and deasserts APE reset to ACONNECT, AAS, AMC, AGIC, AXIBRIDGE, AMISC, REGDEC, AHUB, ADMA.
   - Write (CLK_RST_CONTROLLER_RST_DEV_APE_CLR_0.CLR_SWR_APE_RST = ?) to deassert APE reset.
5. CPU programs AGIC registers to set the Interrupt routing.
6. CPU configures ADSP and L2 cache controller through programming the configuration registers in AMISC.
7. CPU downloads the ADSP boot code to DRAM.
8. Write (CLK_RST_CONTROLLER_RST_DEV_ADSP_CLR_0.CLR_SWR_ADSPINTF_RST = 1) to deassert ADSPINTF_RST.
9. Initialize ADAST/ACAST, EVP, AMC config, ADSP config, and AGIC secure registers in AMISC.
   - Program the EVP registers in AMC. Reset vector of EVP points to the start of the boot code.
     - RegWr32(NV_ADDRESS_MAP_APE_AMC_BASE + AMC_EVP_RESET_VEC_0 , 0xe59ff018);
     - RegWr32(NV_ADDRESS_MAP_APE_AMC_BASE + AMC_EVP_UNDEF_VEC_0 , 0xe59ff018);
     - RegWr32(NV_ADDRESS_MAP_APE_AMC_BASE + AMC_EVP_SWI_VEC_0 , 0xe59ff018);
     - RegWr32(NV_ADDRESS_MAP_APE_AMC_BASE + AMC_EVP_PREFETCH_ABORT_VEC_0, 0xe59ff018);
     - RegWr32(NV_ADDRESS_MAP_APE_AMC_BASE + AMC_EVP_DATA_ABORT_VEC_0 , 0xe59ff018);
     - RegWr32(NV_ADDRESS_MAP_APE_AMC_BASE + AMC_EVP_RSVD_VEC_0 , 0xe59ff018);
     - RegWr32(NV_ADDRESS_MAP_APE_AMC_BASE + AMC_EVP_IRQ_VEC_0 , 0xe59ff018);
     - RegWr32(NV_ADDRESS_MAP_APE_AMC_BASE + AMC_EVP_FIQ_VEC_0 , 0xe59ff018);
     - RegWr32(NV_ADDRESS_MAP_APE_AMC_BASE + AMC_EVP_RESET_ADDR_0 , <<RESET VECTOR>>);
     - RegWr32(NV_ADDRESS_MAP_APE_AMC_BASE + AMC_EVP_UNDEF_ADDR_0 , <<UNDEFINDED VECTOR>>);
     - RegWr32(NV_ADDRESS_MAP_APE_AMC_BASE + AMC_EVP_SWI_ADDR_0 , <<SWI VECTOR>>);

- RegWr32(NV_ADDRESS_MAP_APE_AMC_BASE + AMC_EVP_PREFETCH_ABORT_ADDR_0, <<PREFETCH ABORT VECTOR>>0xc);
- RegWr32(NV_ADDRESS_MAP_APE_AMC_BASE + AMC_EVP_DATA_ABORT_ADDR_0 , <<DATA ABORT VECTOR>>);
- RegWr32(NV_ADDRESS_MAP_APE_AMC_BASE + AMC_EVP_RSVD_ADDR_0 , <<RESERVED>>);
- RegWr32(NV_ADDRESS_MAP_APE_AMC_BASE + AMC_EVP_IRQ_ADDR_0 , <<IRQ VECTOR>>);
- RegWr32(NV_ADDRESS_MAP_APE_AMC_BASE + AMC_EVP_FIQ_ADDR_0 , <<FIQ VECTOR>>);

◦ Program ADAST and ACAS.
◦ Configure AGIC secure registers in AMISC.

10. Write the settings bits in CLK_RST_CONTROLLER_RST_DEV_ADSP_CLR_0 atomically, to deassert ADSP resets:
    ◦ CLR_SWR_ADSPNEON_RST
    ◦ CLR_SWR_ADSPSCU_RST
    ◦ CLR_SWR_ADSPWDT_RST
    ◦ CLR_SWR_ADSPDBG_RST
    ◦ CLR_SWR_ADSPPERIPH_RST
    ◦ CLR_SWR_ADSP _RS
    Note: There should about six ADSP clocks of delay between releasing ADSP resets and releasing of ADSP_INTERFACE_RST.
11. ADSP fetches reset vector from AMC and jumps to the boot code.

Figure 7.46 ADSP Boot Flow



7.7.3.6 Power Management Sequence

7.7.3.6.1 Power Gating

Before power gating, APE states should be saved into DRAM. ADSP L1 and L2 are flushed.

Power Gating Sequence:

1. CPU requests ADSP to power gate. Request could come from mailbox.
2. If ADSP is not ready to be power gated, it sends "not ready" message to CPU. Otherwise, continue.
3. ADSP shuts OFF all activities in AHUB and ADMA. It makes sure there is no outstanding request in ADMA. It is recommended that CPU does not access APE registers after asking ADSP to be power gated and does not send more Interrupts to ADSP. It could unexpectedly interrupt ADSP and divert it from the shut down process.
4. ADSP saves all states and restore information to DRAM.
5. ADSP flushes L1 and L2 cache.
6. ADSP disables Interrupts to itself. Diverts Interrupts to CPU if appropriate.
7. ADSP make sure there are no outstanding activities.
8. ADSP executes WFI/WFE. WFI is recommended because it makes sure there is no outstanding transaction in Cortex-A9.
9. CPU sees the Interrupt from ADSP STANDBYWFI/STANDBYWFE signals. CPU reads the AGIC status register and makes sure STANDBYWFI or STANDBYWFE is asserted.
10. CPU read AMISC register to make sure L2_IDLE and L2_CLKSTOPPED of AMISC_ADSP_STATUS_0 are asserted
11. Flush APE MCCIF
    - Set APE_FLUSH_ENABLE field in register MC_CLIENT_HOTRESET_CTRL_1
    - Poll APE_HOTRESET_STATUS field in register MC_CLIENT_HOTRESET_STATUS_1 for flush done
12. Enable clamps
    - Assert clamps by writing to PMC_IMPL_PART_AUD_CLAMP_CONTROL_0 with (CLAMP = ON)
    - Insert 100 nS delay for clamps to engage.
13. CPU asserts reset through CAR
    - Assert APE Reset. SET_SWR_APE_RST of CLK_RST_CONTROLLER_RST_DEV_APE_SET register
    - Assert ADSP resets. Set the following in the CLK_RST_CONTROLLER_RST_DEV_ADSP_SET register:
      - SET_SWR_ADSPNEON_RST
      - SET_SWR_ADSPSCU_RST
      - SET_SWR_ADSPWDT_RST

- SET_SWR_ADSPDBG_RST
- SET_SWR_ADSPPERIPH_RST
- SET_SWR_ADSP_RST
- SET_SWR_ADSPINTF_RST
- SET_SWR_ADSPCSITEPTM_RST

14. Wait for nine clocks to compensate for reset and clock latencies.
15. CPU disables APE clocks.
    - Disable APE clock. Set field CLR_CLK_ENB_APE in register CLK_RST_CONTROLLER_CLK_OUT_ENB_APE_CLR_0
    - Disable APB2APE clock. Set field CLR_CLK_ENB_APB2APE in register CLK_RST_CONTROLLER_CLK_OUT_ENB_V_CLR_0
    - Disable ADSPNEON clock. Set field CLR_CLK_ENB_ADSPNEON in register CLK_RST_CONTROLLER_CLK_OUT_ENB_ADSP_CLR_0
    - Disable ADSP Clock. Set field CLR_CLK_ENB_ADSP in register CLK_RST_CONTROLLER_CLK_OUT_ENB_ADSP_CLR_0
16. Power gate APE
    - Single Write to PMC_IMPL_PART_AUD_POWER_GATE_CONTROL_0 with:
      - (LOGIC_SLEEP == ON)
      - (SRAM_SLEEP == ON)
      - (SRAM_RET == OFF_)
      - (INTER_PART_DELAY_EN == ENABLE)
      - (START == 1)
    - Poll PMC_IMPL_PART_AUD_POWER_GATE_CONTROL_0 until bit (START == DONE).
    - Read PMC_IMPL_PART_AUD_POWER_GATE_STATUS_0 and verify that:
      - (LOGIC_SLEEP_STS == ON)
      - (SRAM_SLEEP_STS == ON)
      - (SRAM_RET_STS == OFF)
      - If there is a mismatch, flag error.

7.7.3.6.2 Power Ungating

Power Ungating sequence is the same as cold boot. To restore the ADSP context, CPU sends a message to ADSP, with pointers where the context information is saved.

General sequence:

- Set partition power state to PUG
- Enable clocks
- Assert resets (resets might be deasserted during enable clocks. This step ensures the states are reset.)

- Release clamping
- Deassert resets
- Clear MC flush request

Detailed sequence:

1. Unpower-gate on APE if not already in power up state, Single Write to PMC_IMPL_PART_AUD_POWER_GATE_CONTROL_0 with:
   - (LOGIC_SLEEP = OFF)
     (SRAM_SLEEP = OFF)
     (SRAM_RET = OFF)
     (INTER_PART_DELAY_EN = ENABLE)
     (START = 1)
   - Poll PMC_IMPL_PART_AUD_POWER_GATE_CONTROL_0 until bit (START == DONE).
   - Read PMC_IMPL_PART_AUD_POWER_GATE_STATUS_0 and verify that:
     (LOGIC_SLEEP_STS == OFF)
     (SRAM_SLEEP_STS == OFF)
     (SRAM_RET_STS == OFF)
   - If there is mismatch, flag error
2. Repeat Steps 2 and 3 of cold boot sequence to enable clocks.
3. CPU asserts reset through CAR.
   - Assert APE Reset. SET_SWR_APE_RST of CLK_RST_CONTROLLER_RST_DEV_APE_SET register
   - Assert ADSP resets. Set the following in the CLK_RST_CONTROLLER_RST_DEV_ADSP_SET register
     - SET_SWR_ADSPNEON_RST
     - SET_SWR_ADSPSCU_RST
     - SET_SWR_ADSPWDT_RST
     - SET_SWR_ADSPDBG_RST
     - SET_SWR_ADSPPERIPH_RST
     - SET_SWR_ADSP_RST
     - SET_SWR_ADSPINTF_RST
4. Remove clamps.
   - Write to (PMC_IMPL_PART_AUD_CLAMP_CONTROL_0.CLAMP = OFF).
   - Insert 100 nS delay for clamps to engage
5. Repeat Steps 4 to 9, 11, 12 of the Cold Boot Sequence to deassert Reset and initialize APE.
6. CPU sends restore commands, which has pointers to the saved data, to ADSP.

7.7.3.7 SC7 Entry and Exit

Before entering SC7 Entry, APE registers and states are saved to memory. Software device driver is responsible for saving non-secure critical information and restoring them during SC7 exit. BPMP is responsible for saving and restoring secure critical states. It is assumed that BPMP-FW provides the security measure to protect secure sensitive APE contexts. The secure registers and states that require BPMP to save and restore are:

- All EVP registers in AMC (Addr: 0x299_3700 to 0x299 _373c, and 0x299_3000)
  - AMC_EVP_RESET_VEC_0
  - AMC_EVP_UNDEF_VEC_0
  - AMC_EVP_SWI_VEC_0
  - AMC_EVP_PREFETCH_ABORT_VEC_0
  - AMC_EVP_DATA_ABORT_VEC_0
  - AMC_EVP_RSVD_VEC_0
  - AMC_EVP_IRQ_VEC_0
  - AMC_EVP_FIQ_VEC_0
  - AMC_EVP_RESET_ADDR_0
  - AMC_EVP_UNDEF_ADDR_0
  - AMC_EVP_SWI_ADDR_0
  - AMC_EVP_PREFETCH_ABORT_ADDR_0
  - AMC_EVP_DATA_ABORT_ADDR_0
  - AMC_EVP_RSVD_ADDR_0
  - AMC_EVP_IRQ_ADDR_0
  - AMC_EVP_FIQ_ADDR_0

  Followed by writing '1' to EVP_LOCK in AMC_CONFIG_0 to lock
- All AST_CONTROL, AST_STREAMID_CTL, and region 0-7 registers in ACAST and ADAST. Lock bit are properly accordingly.
  - ACAST address range: 0x0299_4000 to 0x0299_41F8
  - ADAST address range: 0x0299_6000 to 0x0299_61F8
- ADSP Configuration registers:
  - MAXCLKLATENCY, CLUSTERID, VINITHI in AMISC_ADSP_CONFIG_0 register.
  - AMISC_ADSP_PERIPHASE_0 register.
  - AMISC_ADSP_L2_CONFIG_0 register.
  - AMISC_ADSP_L2_REGFILEBASE_0 register.
  - AMISC_ADSP_AGIC_CONFIG_0 register
  - Write 1 to ADSP field of AMISC_CONFIG_LOCK_0 to lock

  In summary, address: 0x0299_0004 to 0x0299_0014, and 0x0299_01C.

- All AMISC_KEYSLOTS* registers in AMISC:
  - AMISC_KEYSLOT_NS_0
  - AMISC_KEYSLOT_KEY0_*
  - AMISC_KEYSLOT_KEY1_*
  - AMISC_KEYSLOT_KEY2_*
  - AMISC_KEYSLOT_KEY3_*

AMISC_KEYSLOTS_NS_0 are properly set after keyslot values are restored. Address range: 0x0299_007c to 0x0299_00bf

7.7.3.7.1 SC7 Exit Sequence

Note: Unpower-gate of APE is not necessary because APE is not power-gated by default.

1. CPU programs CAR with desired frequency
2. CPU enable clocks to APE through CAR module (**Note**: ADSP clocks must also be enabled before releasing APE Resets)
   - Enable APE clock. Set field SET_CLK_ENB_APE in register CLK_RST_CONTROLLER_CLK_OUT_ENB_APE_SET_0
   - Enable APB2APE clock. Set field SET_CLK_ENB_APB2APE in register CLK_RST_CONTROLLER_CLK_OUT_ENB_V_SET_0
   - Enable ADSPNEON clock. Set field SET_CLK_ENB_ADSPNEON in register CLK_RST_CONTROLLER_CLK_OUT_ENB_ADSP_0
   - Enable ADSP Clock. Set field SET_CLK_ENB_ADSP in register CLK_RST_CONTROLLER_CLK_OUT_ENB_ADSP_0
3. Re-enable APE MC clients
   - Clear APE_FLUSH_ENABLE field in register MC_CLIENT_HOTRESET_CTRL_1
4. CPU programs CAR and deassert APE reset to ACONNECT, AAS, AMC, AGIC, AXIBRIDGE, AMISC, REGDEC, AHUB, ADMA
   - Deassert APE reset by writing to CLR_SWR_APE_RST field in register CLK_RST_CONTROLLER_RST_DEV_APE_CLR_0
5. Restore EVP, ACAST, ADAST, ADSP configuration, AMISC_ADSP_AGIC_CONFIG_0 and Keyslot registers securely as described above.
6. Software restores all the APE registers and states. This can be done insecurely.
7. Deassert ADSPINTF_RST by writing '1' to CLR_SWR_ADSPINTF_RST of CLK_RST_CONTROLLER_RST_DEV_ADSP_CLR_0
8. Write the settings bits in CLK_RST_CONTROLLER_RST_DEV_ADSP_CLR_0 atomically, to deassert ADSP resets:

- CLR_SWR_ADSPNEON_RST
- CLR_SWR_ADSPSCU_RST
- CLR_SWR_ADSPWDT_RST

- CLR_SWR_ADSPDBG_RST
- CLR_SWR_ADSPPERIPH_RST
- CLR_SWR_ADSP_RST
- CLR_SWR_ADSPCSITEPTM_RST
  Note: There should about six ADSP clocks of delay between releasing ADSP resets and releasing ADSP_INTERFACE_RST)
- ADSP fetches reset vector from AMC and jumps to the boot code

7.7.3.8 AAS Programming Guidelines

The AAS starts functioning right out of reset. No programming is needed. When an error is asserted, read the INT_STATUS register to check for Interrupt status. Write a '1' to INT_STATUS_CLR to clear the corresponding Interrupt status bit. When INT_STATUS is asserted, register ERROR_RESPONSE and ERROR_ADDRESS capture the information about the error status.

7.7.3.9 ACIF Programming Guidelines

The following programming guidelines are common for the ACIF registers in all AHUB modules. The programming guidelines of the respective modules refers to this subsection when it comes to programming the ACIF registers.

1. Set the XBAR_CHANNELS field to the number of channels in the input or output stream of a module, depending on the direction.
2. Set the CLIENT_CHANNELS field to the number of channels of the stream as dealt with inside the module. This setting is different from XBAR_CHANNELS only when a conversion of the number of channels of an audio stream is desired.
3. Similarly, set the XBAR_BITS and CLIENT_BITS fields.
4. If XBAR_BITS and CLIENT_BITS do not match, the EXPAND and TRUNCATE fields are used to determine how to transition from XBAR_BITS to CLIENT_BITS (in the case of receive) or from CLIENT_BITS to XBAR_BITS (in the case of transmit). Expanding always results in the actual data being shifted to the MSB bits and the rest filled with zeros, ones, or some random bits from an LFSR. TRUNCATE can be set to chopping or rounding.
5. For the ADMA interface, if packing of data in the TxCIF is desired, use PACK8_ENABLE or PACK16_ENABLE. Similarly, If unpacking of data in the receive interface is desired, use UNPACK8_ENABLE or UNPACK16_ENABLE.
6. If XBAR_CHANNELS and the CLIENT_CHANNELS are different, the STEREO_CONV field is used to determine how a stereo stream is converted to a mono stream, and the MONO_CONV field is used to determine how a mono stream is converted to a stereo stream.

7.7.3.10 I²S Controller Programming Guidelines

I²S transmit and receive directions can be independently enabled/disabled and controlled.

1. Ensure that the I²S Tx/Rx is in a disabled state (after being used in a previous use case): Check if (I2S_AXBAR_RX/TX_STATUS_ENABLE_STATUS == 0)
2. Configure the I²S Tx/Rx path
   a. Configure CIF parameters (see "ACIF Programming Guidelines" for more details) Populate I2S_XBAR_TX_CIF_CTRL, I2S_XBAR_RX_CIF_CTRL
3. Configure the I²S to operate in the mode of choice
   a. Refer to the "Programming I²S to Operate in Various Modes" section about programming I²S in different modes.
   b. Select the sampling rate indirectly by setting the channel bit count. Use the "I²S Sampling Rate Selection" section to find the correct value for this register Configure I2S_TIMING_CHANNEL_BIT_CNT
4. If I²S module is not already in the enabled state, enable the module
   a. Set (I2S_ENABLE = 1).
5. Enable the I²S receive and/or transmit directions
   a. Set (I2S_AXBAR_RX_ENABLE = 1), (I2S_AXBAR_TX_ENABLE = 1).
6. Ensure that the enabling is complete
   a. Ensure (I2S_AXBAR_RX/TX_STATUS_ENABLE_STATUS == 1).
7. After using the module, disable the module
   a. Set (I2S_AXBAR_RX_ENABLE = 0), (I2S_AXBAR_TX_ENABLE = 0).

7.7.3.10.1 I²S Sampling Rate Selection

The channel_bit_cnt can be calculated using the equation:

channel_bit_cnt = (frequency of bit_clk) / (C * required sampling rate) - 1

Where C is 2 for LRCK modes and 1 for FSYNC modes.

If this calculation returns a fractional value, use the non-symmetry feature of the controller to attain the required sampling rate. In that case, the channel_bit_cnt value should be programmed with an integer that is closest to the fraction, but less than the fraction.

The table below contains examples for common sampling rates with CLK_SOURCE_I2S = 24 MHz:

Table 7.90 Common Sampling Rates (CLK_SOURCE_I2S = 24 MHz)

Sampling Rate	I2S_NEW_TIMING[12:0] (Mark-space Ratio = Left_channel : Right channel)				
	CLK_DIVISOR = 0 BIT_CLK = 24 MHz	CLK_DIVISOR = 1 BIT_CLK = 12 MHz	CLK_DIVISOR = 3 BIT_CLK = 6 MHz	CLK_DIVISOR = 5 BIT_CLK = 4 MHz	CLK_DIVISOR = 7 BIT_CLK = 3 MHz
8 kHz	0x05db	0x02ed	0x0176	0x00f9	0x10bb
	(1500:1500)	(750:750)	(375:375)	(250:250)	(187:188)
32 kHz	0x0176	0x10bb	Not supported	0x103e	Not supported
	(375:375)	(187:188)		(62:63)	
44.1 kHz	0x010f	0x0087	0x0043	0x002c	0x0021
	(272:272)	(136:136)	(68:68)	(45:45)	(34:34)
48 kHz	0x00f9	0x007c	0x103e	Not supported	Not supported
	(250:250)	(125:125)	(62:63)		
96 kHz	0x007c	0x103e	0x101f	Not supported	Not supported
	(125:125)	(62:63)	(31:32)		

Note:

- Ideally, any sampling rate can be generated, either accurately or approximately with any clk_src selected for I^2S, if the clk_divisor and the channel_bit_cnt are programmed accordingly.
- The NON_SYM.EN feature is meant to create the sampling rates approximately by realizing an odd bit rate with a non-50:50 mark/space ratio. However, if the bit rate (2*channel_bit_cnt) itself is a fraction, the resultant sampling rate is not accurate. The entries shown as not supported in the above table are attributed to such a deviation.
- When the channel_bit_cnt is less than 32, the bit_size should be programmed to be less than channel_bit_cnt.

Programming I^2S to Operate in Various Modes

After the module initialization (reset and clock programming), program the corresponding bits of the following registers.

Table 7.91 I^2S Programming for Various Modes

Register	Basic	LJM	RJM	DSP	PCM	NW	TDM
I2S_CTRL_FRAME_FORMAT	0	0	0	1	1	1	1
I2S_CTRL_LRCK_POLARITY	0	1	1	1	1	1	1

Register	Basic	LJM	RJM	DSP	PCM	NW	TDM
I2S_CTRL_CHANNEL_BIT_CNT	# of bit clocks in left channel	# of bit clocks in left channel	# of bit clocks in left channel	(# of bits in left channel + right channel)	# of bit clocks per frame	# of bit clocks per frame	# of bit clocks per frame
TX_DATA_OFFSET	1	0	(CHANNEL_BIT_CNT - BITS_PER_SAMPLE)	1	**0**: Long Fsync **1**: Short Fsync	1	0/1/2
RX_DATA_OFFSET	1	0	(CHANNEL_BIT_CNT - BITS_PER_SAMPLE)	1	**0**: Long Fsync **1**: Short Fsync	1	0/1/2
I2S_CTRL_FSYNC_WIDTH	# of bit clocks in left channel	# of bit clocks in left channel	# of bit clocks in left channel	1	**0**: Short Fsync **1**: Long Fsync	0	Bits per sample (slot size)
HIGHZ_CTRL	0	0	0	0	2	2	0/1/2
EDGE_CTRL	0	0	0	0	1	1	1
I2S_SLOT_CTRL_TOTAL_SLOTS	0	0	0	1	0	3	0 to 15
SLOT_ENABLES	0x01	0x01	0x01	0x03	0x01	**Note 1**	**Note 2**

Note 1: In NW mode, there can only be one active slot. So 0x01, 0x02, 0x04, 0x08 are allowed values.
Note 2: In TDM mode, there is no restriction on what or how many slots are enabled.

The only restriction on the programming sequence is that all other registers must be programmed first before the Tx/Rx channels are enabled (which is done by setting XFER_EN_TX/XFER_EN_RX).

Programming I^2S to Work in Loopback Mode

1. Set the (I2S_CTRL_LPBK = 1).

The rest of the programming is the same as if I^2S is not in the loopback mode.

7.7.3.11 DMIC Programming Guidelines

7.7.3.11.1 Minimum Required Guidelines

1. Ensure that the DMIC is in a disabled state (after being used previously):
   Check if (DMIC_STATUS_ENABLE_STATUS == 0)
2. Configure the DMIC:
   a. Configure TxCIF parameters
      Populate DMIC_XBAR_TX_CIF_CTRL registers
   b. Configure the OSR:
      Populate DMIC_CTRL_OSR register

3. Enable the DMIC module
   Set (DMIC_ENABLE = 1).
4. After using the module, disable the module:
   Set (DMIC_ENABLE = 0).

It is mandatory to program client bits of TxCIF to 24 bit as algorithm works in 24-bit mode. AXBAR bits can be programmed as per requirement.

7.7.3.11.2 Optional Guidelines

While configuring the DMIC controller (step 2 of the section above) one can choose to configure the following optional features:

1. Configure whether to receive the left channel data followed by right channel data or the other way around:
   Use the DMIC_CTRL_LRSEL_POLARITY register field.
2. Configure whether to receive data in both left and right channels or just the left channel or the right channel:
   Use the DMIC_CTRL_CHANNEL_SELECT register field
3. DMIC has three filters in the data path, namely the DC removal filter, Sinc Correction filter and a generic low pass filter. These can be enabled using bits DCR_ENABLE, SC_ENABLE, LP_ENABLE respectively, in the register DMIC_DBG_CTRL_0
   a. Each filter is made up of one or two biquads. Each filter has a filter gain in signed Q23 format configurable using the register <Filter_name>_FILTER_GAIN_0
   b. Associated with each biquad are five registers, namely <Filter_name>_<Biquad_no.>_COEF_0_0, <Filter_name>_<Biquad_no.>_COEF_1_0, <Filter_name>_<Biquad_no.>_COEF_2_0, <Filter_name>_<Biquad_no.>_COEF_3_0, and <Filter_name>_<Biquad_no.>_COEF_4_0 (all in signed Q23 format). They correspond to biquad coefficients b0, b1, b2, a1, a2. Coefficent a0, of course, is assumed to be 1.

7.7.3.12 Mixer Programming Guidelines

7.7.3.12.1 Guidelines for Programming Mixer Inputs

In general, the Mixer's inputs are all independent of each other. The programming guidelines are given for configuring and using a generic input. The guidelines are applicable for any of the Mixer's inputs.

1. Configure the CIF parameters.
   Populate the register MIXER_AXBAR_RX <i>_CIF_CTRL.
2. Make gain application related configurations (see Programming Gain Application below.)
3. Make peak value related configuration
   Configure the register MIXER_AXBAR_RX<i>_PEAK_CTRL
   Note: i = 1, 2, 3, 4, 5, 6, 7, 8, 9, or 10

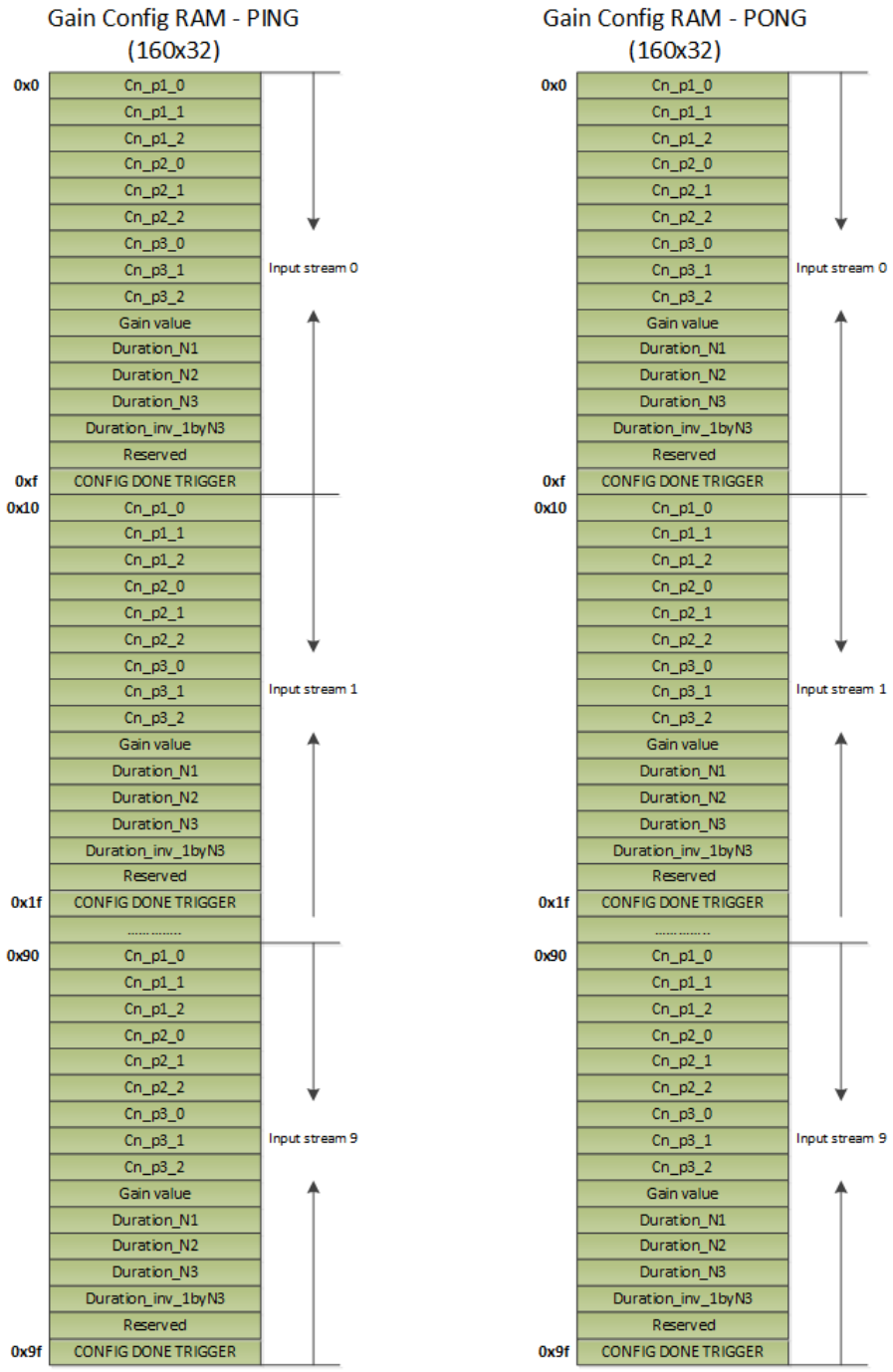
7.7.3.12.2 Guidelines for Programming Mixer Outputs

1. Ensure the output to be used is in a disabled state (After being used in the previous use case)
   Check if (MIXER_AXBAR_TX<j>_STATUS_ENABLE_STATUS == 0).
2. Configure the adder associated with this output by enabling which inputs to be used with this adder
   Populate the register MIXER_AXBAR_TX<j>_ADDER_CONFIG
3. Ensure that the Mixer module is in an enabled state
   Check if (MIXER_STATUS_ENABLE_STATUS == 1)
   a. If not, enable the Mixer module
      Set (MIXER_ENABLE = 1).
   b. Ensure the module is done initializing
      Check if (MIXER_STATUS_ENABLE_STATUS == 1).
4. Enable the Mixer output
   Set (MIXER_AXBAR_TX<j>_ENABLE = 1).
   Note: j = 1, 2, 3, 4, or 5

7.7.3.12.3 Programming Gain Application

All gain-related configurations reside in a RAM as shown below.

Figure 7.47 Gain Configurations in RAM



To write values into this RAM, one must use the AHUBRAMCTL_GAIN_CONFIG_RAM_CTRL, AHUBRAMCTL_GAIN_CONFIG_RAM_DATA. Refer to the Programming Module RAMs subsection for programming guidelines on writing/reading RAM locations. The programming sequence for writing new gain parameters is:

1. Make sure the hardware is not in the middle of updating its internal variables with the old set of gain related parameters:
   Ensure (MIXER_AXBAR_RX<i>_STATUS_CNFG_BUSY == 0), where i = 1, 2, 3, 4, 5, 6, 7, 8, 9, or 10.
2. Program new gain value corresponding to other parameters if needed.
3. Write to 0xf offset register to trigger the hardware to start using a new set of values.

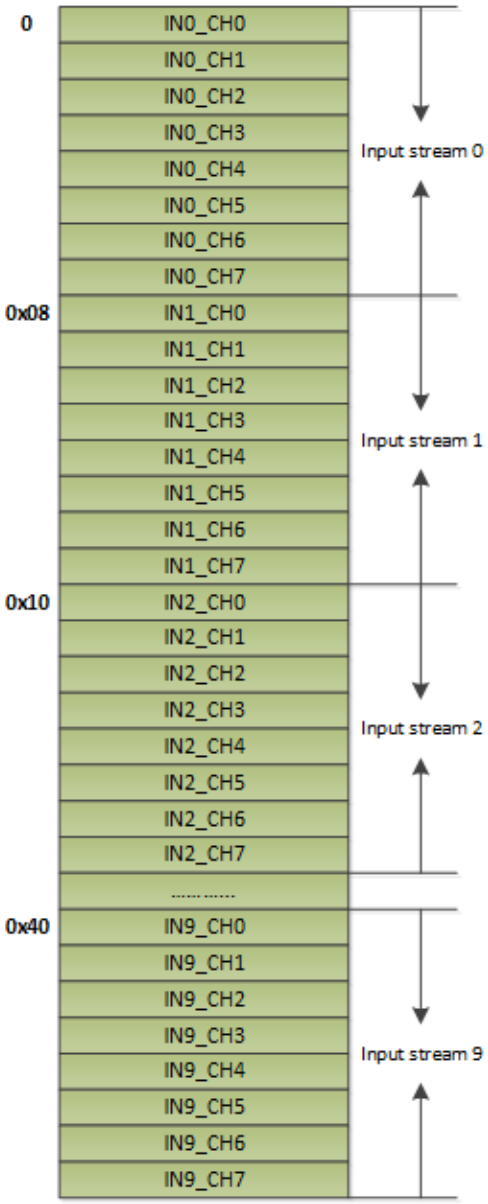
Note:

a. The 'Duration_inv_1byN3' should be calculated by software from the value of 'Duration_N3' (N3) as follows:
   Duration_inv_1byN3 =(1 / N3) * $2^{prescalar}$ * 2^{31} , where prescalar = 6.
b. Polynomial coefficients (Cn_p*-*, offset is 0x0 to 0x8)
   Data formats: Q8.24 (2's complement). Range: -128 to 128-2^{-24} .
c. Gain value's data formats: Q16.16 (2's complement). Range: -32768 to 32768-2^{-16} .

7.7.3.12.4 Reading Peak Values

The peak value statuses are available in a RAM as shown below.

Figure 7.48 Peak Values



To read a value from this RAM, one must use the registers AHUBRAMCTL_PEAKM_CTRL and AHUBRAMCTL_PEAKM_DATA. Refer to the "Programming Module RAMs" subsection for programming guidelines on writing/reading RAM locations.

7.7.3.13 AMX Programming Guidelines

7.7.3.13.1 Mandatory Guidelines

1. Ensure that the AMX is in a disabled state (after being used in a previous use case)
   Check (AMX_STATUS_ENABLE_STATUS == 0).
2. Configure the AMX.
   a. Configure the receive and transmit ACIF parameters (see the "ACIF Programming Guidelines" section for more details
      Populate AMX_XBAR_RX_CIF_CTRL0, AMX_XBAR_RX_CIF_CTRL1,
      AMX_XBAR_RX_CIF_CTRL2,
      AMX_XBAR_RX_CIF_CTRL3 and AMX_XBAR_TX_CIF_CTRL registers.
   b. Enable the inputs that are used for making the output frame
      Use register fields AMX_CTRL_RX<i>_EN, where i = 1, 2, 3, or 4.
   c. Program the byte RAM map
      Use AMX_AUDIORAMCTL_AMX_CTRL and AMX_AUDIORAMCTL_AMX_DATA registers.
      Refer to the "Programming Module RAMs" subsection for programming guidelines on reading RAM locations.
   d. Program the "wait-on" conditions for AMX
      Configure the AMX_CTRL_RX_DEP, AMX_CTRL_MSTR_RX_NUM register fields
   e. Configure which of the bytes in the output stream contain data
      Configure the AMX_OUT_BYTE_EN0 register for the LSB 32 bytes and the AMX_OUT_BYTE_EN1 register for the MSB 32 bytes.
3. Enable the AMX module.
   Set (AMX_ENABLE = 1).
4. Ensure the module is done initializing before sending data to it.
   Ensure (AMX_STATUS_ENABLE_STATUS == 1).
5. After using the module, disable the module.
   Set (AMX_ENABLE = 0).

7.7.3.13.2 Optional Guidelines

1. Configure AMX to stop accepting new input frames right after an input is disabled (otherwise AMX waits until all the continuous incoming frames are received and then disables)
   Set field(s) (AMX_CTRL.RX<i>_FORCE_DISABLE = ENABLE), where i = 1, 2, 3, or 4.
2. For auto disable and auto enable feature, program the IDLE count first in RX<i>_CTRL_FRAME_PERIOD_0_IDLE_CNT register, where i = 1, 2, 3, or 4, then program AMX_CYA_0 [0]. Both need be performed before enabling AMX.
   Note: Refer to the "Common Programming Guidelines" section for information on using some optional features in the AMX (and all other modules).
3. Configure Audio CIF control registers AMX_AUDIOCIF_CH0_CTRL, AMX_AUDIOCIF_CH1_CTRL, AMX_AUDIOCIF_CH2_CTRL, AMX_AUDIOCIF_CH3_CTRL, and

AMX_AUDIOCIF_OUT_CTRL according to the guidelines given in the "ACIF Programming Guidelines" subsection.

4. Configure the CH_DEP parameter field in the AMX_CTRL_0 register to determine whether to wait for all enabled channels to have data before transfer or start sending the data when any one of the input channels has data (WT_ON_ALL and WT_ON_ANY).
5. Configure the MSTR_CH_NUM register for designated master channel, so that if the data is available on a particular channel, the output can be sent.
6. Enable the AMX output by setting BYTE_EN in AMX_OUT_BYTE_EN0 and AMX_OUT_BYTE_EN1.
7. For Byte RAMCTL programming:
   - For programming RAM with the HW_ADR_EN bit cleared in the AMX_AUDIORAMCTL_AMX_CTRL register, set the RAM Offset in the RAM_ADR field of the AMX_AUDIORAMCTL_AMX_CTRL register and set data in the DATA field in the AMX_AUDIORAMCTL_AMX_DATA register.
   - For programming RAM with the HW_ADR_EN bit set in the AMX_AUDIORAMCTL_AMX_CTRL register, set the RAM Offset in the RAM_ADR field of the AMX_AUDIORAMCTL_AMX_CTRL register to 0 and set data in the DATA field in the AMX_AUDIORAMCTL_AMX_DATA register. Hardware auto-increments the RAM_ADR after each write.

Write sequence for single location write (NOT in burst/sequential)

1. Set address in AUDIORAMCTL_AMX_CTRL_0; make sure that (SEQ_ACCESS_EN == 0) for single write/read (NOT in burst).
2. Set RW field in AUDIORAMCTL_AMX_CTRL_0
3. Set ADDR_INIT_EN field - optional for write
4. Program (AUDIORAMCTL_AMX_DATA_0 = data to write).

Read sequence for single location read (NOT in burst/sequential):

1. Set address in AUDIORAMCTL_AMX_CTRL_0; make sure that (SEQ_ACCESS_EN == 0) for single write/read (NOT in burst).
2. Reset RW field in AUDIORAMCTL_AMX_CTRL_0
3. Set ADDR_INIT_EN field
4. Read the data from AUDIORAMCTL_AMX_DATA_0

7.7.3.14 ADX Programming Guidelines

7.7.3.14.1 Mandatory Guidelines

1. Ensure that the ADX is in a disabled state (after being used in a previous use case): Check if (ADX_STATUS.ENABLE_STATUS == 0).
2. Configure the ADX:

a. Configure the receive and transmit ACIF parameters; see "ACIF Programming Guidelines" for more details.
Populate ADX_XBAR_TX_CIF_CTRL0, ADX_XBAR_TX_CIF_CTRL1, ADX_XBAR_TX_CIF_CTRL2, ADX_XBAR_TX_CIF_CTRL3 and ADX_XBAR_RX_CIF_CTRL registers.
b. Enable the inputs that are used for making the output frame:
Use register fields ADX_CTRL_TX<j>_EN, where j = 1, 2, 3, 4, or 5.
c. Program the byte RAM map:
Use ADX_AUDIORAMCTL_ADX_CTRL and ADX_AUDIORAMCTL_ADX_DATA registers. Refer to the "Programming Module RAMs" subsection for programming guidelines on reading RAM locations.
d. Configure which of the bytes in the input stream contain data:
Configure the ADX_IN_BYTE_EN0 register for the LSB 32 bytes and the ADX_IN_BYTE_EN1 register for the MSB 32 bytes.

3. Enable the ADX module:
Set (ADX_ENABLE = 1).
4. Ensure the module is done initializing.
Poll (ADX_STATUS.ENABLE _STATUS == 1).
5. After using the module, disable the module:
Set (ADX_ENABLE = 0).

7.7.3.14.2 Optional Guidelines

- Configure ADX to stop generating new input frames right after an output is disabled (otherwise ADX waits until all the continuous incoming frames are received and then disables)
Set field(s) (ADX_CTRL_0.TX<j>_FORCE_DISABLE = ENABLE), where j = 1, 2, 3, or 4.

7.7.3.15 SFC Programming Guidelines

1. Ensure that the SFC is in a disabled state (after being used in a previous use case):
Check if (SFC_STATUS.ENABLE_STATUS == 0).
2. Configure the SFC:
a. Configure the receive and transmit ACIF parameters (see "ACIF Programming Guidelines" for more details)
Populate SFC_AXBAR_RX_CIF_CTRL and SFC_AXBAR_TX_CIF_CTRL registers.
b. Configure input/output sampling frequencies
Populate SFC_AXBAR_RX_FREQ_FS_IN and SFC_AXBAR_TX_FREQ_FS_OUT registers.
3. There are hardcoded coefficients for commonly used frequency conversions (see below). But for other frequency conversions, the software *must* program filter coefficients. Furthermore, optionally, even for the common use cases, the software may choose to program the filter coefficients. Note that the coefficients are in Q23 number format (1 sign bit and 23 fractional bits).
a. Set (SFC_COEF_RAM_0.COEF_RAM_EN = 1).

   b. Program the RAM via SFC_AHUBRAMCTL_SFC_CTRL_0 and SFC_AHUBRAMCTL_SFC_DATA_0. Refer to the "Programming Module RAMs" subsection for how to program any RAM in an AHUB module.
4. Enable the SFC module:
   Set SFC_ENABLE to 1.
5. Ensure the module initialization is done before sending data to SFC:
   Ensure (SFC_STATUS.ENABLE_STATUS == 1).
6. After using the module, disable the module:
   Set (SFC_ENABLE = 0).

7.7.3.15.1 Commonly Used Frequency Conversions

- 8 to 44.1 kHz
- 8 to 48 kHz
- 16 to 44.1 kHz
- 16 to 48 kHz
- 44.1 to 8 kHz
- 44.1 to 16 kHz
- 48 to 8 kHz
- 48 to 16 kHz

7.7.3.16 AFC Programming Guidelines

1. Ensure that the AFC is in a disabled state (after being used in a previous use case):
   Check if (AFC_STATUS_ENABLE_STATUS == 0).
2. Configure the AFC:
   a. Configure the receive and transmit ACIF parameters (see "ACIF Programming Guidelines" for more details)
      Populate AFC_XBAR_RX_CIF_CTRL and AFC_XBAR_TX_CIF_CTRL registers.
   b. Configure the maximum expected PPM difference between the two clock sources that AFC is used to control the flow between
      Configure the AFC_CLK_PPM_DIff register.
   c. Configure the I^2S FIFO thresholds and AFC TxCIF thresholds that are used by the AFC logic
      Configure the AFC_CLK_PPM_DIff register*.
3. Enable the AFC module:
   Set (AFC_ENABLE = 1).
4. Ensure the module initialization is complete before sending data to AFC:
   Ensure (AFC_STATUS_ENABLE_STATUS == 1).
5. After using the module, disable the module:
   Set (AFC_ENABLE = 0).

The following table shows the guidelines to follow for setting the threshold values under various conditions.

In the table below, the numbers in parentheses are example values.

Table 7.92 Threshold Settings

Use Case	AFC_THRESHOLDS_I2S_0			AFC_THRESHOLDS_AFC_0			Destination I2S FIFO Threshold (Programmed in I2S_AUDIOCIF_I2SRX_CTRL_0.FIFO_THRESHOLD)
Actual (Typ.) Value	HIGH_THRESHOLD	START_THRESHOLD	LOW_THRESHOLD	HIGH_THRESHOLD	START_THRESHOLD	LOW_THRESHOLD	
I2S → AFC → I2S	(START_THRESHOLD + 1) (8)	START_THRESHOLD (7)	(START_THRESHOLD – 1) (6)	(START_THRESHOLD + 1) (8)	START_THRESHOLD (7)	(START_THRESHOLD – 1) (6)	(START_THRESHOLD + 1) AFC_THRESHOLDS_I2S_0 (8)
I2S → SFC → AFC → I2S	(2 * SRC-BURST + 1)	(1 * SRC-BURST + 2)	(1 * SRC-BURST)	(1 * SRC-BURST + 1)	(1 * SRC-BURST + 1)	(1 * SRC-BURST + 1)	(START_THRESHOLD + 1) AFC_THRESHOLDS_I2S_0
I2S → AFC → AMX → I2S	(START_THRESHOLD + 1) (8)	START_THRESHOLD (7)	(START_THRESHOLD – 1) (6)	(START_THRESHOLD + 1) (8)	START_THRESHOLD (7)	(START_THRESHOLD – 1) (6)	(START_THRESHOLD + 1) AFC_THRESHOLDS_I2S_0 (8)
I2S → SFC → AFC → AMX → I2S	(2 * SRC-BURST +1)	(1 * SRC-BURST + 2)	(1 * SRC-BURST)	(2 * SRC-BURST + 4)	(1 * SRC-BURST + 4)	(4)	(START_THRESHOLD + 1) AFC_THRESHOLDS_I2S_0

7.7.3.17 OPE Programming Guidelines

1. Ensure that OPE is in a disabled state (after being used in a previous use case) Check if (OPE_STATUS.ENABLE_STATUS == 0).
2. Configure OPE:

   a. Configure the receive and transmit ACIF parameters (see "ACIF Programming Guidelines" for more details)
      Populate OPE_XBAR_RX_CIF_CTRL and OPE_XBAR_TX_CIF_CTRL registers.
   b. Configure MBDRC and PEQ
3. Enable OPE module:
   Set (OPE_ENABLE = 1).
4. Ensure the module initialization is complete before sending data to MBDRC:
   Ensure (OPE_STATUS.ENABLE_STATUS == 1).
5. After using the module, disable the module:
   Set (OPE_ENABLE = 0).

If an output device is changing, or the coefficients and other settings need to be changed:

1. Disable OPE.
2. Wait for (OPE_STATUS_ENABLE_STATUS == 0) for datapath flushed out, or OPE_SOFT_RESET if immediate disable is needed.
3. Reprogram all coefficients and configuration registers as required.
4. Trigger OPE_SOFT_RESET.
5. Enable OPE as before.

Note: Coefficient Data programmed takes care of the signs, i.e., hardware will always do add, so to implement a BiQuad operation:

> b0 * x[n] + b1 * x[n-1] + b2 * x[n-2] + (-a1) * y[n-1] + (-a2) * y[n-2]. Software needs to provide b0, b1, b2, -a1, -a2 as the coefficients.

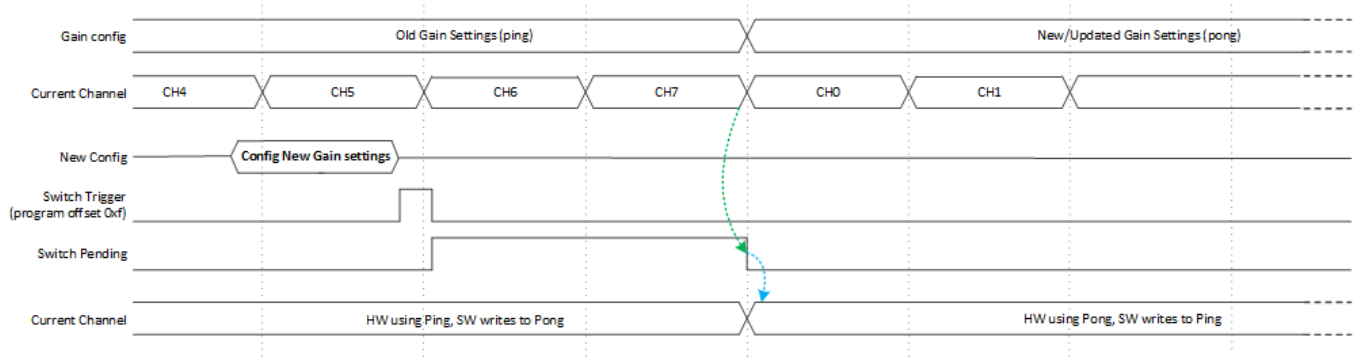
7.7.3.18 MVC Programming Guidelines

1. Ensure that the MVC is in a disabled state (after being used in a previous use case):
   Check if (MVC_STATUS == 0).
2. Configure the MVC:
   a. Configure the receive and transmit ACIF parameters (see "ACIF Programming Guidelines" for more details).
   b. Program MVC_CONFIG registers.
   c. Program all Volume related settings (Target Volume, Duration, Polynomial Coefficients in Polynomial mode).
3. Enable MVC module:
   Set (MVC_ENABLE = 1).
4. Ensure the module initialization is complete before sending data to MVC: Ensure MVC_STATUS is 1.
5. After using the module, disable the module:
   Set (MVC_ENABLE = 0).

If software wants to change polynomial coefficients, duration parameters, or volume parameters (target volume and mute_unmute) on the fly while MVC is in enable state:

1. Check the status of switching: COEFF_SWITCH, DURATION_SWITCH and VOLUME_SWITCH.
2. Program new parameters (For coefficient switching, hardware handles whether that goes to Ping or Pong).
3. Set COEFF_SWICTH/DURATION_SWITCH/VOLUME_SWITCH to let hardware know new parameters are ready.

Figure 7.49 Programming Timeline for Updated Software Configuration



Note: If software wants to program polynomial coefficients, duration parameters and volume parameters at the same time, then software should first program all the required fields for polynomial coefficients, duration parameters, and volume parameters, then set COEFF_SWITCH, DURATION_SWITCH, and VOLUME SWITCH together (they all belong to the same register).

If an output device is changing, or the settings need to be changed:

1. Disable the MVC.
2. Reprogram all registers.
3. Trigger Soft Resets for MVC and CIFs.
4. Enable – in same order above.

If the software is changing parameters while MVC is in disable state, following points should be kept in consideration:

1. If software wants to retain history between disable and enable stages, then when software disables the MVC, it should also set the INIT_VOL register to the last set Volume value by software. Examples:
   a. If software sets the volume to T1 and then disables the module, INIT_VOL should be set to T1.
   b. If software sets the volume to T1, then press Mute button and then disables the module, INIT_VOL should be set to T1.
   c. If software sets the volume to T1 and then disables the module, INIT_VOL should be set to T1. Now if during the disable stage, if software changes the volume to T2, INIT_VOL should

be set to T2
Exception: The only exception to the above case is if the last set volume was 0. If the last set volume was 0, then the INIT_VOL register should be set to the INIT_VOL during boot-up.

2. If software unmutes a channel while in the disabled stage, it needs to update the TARGET_VOL register to the last set volume value by software.

7.7.3.19 DSPK Programming Guidelines

DSPK Controller Programming

1. Ensure that DSPK is in disabled state (after being used in previous use case).
   Check if (DSPK_STATUS.RX_ENABLED == 0).
2. Apply soft-reset to DSPK to flush out all the FIFOs.
   (DSPK_SOFT_RESET.SOFT_RESET = 1).
3. Program CIF CTRL register.
   Configure DSPK_RX_CIF_CTRL register. Only mono and stereo channels are supported.
4. Program DSPK_CORE CTRL register.
   Configure STAGE1_GAIN, STAGE2_GAIN, STAGE3_GAIN, LOWDELAY_FILTER_MODE, CHANNEL_SELECT, OSR and LRSEL_POLARITY bits.
5. Program SDM COEF registers.
   Based on the value of OSR, program these registers appropriately. Specific values are mentioned in next column.
6. Enable the Module.
   Program (DSPK_ENABLE = 1).
7. Disable the Module.
   After using, disable dspk. (DSPK_ENABLE = 0).
8. Wait for module to get disabled.
   Poll for (RX_DONE == 1); wait until it is asserted.

CODEC Programming

1. Program DSPK_CODEC_CTRL register.
   Configure CHANNEL_SELECT for either mono/stereo channels.
   Configure BIT_ORDER to specify LSB/MSB bit to be sent first.
   Configure (CODEC_CONFIG_MODE = 1).
   Configure CNFG_REP_NUM for repetition number.
2. Program DSPK_CODEC_DATA register.
   Configure CH1_CONTROL_WORD and CH0_CONTROL_WORD.
3. Enable dspk module.
   Program (DSPK_ENABLE = 1).
4. Enable CODEC_ENABLE bit.
   Program (CONFIG_START = 1).
5. Wait until codec programming is done.
   Poll for CONFIG_DONE bit.

6. Disable CODEC_CONFIG mode.

DSM Coefficients Programming for Different OSR's

Table 7.93 DSM Coefficients for Different OSR's

	DSMCOEFS_5_256	DSMCOEFS_5_128	DSMCOEFS_5_64	DSMCOEFS_5_32
a = [a2,a3,a4,a5]	[5208,7652,11087,15885]	[5268,7819,11157,17072]	[5051,7359,10711,14210]	[4935,7489,10794,16429]
c= [c1,c2,c3,c4]	[168,6622,11092,18392]	[171,6675,10927,19618]	[166,6489,11167,16920]	[175,6494,11136,19015]
g = [g1,g2]	[7,7]	[28,27]	[116,125]	[462,446]

7.7.3.20 ASRC Programming Guidelines

For enabling a stream, Global and Stream specific registers need to be configured as mentioned below. Global registers are to be configured only once after a "hard reset" and not required for every time a new stream is enabled or disabled.

7.7.3.20.1 Configure ASRC's Global Registers

- Configure Intermediate buffer location using ASRC_GLOBAL_SCRATCH_ADDR register
  - Address needs to be aligned to at least 64 bits.
  - Size of the Buffer required is calculated using 12288 + 1792 * NUM_CHANNELS.
  - For all 12 channels, Size = 12288 + 1792 * 12 = 33792 Bytes.
- Set ASRC_GLOBAL_ENB to "1"
  - Check if (ASRC_GLOBAL_STATUS.GLOBAL_ENABLED == 1).

7.7.3.20.2 Configure ASRC's Stream Specific Registers

- Ensure that particular stream is in not active by checking (STREAM<i>_STATUS.ENABLE_STATUS == 0).
- Configure STREAM<i>_CONFIG.RATIO_TYPE based on whether stream receives on-the-fly ratio updates from software or from ARAD hardware module.
- Configure Input and Output CIF using STREAM<i>_RX_CIF_CTRL and STREAM<i>_TX_CIF_CTRL registers.
  - AXBAR_CHANNELS and AXBAR_BITS fields need to be configured for the receive ACIF.
  - AXBAR_BITS field needs to be configured for the transmit ACIF.
- Enable the stream using STREAM<i>_ENB register.
- Ensure (STREAM<i>_STATUS.ENABLE_STATUS == 1).
- After the processing is done, disable the stream using STREAM<i>_ENB register.
  - Ensure (STREAM<i>_STATUS_ENABLE_STATUS == 0).

Note: i = 1, 2, 3, 4, 5, or 6 for the stream number.

7.7.3.20.3 ASRC Ratio

For doing the conversion, ASRC expects a scalar value which is the ratio of Input Sampling Frequency (Fs_in) to Output Sampling Frequency (Fs_out) expressed in Q32 format. Thus:

Ratio = Fs_in / Fs_out

Example:

If Fs_in = 48 kHz, Fs_out = 44.1 kHz

Ratio = 48000 ÷ 44100 = 1.088435374149659863945578231 2925

Integer portion of the ratio is, RATIO_INT = "1"

Fractional portion of the ratio is, Frac_real = "0.088435374149659863945578231 2925".

In Q32 format, Fractional portion, RATIO_FRAC = Round (Frac_real * 2^{32}) = 379827040 rounds the result to nearest integer.

Note: there is a precision loss while forming the ratio value itself.

Some more examples are listed in the table below.

Table 7.94 ASRC Ratio Examples

Fsin (in kHz)	Fsout (in kHz)	Ratio (in real)	RATIO_INT	RATIO_FRAC
8.0	8.0	1.000000000	1	0
8.0	16.0	0.500000000	0	2147483648
16.0	44.1	0.362811791	0	1558264779
48.0	44.1	1.088435374	1	379827040
44.1	8.0	5.512500000	5	2201170739
8.0	192.0	0.041666667	0	178956971
192.0	96.0	2.000000000	2	0

7.7.3.20.4 On-the-Fly Ratio Updates from Software

- Configure STREAM<i>_CONFIG_RATIO_TYPE to FROM_SW mode.
- Configure (ASRC_STREAM<i>.RATIO_INTEGER_PART = 5-bit integer portion of RATIO_INT); see description in the previous section.

- Configure (ASRC_STREAM<i>.RATIO_FRAC_PART = 32-bit fractional portion of RATIO_FRAC); see description in the previous section.
- Configure (ASRC_STREAM<i>.RATIO_LOCK_STATUS = LOCK) for the new ratio to be used by hardware.

Note: i = 1, 2, 3, 4, 5, or 6 for the stream number.

7.7.3.20.5 On-the-Fly Ratio Updates from ARAD

- Configure (STREAM<i>_CONFIG.RATIO_TYPE) = FROM_ARAD).
- Configure (STREAM<i>_CONFIG.LANE_ID = same lane ID as used by ARAD for the corresponding ratio update packets.
- Connect ASRC "Ratio Update CIF" to ARAD's TxCIF to receive the ratio updates from ARAD.
  - Set (AXBAR_PART_1_ASRC1_RX7_0.ARAD1_TX1 = 1).

Note: i = 1, 2, 3, 4, 5, or 6 for the stream number.

7.7.3.20.6 ASRC Hardware Ratio Compensation

It is recommended to enable Hardware Ratio Compensation for all I/O to I/O use cases irrespective of ratios:

- Configure ENABLE_HW_RATIO_COMP in the ASRC_STREAM<i>_CONFIG register.
- Configure NORMAL_WMARK and UPPER_WMARK for the input buffer in the ASRC_STREAM1_RX_THRESHOLD register.
  - Default values of 6 (upper) and 4 (normal) cause too much ratio compensation to happen.
  - Experiments show 6 and 3 are suitable.
- Configure NORMAL_WMARK and LOWER_WMARK for the output buffer in the ASRC_STREAM1_TX_THRESHOLD register.
  - Default values of 2 (normal) and 1 (upper) cause too much ratio compensation to happen.
  - Experiments show 3 (normal) and 0 (lower) seems to be more suitable.
- Configure COMP_VALUE in the ASRC_STREAM1_RATIO_COMP register.
  - Initially COMP_VALUE = 0xffff helped covering all I/O-I/O scenarios
  - COMP_VALUE = 0x3ffff seemed to help bring the buffer levels back to normal levels without applying any software work-around. This was tested for 48 -> 48 and 48 -> 8 conversion ratios. Need more conversion ratios to be tested to finally conclude a value.

Note: i = 1, 2, 3, 4, 5, or 6 for the stream number.

7.7.3.20.7 Setting ASRC Input and Output Start Buffer Threshold:

- Configure WORD_CNT in the ASRC_STREAM1_RX_THRESHOLD register for input.
- Configure WORD_CNT in the ASRC_STREAM1_TX_THRESHOLD register for output.

By default, both are set to "2" (which means 64 samples), which is sufficient. In the worst case, this corresponds to a total buffering delay of ~8 mS (= 64 / 8 kHz). In case software needs to reduce the buffering delay, it can try configuring these fields to even lesser values (1 or 0).

7.7.3.20.8 Configure ASRC CIF's APR Settings

ASRC follows other AHUB modules with respect to input and output samples security. Hence if the input that comes over the ACIF is APR enabled, output is also APR enabled.

To disable APR_EN for output:

- Software needs to configure ASRC_GLOBAL_DISARM_APR_0 register.
- Disable access to the DISARM_APR register by setting ASRC_GLOBAL_DISARM_APR_ACCESS_CTRL register to "1". This register can be modified only via SECURE register access.

7.7.3.20.9 To Secure ASRC Intermediate Data Buffer

- For securing the intermediate data buffer:
  - Configure (GLOBAL_APR_CTRL.APR_EN = 1).
  - Configure (GLOBAL_APR_CTRL.APR_REGION_ID = APR region parameters).
  - Set (GLOBAL_APR_CTRL.ACCESS_CTRL = 1) to disable access to the APR_CTRL register. This register can be modified only via SECURE register access.
- Once the above three registers are configured, GLOBAL_SCRATCH_ADDR should be configured with an address belonging to APR region to ensure proper operation.

7.7.3.21 ARAD Programming Guidelines

1. Check if the lane is in disabled state before using it:
   If (LANE_STATUS.LANE<i>_ENABLED == FALSE).
2. Choose the Numerator and Denominator Clocks for the lane:
   Configure LANE<i>_NUMERATOR_MUX_SEL and LANE<i>_DENOMINATOR_MUX_SEL.
3. Configure Numerator and Denominator Prescalars if required:
   Configure LANE<i>_NUMERATOR_PRESCALAR and/or LANE<i>_DENOMINATOR_PRESCALAR.
4. Enable the lane for use:
   Set (LANE_ENABLE.LANE<i>_ENABLE = 1).
5. After using a lane, disable it:
   Set (LANE_ENABLE.LANE<i>_ENABLE = 0).

Note: i = 1, 2, 3, 4, 5, or 6 for the lane number.

7.7.3.22 ADMA Programming Guidelines

7.7.3.22.1 ADMA Transfers

1. Software should not attempt to write into registers of an active channel after the TRANSFER_ENABLE bit is set to "1." Accesses to the following registers are exceptions to this constraint:
   - TRANSFER_PAUSE
   - SOFT_RESET
2. Soft reset should not be applied when DMA transfers are in progress. It can only be applied when channel is hung or in PAUSE state.
3. For disabling flow controlled DMA transfers (AHUB-to-Memory, AHUB-to-AHUB, and Memory-to-AHUB) from an ADMA channel, software should always disable it prior to disabling the corresponding ADMAIF channel.
4. Software should wait until the TRANSFER_ENABLED status bit goes low and clear any pending Interrupts for re-enabling a disabled channel.
5. TRANSFER_COUNT should be set to a non-zero value
6. Software should ensure AHUB_FIFO_CTRL_0_TX_FIFO_SIZE or AHUB_FIFO_CTRL_0_RX_FIFO_SIZE registers in an ADMA channel and FIFO_CTRL_0_DMA_FIFO_SIZE register of AHUB channel mapped to this ADMA channel are programmed to the same value. ADMA and AHUB channel mapping is done through CTRL_0_RX_REQUEST_SELECT or CTRL_0_TX_REQUEST_SELECT registers in ADMA.
7. TRANSFER_COUNT status may not reflect the true value while transfer is in progress, so the channel should be PAUSED for getting the correct status.
8. Software may choose to terminate a transfer or disable a channel abruptly by clearing the TRANSFER_ENABLE bit or wait until hardware disables it automatically up on normal completion of transfer.
9. Software should wait until the TRANSFER_ENABLED status bit goes low before changing any of the ADMA channel control and configuration registers.

7.7.3.22.2 Configuring ADMA Pages for Virtualization

ADMA page configuration should be done by the host OS, which has access to the ADMA_GLOBAL registers. Each guest OS only has access to the corresponding ADMA channel page. This restriction is done outside of APE.

1. For each ADMA OS page N:
   - Assign ADMA channel groups to ADMA pages in ADMA_GLOBAL_PAGE[N]_CHGRP. Each channel group has four ADMA channels.
   - Enable the ADMAIF_TX channel that the page can access in ADMA_GLOBAL_PAGE[N]_TX_REQUESTORS.
   - Enable the ADMAIF_RX channel that the page can access in ADMA_GLOBAL_PAGE[N]_RX_REQUESTORS.

- Program whether the page can access ARAM (ADMA_GLOBAL_PAGE[N]_ARAM_CFG). It is recommended Guest OS does not have access to ARAM.

2. For each ADMA channel N:
   - Assign REGION_ID in ADMA_GLOBAL_CH[N]_REGION_ID. For channel that belongs to the same page should be assigned with the same REGION_ID.

7.7.4 APE Registers

Refer to "Reading Register Tables" in the Introduction chapter for the register table protocol as well as recommendations for accessing registers.

The register descriptions below give the offset of each register within the APE address range. Note that there are two instances of APE_AST, namely APE_ACAST_0 and APE_ADAST, therefore two sets of ape_ast and ape_ast_scr registers, which differ only in their base addresses. The Base Addresses of the registers in the APE and its internal modules are specified in the Address Map section of the Orin TRM.

> Audio Address Space Translator (AST) Registers:
>
> There are two instances of the Audio Address Space Translator (AST) Registers, by the name of APE_ACAST and APE_ADAST. For details of the registers descriptions and base addresses, refer to the Address Space Translator (AST) Registers in the Address Space Translation (AST) chapter.

7.7.4.1 Audio Crossbar (AXBAR) Registers

AXBAR_PART_<*i*>_ADMAIF_RX<*j*>_0,

where <*i*> = 0, 1, 2, 3 and

<*j*> = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 1, 19, 20.

AXBAR_PART_0_ADMAIF_RX1_0
Offset: 0x00000

AXBAR_PART_0_ADMAIF_RX2_0
Offset: 0x00004

AXBAR_PART_0_ADMAIF_RX3_0
Offset: 0x00008

AXBAR_PART_0_ADMAIF_RX4_0
Offset: 0x0000c

AXBAR_PART_0_ADMAIF_RX5_0
Offset: 0x00010

AXBAR_PART_0_ADMAIF_RX6_0
Offset: 0x00014

AXBAR_PART_0_ADMAIF_RX7_0
Offset: 0x00018

AXBAR_PART_0_ADMAIF_RX8_0
Offset: 0x0001c

AXBAR_PART_0_ADMAIF_RX9_0
Offset: 0x00020

AXBAR_PART_0_ADMAIF_RX10_0
Offset: 0x00024

AXBAR_PART_0_ADMAIF_RX11_0
Offset: 0x00028

AXBAR_PART_0_ADMAIF_RX12_0
Offset: 0x0002c

AXBAR_PART_0_ADMAIF_RX13_0
Offset: 0x00030

AXBAR_PART_0_ADMAIF_RX14_0
Offset: 0x00034

AXBAR_PART_0_ADMAIF_RX15_0
Offset: 0x00038

AXBAR_PART_0_ADMAIF_RX16_0
Offset: 0x0003c

AXBAR_PART_0_ADMAIF_RX17_0
Offset: 0x001a0

AXBAR_PART_0_ADMAIF_RX18_0
Offset: 0x001a4

AXBAR_PART_0_ADMAIF_RX19_0
Offset: 0x001a8

AXBAR_PART_0_ADMAIF_RX20_0
Offset: 0x001ac

AXBAR_PART_1_ADMAIF_RX1_0
Offset: 0x00200

AXBAR_PART_1_ADMAIF_RX2_0
Offset: 0x00204

AXBAR_PART_1_ADMAIF_RX3_0
Offset: 0x00208

AXBAR_PART_1_ADMAIF_RX4_0
Offset: 0x0020c

AXBAR_PART_1_ADMAIF_RX5_0
Offset: 0x00210

AXBAR_PART_1_ADMAIF_RX6_0
Offset: 0x00214

AXBAR_PART_1_ADMAIF_RX7_0
Offset: 0x00218

AXBAR_PART_1_ADMAIF_RX8_0
Offset: 0x0021c

AXBAR_PART_1_ADMAIF_RX9_0
Offset: 0x00220

AXBAR_PART_1_ADMAIF_RX10_0
Offset: 0x00224

AXBAR_PART_1_ADMAIF_RX11_0
Offset: 0x00228

AXBAR_PART_1_ADMAIF_RX12_0
Offset: 0x0022c

AXBAR_PART_1_ADMAIF_RX13_0
Offset: 0x00230

AXBAR_PART_1_ADMAIF_RX14_0
Offset: 0x00234

AXBAR_PART_1_ADMAIF_RX15_0
Offset: 0x00238

AXBAR_PART_1_ADMAIF_RX16_0
Offset: 0x0023c

AXBAR_PART_1_ADMAIF_RX17_0
Offset: 0x003a0

AXBAR_PART_1_ADMAIF_RX18_0
Offset: 0x003a4

AXBAR_PART_1_ADMAIF_RX19_0
Offset: 0x003a8

AXBAR_PART_1_ADMAIF_RX20_0
Offset: 0x003ac

AXBAR_PART_2_ADMAIF_RX1_0
Offset: 0x00400

AXBAR_PART_2_ADMAIF_RX2_0
Offset: 0x00404

AXBAR_PART_2_ADMAIF_RX3_0
Offset: 0x00408

AXBAR_PART_2_ADMAIF_RX4_0
Offset: 0x0040c

AXBAR_PART_2_ADMAIF_RX5_0
Offset: 0x00410

AXBAR_PART_2_ADMAIF_RX6_0
Offset: 0x00414

AXBAR_PART_2_ADMAIF_RX7_0
Offset: 0x00418

AXBAR_PART_2_ADMAIF_RX8_0
Offset: 0x0041c

AXBAR_PART_2_ADMAIF_RX9_0
Offset: 0x00420

AXBAR_PART_2_ADMAIF_RX10_0
Offset: 0x00424

AXBAR_PART_2_ADMAIF_RX11_0
Offset: 0x00428

AXBAR_PART_2_ADMAIF_RX12_0
Offset: 0x0042c

AXBAR_PART_2_ADMAIF_RX13_0
Offset: 0x00430

AXBAR_PART_2_ADMAIF_RX14_0
Offset: 0x00434

AXBAR_PART_2_ADMAIF_RX15_0
Offset: 0x00438

AXBAR_PART_2_ADMAIF_RX16_0
Offset: 0x0043c

AXBAR_PART_2_ADMAIF_RX17_0
Offset: 0x005a0

AXBAR_PART_2_ADMAIF_RX18_0
Offset: 0x005a4

AXBAR_PART_2_ADMAIF_RX19_0
Offset: 0x005a8

AXBAR_PART_2_ADMAIF_RX20_0
Offset: 0x005ac

AXBAR_PART_3_ADMAIF_RX1_0
Offset: 0x00000

AXBAR_PART_3_ADMAIF_RX2_0
Offset: 0x00004

AXBAR_PART_3_ADMAIF_RX3_0
Offset: 0x00008

AXBAR_PART_3_ADMAIF_RX4_0
Offset: 0x0000c

AXBAR_PART_3_ADMAIF_RX5_0
Offset: 0x00010

AXBAR_PART_3_ADMAIF_RX6_0
Offset: 0x00014

AXBAR_PART_3_ADMAIF_RX7_0
Offset: 0x00018

AXBAR_PART_3_ADMAIF_RX8_0
Offset: 0x0001c

AXBAR_PART_3_ADMAIF_RX9_0
Offset: 0x00020

AXBAR_PART_3_ADMAIF_RX10_0
Offset: 0x00024

AXBAR_PART_3_ADMAIF_RX11_0
Offset: 0x00028

AXBAR_PART_3_ADMAIF_RX12_0
Offset: 0x0002c

AXBAR_PART_3_ADMAIF_RX13_0
Offset: 0x00030

AXBAR_PART_3_ADMAIF_RX14_0
Offset: 0x00034

AXBAR_PART_3_ADMAIF_RX15_0
Offset: 0x00038

AXBAR_PART_3_ADMAIF_RX16_0
Offset: 0x0003c

AXBAR_PART_3_ADMAIF_RX17_0
Offset: 0x007a0

AXBAR_PART_3_ADMAIF_RX18_0
Offset: 0x007a4

AXBAR_PART_3_ADMAIF_RX19_0
Offset: 0x007a8

AXBAR_PART_3_ADMAIF_RX20_0
Offset: 0x007ac

Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,0000,xx00,0000,0000,0000,0000,0000)

Bit	Reset	Description
27	0x0	**SFC4_TX1**: **0** = DISABLE **1** = ENABLE
26	0x0	**SFC3_TX1**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
25	0x0	**SFC2_TX1**: **0** = DISABLE **1** = ENABLE
24	0x0	**SFC1_TX1**: **0** = DISABLE **1** = ENABLE
21	0x0	**I2S6_TX1**: **0** = DISABLE **1** = ENABLE
20	0x0	**I2S5_TX1**: **0** = DISABLE **1** = ENABLE
19	0x0	**I2S4_TX1**: **0** = DISABLE **1** = ENABLE
18	0x0	**I2S3_TX1**: **0** = DISABLE **1** = ENABLE
17	0x0	**I2S2_TX1**: **0** = DISABLE **1** = ENABLE
16	0x0	**I2S1_TX1**: **0** = DISABLE **1** = ENABLE
15	0x0	**ADMAIF_TX16**: **0** = DISABLE **1** = ENABLE
14	0x0	**ADMAIF_TX15**: **0** = DISABLE **1** = ENABLE
13	0x0	**ADMAIF_TX14**: **0** = DISABLE **1** = ENABLE
12	0x0	**ADMAIF_TX13**: **0** = DISABLE **1** = ENABLE
11	0x0	**ADMAIF_TX12**: **0** = DISABLE **1** = ENABLE
10	0x0	**ADMAIF_TX11**: **0** = DISABLE **1** = ENABLE
9	0x0	**ADMAIF_TX10**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
8	0x0	**ADMAIF_TX9**: **0** = DISABLE **1** = ENABLE
7	0x0	**ADMAIF_TX8**: **0** = DISABLE **1** = ENABLE
6	0x0	**ADMAIF_TX7**: **0** = DISABLE **1** = ENABLE
5	0x0	**ADMAIF_TX6**: **0** = DISABLE **1** = ENABLE
4	0x0	**ADMAIF_TX5**: **0** = DISABLE **1** = ENABLE
3	0x0	**ADMAIF_TX4**: **0** = DISABLE **1** = ENABLE
2	0x0	**ADMAIF_TX3**: **0** = DISABLE **1** = ENABLE
1	0x0	**ADMAIF_TX2**: **0** = DISABLE **1** = ENABLE
0	0x0	**ADMAIF_TX1**: **0** = DISABLE **1** = ENABLE

AXBAR_PART_<*i*>_I2S<*k*>_RX1_0,

where <*i*> = 0, 1, 2, 3 and

<*k*> = 1, 2, 3, 4, 5, 6.

AXBAR_PART_0_I2S1_RX1_0
Offset: 0x00040

AXBAR_PART_0_I2S2_RX1_0
Offset: 0x00044

AXBAR_PART_0_I2S3_RX1_0
Offset: 0x00048

AXBAR_PART_0_I2S4_RX1_0
Offset: 0x0004c

AXBAR_PART_0_I2S5_RX1_0
Offset: 0x00050

AXBAR_PART_0_I2S6_RX1_0
Offset: 0x00054

AXBAR_PART_1_I2S1_RX1_0
Offset: 0x00240

AXBAR_PART_1_I2S2_RX1_0
Offset: 0x00244

AXBAR_PART_1_I2S3_RX1_0
Offset: 0x00248

AXBAR_PART_1_I2S4_RX1_0
Offset: 0x0024c

AXBAR_PART_1_I2S5_RX1_0
Offset: 0x00250

AXBAR_PART_1_I2S6_RX1_0
Offset: 0x00254

AXBAR_PART_2_I2S1_RX1_0
Offset: 0x00440

AXBAR_PART_2_I2S2_RX1_0
Offset: 0x00444

AXBAR_PART_2_I2S3_RX1_0
Offset: 0x00448

AXBAR_PART_2_I2S4_RX1_0
Offset: 0x0044c

AXBAR_PART_2_I2S5_RX1_0
Offset: 0x00450

AXBAR_PART_2_I2S6_RX1_0
Offset: 0x00454

AXBAR_PART_3_I2S1_RX1_0
Offset: 0x00640

AXBAR_PART_3_I2S2_RX1_0
Offset: 0x00644

AXBAR_PART_3_I2S3_RX1_0
Offset: 0x00648

AXBAR_PART_3_I2S4_RX1_0
Offset: 0x0064c

AXBAR_PART_3_I2S5_RX1_0
Offset: 0x00650

AXBAR_PART_3_I2S6_RX1_0
Offset: 0x00654

Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,0000,xx00,0000,0000,0000,0000,0000)

Bit	Reset	Description
27	0x0	**SFC4_TX1**: **0** = DISABLE **1** = ENABLE
26	0x0	**SFC3_TX1**: **0** = DISABLE **1** = ENABLE
25	0x0	**SFC2_TX1**: **0** = DISABLE **1** = ENABLE
24	0x0	**SFC1_TX1**: **0** = DISABLE **1** = ENABLE
21	0x0	**I2S6_TX1**: **0** = DISABLE **1** = ENABLE
20	0x0	**I2S5_TX1**: **0** = DISABLE **1** = ENABLE
19	0x0	**I2S4_TX1**: **0** = DISABLE **1** = ENABLE
18	0x0	**I2S3_TX1**: **0** = DISABLE **1** = ENABLE
17	0x0	**I2S2_TX1**: **0** = DISABLE **1** = ENABLE
16	0x0	**I2S1_TX1**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
15	0x0	**ADMAIF_TX16**: **0** = DISABLE **1** = ENABLE
14	0x0	**ADMAIF_TX15**: **0** = DISABLE **1** = ENABLE
13	0x0	**ADMAIF_TX14**: **0** = DISABLE **1** = ENABLE
12	0x0	**ADMAIF_TX13**: **0** = DISABLE **1** = ENABLE
11	0x0	**ADMAIF_TX12**: **0** = DISABLE **1** = ENABLE
10	0x0	**ADMAIF_TX11**: **0** = DISABLE **1** = ENABLE
9	0x0	**ADMAIF_TX10**: **0** = DISABLE **1** = ENABLE
8	0x0	**ADMAIF_TX9**: **0** = DISABLE **1** = ENABLE
7	0x0	**ADMAIF_TX8**: **0** = DISABLE **1** = ENABLE
6	0x0	**ADMAIF_TX7**: **0** = DISABLE **1** = ENABLE
5	0x0	**ADMAIF_TX6**: **0** = DISABLE **1** = ENABLE
4	0x0	**ADMAIF_TX5**: **0** = DISABLE **1** = ENABLE
3	0x0	**ADMAIF_TX4**: **0** = DISABLE **1** = ENABLE
2	0x0	**ADMAIF_TX3**: **0** = DISABLE **1** = ENABLE
1	0x0	**ADMAIF_TX2**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
0	0x0	**ADMAIF_TX1**: **0** = DISABLE **1** = ENABLE

AXBAR_PART_<*i*>_SFC<*m*>_RX1_0,

where <*i*> = 0, 1, 2, 3 and

<*m*> = 1, 2, 3, 4.

AXBAR_PART_0_SFC1_RX1_0
Offset: 0x00060

AXBAR_PART_0_SFC2_RX1_0
Offset: 0x00064

AXBAR_PART_0_SFC3_RX1_0
Offset: 0x00068

AXBAR_PART_0_SFC4_RX1_0
Offset: 0x0006c

AXBAR_PART_1_SFC1_RX1_0
Offset: 0x00260

AXBAR_PART_1_SFC2_RX1_0
Offset: 0x00264

AXBAR_PART_1_SFC3_RX1_0
Offset: 0x00268

AXBAR_PART_1_SFC4_RX1_0
Offset: 0x0026c

AXBAR_PART_2_SFC1_RX1_0
Offset: 0x00460

AXBAR_PART_2_SFC2_RX1_0
Offset: 0x00464

AXBAR_PART_2_SFC3_RX1_0
Offset: 0x00468

AXBAR_PART_2_SFC4_RX1_0
Offset: 0x0046c

AXBAR_PART_3_SFC1_RX1_0
Offset: 0x00660

AXBAR_PART_3_SFC2_RX1_0
Offset: 0x00664

AXBAR_PART_3_SFC3_RX1_0
Offset: 0x00668

AXBAR_PART_3_SFC4_RX1_0
Offset: 0x0066c

Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,0000,xx00,0000,0000,0000,0000,0000)

Bit	Reset	Description
27	0x0	**SFC4_TX1**: **0** = DISABLE **1** = ENABLE
26	0x0	**SFC3_TX1**: **0** = DISABLE **1** = ENABLE
25	0x0	**SFC2_TX1**: **0** = DISABLE **1** = ENABLE
24	0x0	**SFC1_TX1**: **0** = DISABLE **1** = ENABLE
21	0x0	**I2S6_TX1**: **0** = DISABLE **1** = ENABLE
20	0x0	**I2S5_TX1**: **0** = DISABLE **1** = ENABLE
19	0x0	**I2S4_TX1**: **0** = DISABLE **1** = ENABLE
18	0x0	**I2S3_TX1**: **0** = DISABLE **1** = ENABLE
17	0x0	**I2S2_TX1**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
16	0x0	**I2S1_TX1**: **0** = DISABLE **1** = ENABLE
15	0x0	**ADMAIF_TX16**: **0** = DISABLE **1** = ENABLE
14	0x0	**ADMAIF_TX15**: **0** = DISABLE **1** = ENABLE
13	0x0	**ADMAIF_TX14**: **0** = DISABLE **1** = ENABLE
12	0x0	**ADMAIF_TX13**: **0** = DISABLE **1** = ENABLE
11	0x0	**ADMAIF_TX12**: **0** = DISABLE **1** = ENABLE
10	0x0	**ADMAIF_TX11**: **0** = DISABLE **1** = ENABLE
9	0x0	**ADMAIF_TX10**: **0** = DISABLE **1** = ENABLE
8	0x0	**ADMAIF_TX9**: **0** = DISABLE **1** = ENABLE
7	0x0	**ADMAIF_TX8**: **0** = DISABLE **1** = ENABLE
6	0x0	**ADMAIF_TX7**: **0** = DISABLE **1** = ENABLE
5	0x0	**ADMAIF_TX6**: **0** = DISABLE **1** = ENABLE
4	0x0	**ADMAIF_TX5**: **0** = DISABLE **1** = ENABLE
3	0x0	**ADMAIF_TX4**: **0** = DISABLE **1** = ENABLE
2	0x0	**ADMAIF_TX3**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
1	0x0	**ADMAIF_TX2**: **0** = DISABLE **1** = ENABLE
0	0x0	**ADMAIF_TX1**: **0** = DISABLE **1** = ENABLE

AXBAR_PART_<*i*>_MIXER1_RX<*p*>_0,

where <*i*> = 0, 1, 2, 3 and

<*p*> = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

AXBAR_PART_0_MIXER1_RX1_0
Offset: 0x00080

AXBAR_PART_0_MIXER1_RX2_0
Offset: 0x00084

AXBAR_PART_0_MIXER1_RX3_0
Offset: 0x00088

AXBAR_PART_0_MIXER1_RX4_0
Offset: 0x0008c

AXBAR_PART_0_MIXER1_RX5_0
Offset: 0x00090

AXBAR_PART_0_MIXER1_RX6_0
Offset: 0x00094

AXBAR_PART_0_MIXER1_RX7_0
Offset: 0x00098

AXBAR_PART_0_MIXER1_RX8_0
Offset: 0x0009c

AXBAR_PART_0_MIXER1_RX9_0
Offset: 0x000a0

AXBAR_PART_0_MIXER1_RX10_0
Offset: 0x000a4

AXBAR_PART_1_MIXER1_RX1_0
Offset: 0x00280

AXBAR_PART_1_MIXER1_RX2_0
Offset: 0x00284

AXBAR_PART_1_MIXER1_RX3_0
Offset: 0x00288

AXBAR_PART_1_MIXER1_RX4_0
Offset: 0x0028c

AXBAR_PART_1_MIXER1_RX5_0
Offset: 0x00290

AXBAR_PART_1_MIXER1_RX6_0
Offset: 0x00294

AXBAR_PART_1_MIXER1_RX7_0
Offset: 0x00298

AXBAR_PART_1_MIXER1_RX8_0
Offset: 0x0029c

AXBAR_PART_1_MIXER1_RX9_0
Offset: 0x002a0

AXBAR_PART_1_MIXER1_RX10_0
Offset: 0x002a4

AXBAR_PART_2_MIXER1_RX1_0
Offset: 0x00480

AXBAR_PART_2_MIXER1_RX2_0
Offset: 0x00484

AXBAR_PART_2_MIXER1_RX3_0
Offset: 0x00488

AXBAR_PART_2_MIXER1_RX4_0
Offset: 0x0048c

AXBAR_PART_2_MIXER1_RX5_0
Offset: 0x00490

AXBAR_PART_2_MIXER1_RX6_0
Offset: 0x00494

AXBAR_PART_2_MIXER1_RX7_0
Offset: 0x00498

AXBAR_PART_2_MIXER1_RX8_0
Offset: 0x0049c

AXBAR_PART_2_MIXER1_RX9_0
Offset: 0x004a0

AXBAR_PART_2_MIXER1_RX10_0
Offset: 0x004a4

AXBAR_PART_3_MIXER1_RX1_0
Offset: 0x00680

AXBAR_PART_3_MIXER1_RX2_0
Offset: 0x00684

AXBAR_PART_3_MIXER1_RX3_0
Offset: 0x00688

AXBAR_PART_3_MIXER1_RX4_0
Offset: 0x0068c

AXBAR_PART_3_MIXER1_RX5_0
Offset: 0x00690

AXBAR_PART_3_MIXER1_RX6_0
Offset: 0x00694

AXBAR_PART_3_MIXER1_RX7_0
Offset: 0x00698

AXBAR_PART_3_MIXER1_RX8_0
Offset: 0x0069c

AXBAR_PART_3_MIXER1_RX9_0
Offset: 0x006a0

AXBAR_PART_3_MIXER1_RX10_0
Offset: 0x006a4

Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,0000,xx00,0000,0000,0000,0000,0000)

Bit	Reset	Description
27	0x0	**SFC4_TX1**: **0** = DISABLE **1** = ENABLE
26	0x0	**SFC3_TX1**: **0** = DISABLE **1** = ENABLE
25	0x0	**SFC2_TX1**: **0** = DISABLE **1** = ENABLE
24	0x0	**SFC1_TX1**: **0** = DISABLE **1** = ENABLE
21	0x0	**I2S6_TX1**: **0** = DISABLE **1** = ENABLE
20	0x0	**I2S5_TX1**: **0** = DISABLE **1** = ENABLE
19	0x0	**I2S4_TX1**: **0** = DISABLE **1** = ENABLE
18	0x0	**I2S3_TX1**: **0** = DISABLE **1** = ENABLE
17	0x0	**I2S2_TX1**: **0** = DISABLE **1** = ENABLE
16	0x0	**I2S1_TX1**: **0** = DISABLE **1** = ENABLE
15	0x0	**ADMAIF_TX16**: **0** = DISABLE **1** = ENABLE
14	0x0	**ADMAIF_TX15**: **0** = DISABLE **1** = ENABLE
13	0x0	**ADMAIF_TX14**: **0** = DISABLE **1** = ENABLE
12	0x0	**ADMAIF_TX13**: **0** = DISABLE **1** = ENABLE
11	0x0	**ADMAIF_TX12**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
10	0x0	**ADMAIF_TX11**: **0** = DISABLE **1** = ENABLE
9	0x0	**ADMAIF_TX10**: **0** = DISABLE **1** = ENABLE
8	0x0	**ADMAIF_TX9**: **0** = DISABLE **1** = ENABLE
7	0x0	**ADMAIF_TX8**: **0** = DISABLE **1** = ENABLE
6	0x0	**ADMAIF_TX7**: **0** = DISABLE **1** = ENABLE
5	0x0	**ADMAIF_TX6**: **0** = DISABLE **1** = ENABLE
4	0x0	**ADMAIF_TX5**: **0** = DISABLE **1** = ENABLE
3	0x0	**ADMAIF_TX4**: **0** = DISABLE **1** = ENABLE
2	0x0	**ADMAIF_TX3**: **0** = DISABLE **1** = ENABLE
1	0x0	**ADMAIF_TX2**: **0** = DISABLE **1** = ENABLE
0	0x0	**ADMAIF_TX1**: **0** = DISABLE **1** = ENABLE

AXBAR_PART_<*i*>_DSPK<*n*>_RX1_0,

where <*i*> = 0, 1, 2, 3 and

<*n*> = 1, 2.

AXBAR_PART_0_DSPK1_RX1_0
Offset: 0x000c0

AXBAR_PART_0_DSPK2_RX1_0
Offset: 0x000c4

AXBAR_PART_1_DSPK1_RX1_0
Offset: 0x002c0

AXBAR_PART_1_DSPK2_RX1_0
Offset: 0x002c4

AXBAR_PART_2_DSPK1_RX1_0
Offset: 0x004c0

AXBAR_PART_2_DSPK2_RX1_0
Offset: 0x004c4

AXBAR_PART_3_DSPK1_RX1_0
Offset: 0x004c0

AXBAR_PART_3_DSPK2_RX1_0
Offset: 0x004c4

Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,0000,xx00,0000,0000,0000,0000,0000)

Bit	Reset	Description
27	0x0	**SFC4_TX1**: **0** = DISABLE **1** = ENABLE
26	0x0	**SFC3_TX1**: **0** = DISABLE **1** = ENABLE
25	0x0	**SFC2_TX1**: **0** = DISABLE **1** = ENABLE
24	0x0	**SFC1_TX1**: **0** = DISABLE **1** = ENABLE
21	0x0	**I2S6_TX1**: **0** = DISABLE **1** = ENABLE
20	0x0	**I2S5_TX1**: **0** = DISABLE **1** = ENABLE
19	0x0	**I2S4_TX1**: **0** = DISABLE **1** = ENABLE
18	0x0	**I2S3_TX1**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
17	0x0	**I2S2_TX1**: **0** = DISABLE **1** = ENABLE
16	0x0	**I2S1_TX1**: **0** = DISABLE **1** = ENABLE
15	0x0	**ADMAIF_TX16**: **0** = DISABLE **1** = ENABLE
14	0x0	**ADMAIF_TX15**: **0** = DISABLE **1** = ENABLE
13	0x0	**ADMAIF_TX14**: **0** = DISABLE **1** = ENABLE
12	0x0	**ADMAIF_TX13**: **0** = DISABLE **1** = ENABLE
11	0x0	**ADMAIF_TX12**: **0** = DISABLE **1** = ENABLE
10	0x0	**ADMAIF_TX11**: **0** = DISABLE **1** = ENABLE
9	0x0	**ADMAIF_TX10**: **0** = DISABLE **1** = ENABLE
8	0x0	**ADMAIF_TX9**: **0** = DISABLE **1** = ENABLE
7	0x0	**ADMAIF_TX8**: **0** = DISABLE **1** = ENABLE
6	0x0	**ADMAIF_TX7**: **0** = DISABLE **1** = ENABLE
5	0x0	**ADMAIF_TX6**: **0** = DISABLE **1** = ENABLE
4	0x0	**ADMAIF_TX5**: **0** = DISABLE **1** = ENABLE
3	0x0	**ADMAIF_TX4**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
2	0x0	**ADMAIF_TX3**: **0** = DISABLE **1** = ENABLE
1	0x0	**ADMAIF_TX2**: **0** = DISABLE **1** = ENABLE
0	0x0	**ADMAIF_TX1**: **0** = DISABLE **1** = ENABLE

AXBAR_PART_<*i*>_AFC<*k*>_RX1_0,

where <*i*> = 0, 1, 2, 3 and

<*k*> = 1, 2, 3, 4, 5, 6.

AXBAR_PART_0_AFC1_RX1_0

Offset: 0x000e0

AXBAR_PART_0_AFC2_RX1_0
Offset: 0x000e4

AXBAR_PART_0_AFC3_RX1_0
Offset: 0x000e8

AXBAR_PART_0_AFC4_RX1_0
Offset: 0x000ec

AXBAR_PART_0_AFC5_RX1_0
Offset: 0x000f0

AXBAR_PART_0_AFC6_RX1_0
Offset: 0x000f4

AXBAR_PART_0_AFC1_RX1_0
Offset: 0x002e0

AXBAR_PART_0_AFC2_RX1_0
Offset: 0x002e4

AXBAR_PART_0_AFC3_RX1_0
Offset: 0x002e8

AXBAR_PART_0_AFC4_RX1_0
Offset: 0x002ec

AXBAR_PART_0_AFC5_RX1_0
Offset: 0x002f0

AXBAR_PART_0_AFC6_RX1_0
Offset: 0x002f4

AXBAR_PART_0_AFC1_RX1_0
Offset: 0x004e0

AXBAR_PART_0_AFC2_RX1_0
Offset: 0x004e4

AXBAR_PART_0_AFC3_RX1_0
Offset: 0x004e8

AXBAR_PART_0_AFC4_RX1_0
Offset: 0x004ec

AXBAR_PART_0_AFC5_RX1_0
Offset: 0x004f0

AXBAR_PART_0_AFC6_RX1_0
Offset: 0x004f4

AXBAR_PART_0_AFC1_RX1_0
Offset: 0x006e0

AXBAR_PART_0_AFC2_RX1_0
Offset: 0x006e4

AXBAR_PART_0_AFC3_RX1_0
Offset: 0x006e8

AXBAR_PART_0_AFC4_RX1_0
Offset: 0x006ec

AXBAR_PART_0_AFC5_RX1_0
Offset: 0x006f0

AXBAR_PART_0_AFC6_RX1_0
Offset: 0x006f4

Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,0000,xx00,0000,0000,0000,0000,0000)

Bit	Reset	Description
27	0x0	**SFC4_TX1**: **0** = DISABLE **1** = ENABLE
26	0x0	**SFC3_TX1**: **0** = DISABLE **1** = ENABLE
25	0x0	**SFC2_TX1**: **0** = DISABLE **1** = ENABLE
24	0x0	**SFC1_TX1**: **0** = DISABLE **1** = ENABLE
21	0x0	**I2S6_TX1**: **0** = DISABLE **1** = ENABLE
20	0x0	**I2S5_TX1**: **0** = DISABLE **1** = ENABLE
19	0x0	**I2S4_TX1**: **0** = DISABLE **1** = ENABLE
18	0x0	**I2S3_TX1**: **0** = DISABLE **1** = ENABLE
17	0x0	**I2S2_TX1**: **0** = DISABLE **1** = ENABLE
16	0x0	**I2S1_TX1**: **0** = DISABLE **1** = ENABLE
15	0x0	**ADMAIF_TX16**: **0** = DISABLE **1** = ENABLE
14	0x0	**ADMAIF_TX15**: **0** = DISABLE **1** = ENABLE
13	0x0	**ADMAIF_TX14**: **0** = DISABLE **1** = ENABLE
12	0x0	**ADMAIF_TX13**: **0** = DISABLE **1** = ENABLE
11	0x0	**ADMAIF_TX12**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
10	0x0	**ADMAIF_TX11**: **0** = DISABLE **1** = ENABLE
9	0x0	**ADMAIF_TX10**: **0** = DISABLE **1** = ENABLE
8	0x0	**ADMAIF_TX9**: **0** = DISABLE **1** = ENABLE
7	0x0	**ADMAIF_TX8**: **0** = DISABLE **1** = ENABLE
6	0x0	**ADMAIF_TX7**: **0** = DISABLE **1** = ENABLE
5	0x0	**ADMAIF_TX6**: **0** = DISABLE **1** = ENABLE
4	0x0	**ADMAIF_TX5**: **0** = DISABLE **1** = ENABLE
3	0x0	**ADMAIF_TX4**: **0** = DISABLE **1** = ENABLE
2	0x0	**ADMAIF_TX3**: **0** = DISABLE **1** = ENABLE
1	0x0	**ADMAIF_TX2**: **0** = DISABLE **1** = ENABLE
0	0x0	**ADMAIF_TX1**: **0** = DISABLE **1** = ENABLE

AXBAR_PART_<*i*>_OPE1_RX1_0

where <*i*> = 0, 1, 2, 3.

AXBAR_PART_0_OPE1_RX1_0
Offset: 0x00100

AXBAR_PART_1_OPE1_RX1_0
Offset: 0x00300

AXBAR_PART_2_OPE1_RX1_0
Offset: 0x00500

AXBAR_PART_3_OPE1_RX1_0
Offset: 0x00700

Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,0000,xx00,0000,0000,0000,0000,0000)

Bit	Reset	Description
27	0x0	**SFC4_TX1**: **0** = DISABLE **1** = ENABLE
26	0x0	**SFC3_TX1**: **0** = DISABLE **1** = ENABLE
25	0x0	**SFC2_TX1**: **0** = DISABLE **1** = ENABLE
24	0x0	**SFC1_TX1**: **0** = DISABLE **1** = ENABLE
21	0x0	**I2S6_TX1**: **0** = DISABLE **1** = ENABLE
20	0x0	**I2S5_TX1**: **0** = DISABLE **1** = ENABLE
19	0x0	**I2S4_TX1**: **0** = DISABLE **1** = ENABLE
18	0x0	**I2S3_TX1**: **0** = DISABLE **1** = ENABLE
17	0x0	**I2S2_TX1**: **0** = DISABLE **1** = ENABLE
16	0x0	**I2S1_TX1**: **0** = DISABLE **1** = ENABLE
15	0x0	**ADMAIF_TX16**: **0** = DISABLE **1** = ENABLE
14	0x0	**ADMAIF_TX15**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
13	0x0	**ADMAIF_TX14**: **0** = DISABLE **1** = ENABLE
12	0x0	**ADMAIF_TX13**: **0** = DISABLE **1** = ENABLE
11	0x0	**ADMAIF_TX12**: **0** = DISABLE **1** = ENABLE
10	0x0	**ADMAIF_TX11**: **0** = DISABLE **1** = ENABLE
9	0x0	**ADMAIF_TX10**: **0** = DISABLE **1** = ENABLE
8	0x0	**ADMAIF_TX9**: **0** = DISABLE **1** = ENABLE
7	0x0	**ADMAIF_TX8**: **0** = DISABLE **1** = ENABLE
6	0x0	**ADMAIF_TX7**: **0** = DISABLE **1** = ENABLE
5	0x0	**ADMAIF_TX6**: **0** = DISABLE **1** = ENABLE
4	0x0	**ADMAIF_TX5**: **0** = DISABLE **1** = ENABLE
3	0x0	**ADMAIF_TX4**: **0** = DISABLE **1** = ENABLE
2	0x0	**ADMAIF_TX3**: **0** = DISABLE **1** = ENABLE
1	0x0	**ADMAIF_TX2**: **0** = DISABLE **1** = ENABLE
0	0x0	**ADMAIF_TX1**: **0** = DISABLE **1** = ENABLE

AXBAR_PART_<*i*>_SPKPROT1_RX1_0,

where <*i*> = 0, 1, 2, 3.

AXBAR_PART_0_SPKPROT1_RX1_0
Offset: 0x00110

AXBAR_PART_1_SPKPROT1_RX1_0
Offset: 0x00310

AXBAR_PART_2_SPKPROT1_RX1_0
Offset: 0x00510

AXBAR_PART_3_SPKPROT1_RX1_0
Offset: 0x00710

Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,0000,xx00,0000,0000,0000,0000,0000)

Bit	Reset	Description
27	0x0	**SFC4_TX1**: **0** = DISABLE **1** = ENABLE
26	0x0	**SFC3_TX1**: **0** = DISABLE **1** = ENABLE
25	0x0	**SFC2_TX1**: **0** = DISABLE **1** = ENABLE
24	0x0	**SFC1_TX1**: **0** = DISABLE **1** = ENABLE
21	0x0	**I2S6_TX1**: **0** = DISABLE **1** = ENABLE
20	0x0	**I2S5_TX1**: **0** = DISABLE **1** = ENABLE
19	0x0	**I2S4_TX1**: **0** = DISABLE **1** = ENABLE
18	0x0	**I2S3_TX1**: **0** = DISABLE **1** = ENABLE
17	0x0	**I2S2_TX1**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
16	0x0	**I2S1_TX1**: **0** = DISABLE **1** = ENABLE
15	0x0	**ADMAIF_TX16**: **0** = DISABLE **1** = ENABLE
14	0x0	**ADMAIF_TX15**: **0** = DISABLE **1** = ENABLE
13	0x0	**ADMAIF_TX14**: **0** = DISABLE **1** = ENABLE
12	0x0	**ADMAIF_TX13**: **0** = DISABLE **1** = ENABLE
11	0x0	**ADMAIF_TX12**: **0** = DISABLE **1** = ENABLE
10	0x0	**ADMAIF_TX11**: **0** = DISABLE **1** = ENABLE
9	0x0	**ADMAIF_TX10**: **0** = DISABLE **1** = ENABLE
8	0x0	**ADMAIF_TX9**: **0** = DISABLE **1** = ENABLE
7	0x0	**ADMAIF_TX8**: **0** = DISABLE **1** = ENABLE
6	0x0	**ADMAIF_TX7**: **0** = DISABLE **1** = ENABLE
5	0x0	**ADMAIF_TX6**: **0** = DISABLE **1** = ENABLE
4	0x0	**ADMAIF_TX5**: **0** = DISABLE **1** = ENABLE
3	0x0	**ADMAIF_TX4**: **0** = DISABLE **1** = ENABLE
2	0x0	**ADMAIF_TX3**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
1	0x0	**ADMAIF_TX2**: **0** = DISABLE **1** = ENABLE
0	0x0	**ADMAIF_TX1**: **0** = DISABLE **1** = ENABLE

AXBAR_PART_<*i*>_MVC<*n*>_RX1_0,

where <*i*> = 0, 1, 2, 3 and

<*n*> = 1, 2.

AXBAR_PART_0_MVC1_RX1_0
Offset: 0x00120

AXBAR_PART_0_MVC2_RX1_0
Offset: 0x00124

AXBAR_PART_1_MVC1_RX1_0
Offset: 0x00320

AXBAR_PART_1_MVC2_RX1_0
Offset: 0x00324

AXBAR_PART_2_MVC1_RX1_0
Offset: 0x00520

AXBAR_PART_2_MVC2_RX1_0
Offset: 0x00524

AXBAR_PART_3_MVC1_RX1_0
Offset: 0x00720

AXBAR_PART_3_MVC2_RX1_0
Offset: 0x00724

Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,0000,xx00,0000,0000,0000,0000,0000)

Bit	Reset	Description
27	0x0	**SFC4_TX1**: **0** = DISABLE **1** = ENABLE
26	0x0	**SFC3_TX1**: **0** = DISABLE **1** = ENABLE
25	0x0	**SFC2_TX1**: **0** = DISABLE **1** = ENABLE
24	0x0	**SFC1_TX1**: **0** = DISABLE **1** = ENABLE
21	0x0	**I2S6_TX1**: **0** = DISABLE **1** = ENABLE
20	0x0	**I2S5_TX1**: **0** = DISABLE **1** = ENABLE
19	0x0	**I2S4_TX1**: **0** = DISABLE **1** = ENABLE
18	0x0	**I2S3_TX1**: **0** = DISABLE **1** = ENABLE
17	0x0	**I2S2_TX1**: **0** = DISABLE **1** = ENABLE
16	0x0	**I2S1_TX1**: **0** = DISABLE **1** = ENABLE
15	0x0	**ADMAIF_TX16**: **0** = DISABLE **1** = ENABLE
14	0x0	**ADMAIF_TX15**: **0** = DISABLE **1** = ENABLE
13	0x0	**ADMAIF_TX14**: **0** = DISABLE **1** = ENABLE
12	0x0	**ADMAIF_TX13**: **0** = DISABLE **1** = ENABLE
11	0x0	**ADMAIF_TX12**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
10	0x0	**ADMAIF_TX11**: **0** = DISABLE **1** = ENABLE
9	0x0	**ADMAIF_TX10**: **0** = DISABLE **1** = ENABLE
8	0x0	**ADMAIF_TX9**: **0** = DISABLE **1** = ENABLE
7	0x0	**ADMAIF_TX8**: **0** = DISABLE **1** = ENABLE
6	0x0	**ADMAIF_TX7**: **0** = DISABLE **1** = ENABLE
5	0x0	**ADMAIF_TX6**: **0** = DISABLE **1** = ENABLE
4	0x0	**ADMAIF_TX5**: **0** = DISABLE **1** = ENABLE
3	0x0	**ADMAIF_TX4**: **0** = DISABLE **1** = ENABLE
2	0x0	**ADMAIF_TX3**: **0** = DISABLE **1** = ENABLE
1	0x0	**ADMAIF_TX2**: **0** = DISABLE **1** = ENABLE
0	0x0	**ADMAIF_TX1**: **0** = DISABLE **1** = ENABLE

AXBAR_PART_<*i*>_AMX<*m*>_RX<*m*>_0,

where <*i*> = 0, 1, 2, 3 and

<*m*> = 1, 2, 3, 4.

AXBAR_PART_0_AMX1_RX1_0
Offset: 0x00140

AXBAR_PART_0_AMX1_RX2_0
Offset: 0x00144

AXBAR_PART_0_AMX1_RX3_0
Offset: 0x00148

AXBAR_PART_0_AMX1_RX4_0
Offset: 0x0014c

AXBAR_PART_0_AMX2_RX1_0
Offset: 0x00150

AXBAR_PART_0_AMX2_RX2_0
Offset: 0x00154

AXBAR_PART_0_AMX2_RX3_0
Offset: 0x00158

AXBAR_PART_0_AMX2_RX4_0
Offset: 0x0015c

AXBAR_PART_0_AMX3_RX1_0
Offset: 0x00160

AXBAR_PART_0_AMX3_RX2_0
Offset: 0x00164

AXBAR_PART_0_AMX3_RX3_0
Offset: 0x00168

AXBAR_PART_0_AMX3_RX4_0
Offset: 0x0016c

AXBAR_PART_0_AMX4_RX1_0
Offset: 0x00190

AXBAR_PART_0_AMX4_RX2_0
Offset: 0x00194

AXBAR_PART_0_AMX4_RX3_0
Offset: 0x00198

AXBAR_PART_0_AMX4_RX4_0
Offset: 0x0019c

AXBAR_PART_1_AMX1_RX1_0
Offset: 0x00340

AXBAR_PART_1_AMX1_RX2_0
Offset: 0x00344

AXBAR_PART_1_AMX1_RX3_0
Offset: 0x00348

AXBAR_PART_1_AMX1_RX4_0
Offset: 0x0034c

AXBAR_PART_1_AMX2_RX1_0
Offset: 0x00350

AXBAR_PART_1_AMX2_RX2_0
Offset: 0x00354

AXBAR_PART_1_AMX2_RX3_0
Offset: 0x00358

AXBAR_PART_1_AMX2_RX4_0
Offset: 0x0035c

AXBAR_PART_1_AMX3_RX1_0
Offset: 0x00360

AXBAR_PART_1_AMX3_RX2_0
Offset: 0x00364

AXBAR_PART_1_AMX3_RX3_0
Offset: 0x00368

AXBAR_PART_1_AMX3_RX4_0
Offset: 0x0036c

AXBAR_PART_1_AMX4_RX1_0
Offset: 0x00390

AXBAR_PART_1_AMX4_RX2_0
Offset: 0x00394

AXBAR_PART_1_AMX4_RX3_0
Offset: 0x00398

AXBAR_PART_1_AMX4_RX4_0
Offset: 0x0039c

AXBAR_PART_2_AMX1_RX1_0
Offset: 0x00540

AXBAR_PART_2_AMX1_RX2_0
Offset: 0x00544

AXBAR_PART_2_AMX1_RX3_0
Offset: 0x00548

AXBAR_PART_2_AMX1_RX4_0
Offset: 0x0054c

AXBAR_PART_2_AMX2_RX1_0
Offset: 0x00550

AXBAR_PART_2_AMX2_RX2_0
Offset: 0x00554

AXBAR_PART_2_AMX2_RX3_0
Offset: 0x00558

AXBAR_PART_2_AMX2_RX4_0
Offset: 0x0055c

AXBAR_PART_2_AMX3_RX1_0
Offset: 0x00560

AXBAR_PART_2_AMX3_RX2_0
Offset: 0x00564

AXBAR_PART_2_AMX3_RX3_0
Offset: 0x00568

AXBAR_PART_2_AMX3_RX4_0
Offset: 0x0056c

AXBAR_PART_2_AMX4_RX1_0
Offset: 0x00590

AXBAR_PART_2_AMX4_RX2_0
Offset: 0x00594

AXBAR_PART_2_AMX4_RX3_0
Offset: 0x00598

AXBAR_PART_2_AMX4_RX4_0
Offset: 0x0059c

AXBAR_PART_3_AMX1_RX1_0
Offset: 0x00740

AXBAR_PART_3_AMX1_RX2_0
Offset: 0x00744

AXBAR_PART_3_AMX1_RX3_0
Offset: 0x00748

AXBAR_PART_3_AMX1_RX4_0
Offset: 0x0074c

AXBAR_PART_3_AMX2_RX1_0
Offset: 0x00750

AXBAR_PART_3_AMX2_RX2_0
Offset: 0x00754

AXBAR_PART_3_AMX2_RX3_0
Offset: 0x00758

AXBAR_PART_3_AMX2_RX4_0
Offset: 0x0075c

AXBAR_PART_3_AMX3_RX1_0
Offset: 0x00760

AXBAR_PART_3_AMX3_RX2_0
Offset: 0x00764

AXBAR_PART_3_AMX3_RX3_0
Offset: 0x00768

AXBAR_PART_3_AMX3_RX4_0
Offset: 0x0076c

AXBAR_PART_3_AMX4_RX1_0
Offset: 0x00790

AXBAR_PART_3_AMX4_RX2_0
Offset: 0x00794

AXBAR_PART_3_AMX4_RX3_0
Offset: 0x00798

AXBAR_PART_3_AMX4_RX4_0
Offset: 0x0079c

Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,0000,xx00,0000,0000,0000,0000,0000)

Bit	Reset	Description
27	0x0	**SFC4_TX1**: **0** = DISABLE **1** = ENABLE
26	0x0	**SFC3_TX1**: **0** = DISABLE **1** = ENABLE
25	0x0	**SFC2_TX1**: **0** = DISABLE **1** = ENABLE
24	0x0	**SFC1_TX1**: **0** = DISABLE **1** = ENABLE
21	0x0	**I2S6_TX1**: **0** = DISABLE **1** = ENABLE
20	0x0	**I2S5_TX1**: **0** = DISABLE **1** = ENABLE
19	0x0	**I2S4_TX1**: **0** = DISABLE **1** = ENABLE
18	0x0	**I2S3_TX1**: **0** = DISABLE **1** = ENABLE
17	0x0	**I2S2_TX1**: **0** = DISABLE **1** = ENABLE
16	0x0	**I2S1_TX1**: **0** = DISABLE **1** = ENABLE
15	0x0	**ADMAIF_TX16**: **0** = DISABLE **1** = ENABLE
14	0x0	**ADMAIF_TX15**: **0** = DISABLE **1** = ENABLE
13	0x0	**ADMAIF_TX14**: **0** = DISABLE **1** = ENABLE
12	0x0	**ADMAIF_TX13**: **0** = DISABLE **1** = ENABLE
11	0x0	**ADMAIF_TX12**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
10	0x0	**ADMAIF_TX11**: **0** = DISABLE **1** = ENABLE
9	0x0	**ADMAIF_TX10**: **0** = DISABLE **1** = ENABLE
8	0x0	**ADMAIF_TX9**: **0** = DISABLE **1** = ENABLE
7	0x0	**ADMAIF_TX8**: **0** = DISABLE **1** = ENABLE
6	0x0	**ADMAIF_TX7**: **0** = DISABLE **1** = ENABLE
5	0x0	**ADMAIF_TX6**: **0** = DISABLE **1** = ENABLE
4	0x0	**ADMAIF_TX5**: **0** = DISABLE **1** = ENABLE
3	0x0	**ADMAIF_TX4**: **0** = DISABLE **1** = ENABLE
2	0x0	**ADMAIF_TX3**: **0** = DISABLE **1** = ENABLE
1	0x0	**ADMAIF_TX2**: **0** = DISABLE **1** = ENABLE
0	0x0	**ADMAIF_TX1**: **0** = DISABLE **1** = ENABLE

AXBAR_PART_<*i*>_ADX<*m*>_RX1_0,

where <*i*> = 0, 1, 2, 3 and

<*m*> = 1, 2, 3, 4.

AXBAR_PART_0_ADX1_RX1_0
Offset: 0x00180

AXBAR_PART_0_ADX2_RX1_0
Offset: 0x00184

AXBAR_PART_0_ADX3_RX1_0
Offset: 0x00188

AXBAR_PART_0_ADX4_RX1_0
Offset: 0x0018c

AXBAR_PART_1_ADX1_RX1_0
Offset: 0x00380

AXBAR_PART_1_ADX2_RX1_0
Offset: 0x00384

AXBAR_PART_1_ADX3_RX1_0
Offset: 0x00388

AXBAR_PART_1_ADX4_RX1_0
Offset: 0x0038c

AXBAR_PART_2_ADX1_RX1_0
Offset: 0x00580

AXBAR_PART_2_ADX2_RX1_0
Offset: 0x00584

AXBAR_PART_2_ADX3_RX1_0
Offset: 0x00588

AXBAR_PART_2_ADX4_RX1_0
Offset: 0x0058c

AXBAR_PART_3_ADX1_RX1_0
Offset: 0x00780

AXBAR_PART_3_ADX2_RX1_0
Offset: 0x00784

AXBAR_PART_3_ADX3_RX1_0
Offset: 0x00788

AXBAR_PART_3_ADX4_RX1_0
Offset: 0x0078c

Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,0000,xx00,0000,0000,0000,0000,0000)

Bit	Reset	Description
27	0x0	**SFC4_TX1**: **0** = DISABLE **1** = ENABLE
26	0x0	**SFC3_TX1**: **0** = DISABLE **1** = ENABLE
25	0x0	**SFC2_TX1**: **0** = DISABLE **1** = ENABLE
24	0x0	**SFC1_TX1**: **0** = DISABLE **1** = ENABLE
21	0x0	**I2S6_TX1**: **0** = DISABLE **1** = ENABLE
20	0x0	**I2S5_TX1**: **0** = DISABLE **1** = ENABLE
19	0x0	**I2S4_TX1**: **0** = DISABLE **1** = ENABLE
18	0x0	**I2S3_TX1**: **0** = DISABLE **1** = ENABLE
17	0x0	**I2S2_TX1**: **0** = DISABLE **1** = ENABLE
16	0x0	**I2S1_TX1**: **0** = DISABLE **1** = ENABLE
15	0x0	**ADMAIF_TX16**: **0** = DISABLE **1** = ENABLE
14	0x0	**ADMAIF_TX15**: **0** = DISABLE **1** = ENABLE
13	0x0	**ADMAIF_TX14**: **0** = DISABLE **1** = ENABLE
12	0x0	**ADMAIF_TX13**: **0** = DISABLE **1** = ENABLE
11	0x0	**ADMAIF_TX12**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
10	0x0	**ADMAIF_TX11**: **0** = DISABLE **1** = ENABLE
9	0x0	**ADMAIF_TX10**: **0** = DISABLE **1** = ENABLE
8	0x0	**ADMAIF_TX9**: **0** = DISABLE **1** = ENABLE
7	0x0	**ADMAIF_TX8**: **0** = DISABLE **1** = ENABLE
6	0x0	**ADMAIF_TX7**: **0** = DISABLE **1** = ENABLE
5	0x0	**ADMAIF_TX6**: **0** = DISABLE **1** = ENABLE
4	0x0	**ADMAIF_TX5**: **0** = DISABLE **1** = ENABLE
3	0x0	**ADMAIF_TX4**: **0** = DISABLE **1** = ENABLE
2	0x0	**ADMAIF_TX3**: **0** = DISABLE **1** = ENABLE
1	0x0	**ADMAIF_TX2**: **0** = DISABLE **1** = ENABLE
0	0x0	**ADMAIF_TX1**: **0** = DISABLE **1** = ENABLE

AXBAR_PART_<*i*>_ASRC1_RX<*q*>_0,

where <*i*> = 0, 1, 2, 3 and

<*q*> = 1, 2, 3, 4, 5, 6, 7.

AXBAR_PART_0_ASRC1_RX1_0
Offset: 0x001b0

AXBAR_PART_0_ASRC1_RX2_0
Offset: 0x001b4

AXBAR_PART_0_ASRC1_RX3_0
Offset: 0x001b8

AXBAR_PART_0_ASRC1_RX4_0
Offset: 0x001bc

AXBAR_PART_0_ASRC1_RX5_0
Offset: 0x001c0

AXBAR_PART_0_ASRC1_RX6_0
Offset: 0x001c4

AXBAR_PART_0_ASRC1_RX7_0
Offset: 0x001c8

AXBAR_PART_1_ASRC1_RX1_0
Offset: 0x003b0

AXBAR_PA3T_1_ASRC1_RX2_0
Offset: 0x003b4

AXBAR_PART_1_ASRC1_RX3_0
Offset: 0x003b8

AXBAR_PART_1_ASRC1_RX4_0
Offset: 0x003bc

AXBAR_PART_1_ASRC1_RX5_0
Offset: 0x003c0

AXBAR_PART_1_ASRC1_RX6_0
Offset: 0x003c4

AXBAR_PART_1_ASRC1_RX7_0
Offset: 0x003c8

AXBAR_PART_2_ASRC1_RX1_0
Offset: 0x005b0

AXBAR_PART_2_ASRC1_RX2_0
Offset: 0x005b4

AXBAR_PART_2_ASRC1_RX3_0
Offset: 0x005b8

AXBAR_PART_2_ASRC1_RX4_0
Offset: 0x005bc

AXBAR_PART_2_ASRC1_RX5_0
Offset: 0x005c0

AXBAR_PART_2_ASRC1_RX6_0
Offset: 0x005c4

AXBAR_PART_2_ASRC1_RX7_0
Offset: 0x005c8

AXBAR_PART_3_ASRC1_RX1_0
Offset: 0x007b0

AXBAR_PART_3_ASRC1_RX2_0
Offset: 0x007b4

AXBAR_PART_3_ASRC1_RX3_0
Offset: 0x007b8

AXBAR_PART_3_ASRC1_RX4_0
Offset: 0x007bc

AXBAR_PART_3_ASRC1_RX5_0
Offset: 0x007c0

AXBAR_PART_3_ASRC1_RX6_0
Offset: 0x007c4

AXBAR_PART_3_ASRC1_RX7_0
Offset: 0x007c8

Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,0000,xx00,0000,0000,0000,0000,0000)

Bit	Reset	Description
27	0x0	**SFC4_TX1**: **0** = DISABLE **1** = ENABLE
26	0x0	**SFC3_TX1**: **0** = DISABLE **1** = ENABLE
25	0x0	**SFC2_TX1**: **0** = DISABLE **1** = ENABLE
24	0x0	**SFC1_TX1**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
21	0x0	**I2S6_TX1**: **0** = DISABLE **1** = ENABLE
20	0x0	**I2S5_TX1**: **0** = DISABLE **1** = ENABLE
19	0x0	**I2S4_TX1**: **0** = DISABLE **1** = ENABLE
18	0x0	**I2S3_TX1**: **0** = DISABLE **1** = ENABLE
17	0x0	**I2S2_TX1**: **0** = DISABLE **1** = ENABLE
16	0x0	**I2S1_TX1**: **0** = DISABLE **1** = ENABLE
15	0x0	**ADMAIF_TX16**: **0** = DISABLE **1** = ENABLE
14	0x0	**ADMAIF_TX15**: **0** = DISABLE **1** = ENABLE
13	0x0	**ADMAIF_TX14**: **0** = DISABLE **1** = ENABLE
12	0x0	**ADMAIF_TX13**: **0** = DISABLE **1** = ENABLE
11	0x0	**ADMAIF_TX12**: **0** = DISABLE **1** = ENABLE
10	0x0	**ADMAIF_TX11**: **0** = DISABLE **1** = ENABLE
9	0x0	**ADMAIF_TX10**: **0** = DISABLE **1** = ENABLE
8	0x0	**ADMAIF_TX9**: **0** = DISABLE **1** = ENABLE
7	0x0	**ADMAIF_TX8**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	0x0	**ADMAIF_TX7**: **0** = DISABLE **1** = ENABLE
5	0x0	**ADMAIF_TX6**: **0** = DISABLE **1** = ENABLE
4	0x0	**ADMAIF_TX5**: **0** = DISABLE **1** = ENABLE
3	0x0	**ADMAIF_TX4**: **0** = DISABLE **1** = ENABLE
2	0x0	**ADMAIF_TX3**: **0** = DISABLE **1** = ENABLE
1	0x0	**ADMAIF_TX2**: **0** = DISABLE **1** = ENABLE
0	0x0	**ADMAIF_TX1**: **0** = DISABLE **1** = ENABLE

7.7.4.2 Audio Hub Configuration (AHC) Registers

AHC_SOFT_RESET_0

Offset: 0x4
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**SOFT_RESET**: **0** = FALSE **1** = TRUE

AHC_CG_0

Offset: 0x8
Read/Write: RW
Parity Protection: N
Reset: 0x00000101 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx1,xxxx,xxx1)

Bit	Reset	Description
8	TRUE	**SLAVE_SLCG_EN**: Second level clock gating enable, **0** = FALSE **1** = TRUE
0	TRUE	**MASTER_SLCG_EN**: Second level clock gating enable, **0** = FALSE **1** = TRUE

AHC_STATUS_0

Offset: 0xc
Read/Write: RO
Parity Protection: N
Reset: 0x00000011 (0bxxxx,0000,0000,0000,xx00,xx00,xx01,xx01)

Bit	Reset	Description
27:16	0x0	**REQUEST_TIMEOUT_COUNTER**: config access timeout value
13:12	FALSE	**LIVE_STATUS**: **0** = FALSE **1** = TRUE
9	FALSE	**MASTER_CLKEN**: **0** = FALSE **1** = TRUE
8	FALSE	**SLAVE_CLKEN**: **0** = FALSE **1** = TRUE
5	FALSE	**HRD_FIFO_FULL**: **0** = FALSE **1** = TRUE
4	TRUE	**HRD_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
1	FALSE	**HWR_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	TRUE	**HWR_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

AHC_INT_STATUS_0

Offset: 0x10
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	CLEAR	**CONFIG_REQUEST_TIMEOUT**: time out status indication bit **0** = CLEAR **1** = SET

AHC_INT_MASK_0

Offset: 0x14
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**CONFIG_REQUEST_TIMEOUT**: **0** = UNMASK **1** = MASK

AHC_INT_SET_0

Offset: 0x18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**CONFIG_REQUEST_TIMEOUT**: **0** = FALSE **1** = TRUE

AHC_INT_CLEAR_0

Offset: 0x1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**CONFIG_REQUEST_TIMEOUT**: **0** = FALSE **1** = TRUE

AHC_CTRL_0

Offset: 0x24
Read/Write: RW
Parity Protection: N
Reset: 0x00000021 (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0010,0001)

Bit	Reset	Description
11:0	0x21	**REQUEST_TIMEOUT_COUNT**: timeout value in terms of ahub clock cycles

AHC_AHUB_INTR_STATUS_0_0

Offset: 0x28
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bx000,0000,0000,0000,0000,0000,xx00,0000)

Bit	Reset	Description
30	CLEAR	**ARAD1**: **0** = CLEAR **1** = SET
29	CLEAR	**AFC6**: **0** = CLEAR **1** = SET
28	CLEAR	**AFC5**: **0** = CLEAR **1** = SET
27	CLEAR	**AFC4**: **0** = CLEAR **1** = SET
26	CLEAR	**AFC3**: **0** = CLEAR **1** = SET
25	CLEAR	**AFC2**: **0** = CLEAR **1** = SET
24	CLEAR	**AFC1**: **0** = CLEAR **1** = SET

Bit	Reset	Description
23	CLEAR	**DMIC4**: **0** = CLEAR **1** = SET
22	CLEAR	**DMIC3**: **0** = CLEAR **1** = SET
21	CLEAR	**DMIC2**: **0** = CLEAR **1** = SET
20	CLEAR	**DMIC1**: **0** = CLEAR **1** = SET
19	CLEAR	**ADX4**: **0** = CLEAR **1** = SET
18	CLEAR	**ADX3**: **0** = CLEAR **1** = SET
17	CLEAR	**ADX2**: **0** = CLEAR **1** = SET
16	CLEAR	**ADX1**: **0** = CLEAR **1** = SET
15	CLEAR	**AMX4**: **0** = CLEAR **1** = SET
14	CLEAR	**AMX3**: **0** = CLEAR **1** = SET
13	CLEAR	**AMX2**: **0** = CLEAR **1** = SET
12	CLEAR	**AMX1**: **0** = CLEAR **1** = SET
11	CLEAR	**SFC4**: **0** = CLEAR **1** = SET
10	CLEAR	**SFC3**: **0** = CLEAR **1** = SET
9	CLEAR	**SFC2**: **0** = CLEAR **1** = SET

Bit	Reset	Description
8	CLEAR	**SFC1**: **0** = CLEAR **1** = SET
5	CLEAR	**I2S6**: **0** = CLEAR **1** = SET
4	CLEAR	**I2S5**: **0** = CLEAR **1** = SET
3	CLEAR	**I2S4**: **0** = CLEAR **1** = SET
2	CLEAR	**I2S3**: **0** = CLEAR **1** = SET
1	CLEAR	**I2S2**: **0** = CLEAR **1** = SET
0	CLEAR	**I2S1**: **0** = CLEAR **1** = SET

AHC_AHUB_INTR_STATUS_1_0

Offset: 0x2c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0x00,xxx0,xxx0,00xx,0000,xx00,xxx0,xxx0)

Bit	Reset	Description
31	CLEAR	**AHC**: **0** = CLEAR **1** = SET
29	CLEAR	**ASRC1**: **0** = CLEAR **1** = SET
28	CLEAR	**XBAR**: **0** = CLEAR **1** = SET
24	CLEAR	**ADMAIF**: **0** = CLEAR **1** = SET
20	CLEAR	**MIXER1**: **0** = CLEAR **1** = SET

Bit	Reset	Description
19	CLEAR	**DSPK2**: **0** = CLEAR **1** = SET
18	CLEAR	**DSPK1**: **0** = CLEAR **1** = SET
15	CLEAR	**IQC2**: **0** = CLEAR **1** = SET
14	CLEAR	**IQC1**: **0** = CLEAR **1** = SET
13	CLEAR	**MDMIF1**: **0** = CLEAR **1** = SET
12	CLEAR	**MDMIF0**: **0** = CLEAR **1** = SET
9	CLEAR	**MVC2**: **0** = CLEAR **1** = SET
8	CLEAR	**MVC1**: **0** = CLEAR **1** = SET
4	CLEAR	**SPKPROT1**: **0** = CLEAR **1** = SET
0	CLEAR	**OPE1**: **0** = CLEAR **1** = SET

AHC_AHUB_ENABLE_STATUS_0_0

Offset: 0x30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bx000,0000,0000,0000,0000,0000,xx00,0000)

Bit	Reset	Description
30	CLEAR	**ARAD1**: **0** = CLEAR **1** = SET
29	CLEAR	**AFC6**: **0** = CLEAR **1** = SET

Bit	Reset	Description
28	CLEAR	**AFC5**: **0** = CLEAR **1** = SET
27	CLEAR	**AFC4**: **0** = CLEAR **1** = SET
26	CLEAR	**AFC3**: **0** = CLEAR **1** = SET
25	CLEAR	**AFC2**: **0** = CLEAR **1** = SET
24	CLEAR	**AFC1**: **0** = CLEAR **1** = SET
23	CLEAR	**DMIC4**: **0** = CLEAR **1** = SET
22	CLEAR	**DMIC3**: **0** = CLEAR **1** = SET
21	CLEAR	**DMIC2**: **0** = CLEAR **1** = SET
20	CLEAR	**DMIC1**: **0** = CLEAR **1** = SET
19	CLEAR	**ADX4**: **0** = CLEAR **1** = SET
18	CLEAR	**ADX3**: **0** = CLEAR **1** = SET
17	CLEAR	**ADX2**: **0** = CLEAR **1** = SET
16	CLEAR	**ADX1**: **0** = CLEAR **1** = SET
15	CLEAR	**AMX4**: **0** = CLEAR **1** = SET
14	CLEAR	**AMX3**: **0** = CLEAR **1** = SET

Bit	Reset	Description
13	CLEAR	**AMX2**: **0** = CLEAR **1** = SET
12	CLEAR	**AMX1**: **0** = CLEAR **1** = SET
11	CLEAR	**SFC4**: **0** = CLEAR **1** = SET
10	CLEAR	**SFC3**: **0** = CLEAR **1** = SET
9	CLEAR	**SFC2**: **0** = CLEAR **1** = SET
8	CLEAR	**SFC1**: **0** = CLEAR **1** = SET
5	CLEAR	**I2S6**: **0** = CLEAR **1** = SET
4	CLEAR	**I2S5**: **0** = CLEAR **1** = SET
3	CLEAR	**I2S4**: **0** = CLEAR **1** = SET
2	CLEAR	**I2S3**: **0** = CLEAR **1** = SET
1	CLEAR	**I2S2**: **0** = CLEAR **1** = SET
0	CLEAR	**I2S1**: **0** = CLEAR **1** = SET

AHC_AHUB_ENABLE_STATUS_1_0

Offset: 0x34
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0x00,xxx0,xxx0,00xx,0000,xx00,xxx0,xxx0)

Bit	Reset	Description
31	CLEAR	**AHC**: **0** = CLEAR **1** = SET
29	CLEAR	**ASRC1**: **0** = CLEAR **1** = SET
28	CLEAR	**XBAR**: **0** = CLEAR **1** = SET
24	CLEAR	**ADMAIF**: **0** = CLEAR **1** = SET
20	CLEAR	**MIXER1**: **0** = CLEAR **1** = SET
19	CLEAR	**DSPK2**: **0** = CLEAR **1** = SET
18	CLEAR	**DSPK1**: **0** = CLEAR **1** = SET
15	CLEAR	**IQC2**: **0** = CLEAR **1** = SET
14	CLEAR	**IQC1**: **0** = CLEAR **1** = SET
13	CLEAR	**MDMIF1**: **0** = CLEAR **1** = SET
12	CLEAR	**MDMIF0**: **0** = CLEAR **1** = SET
9	CLEAR	**MVC2**: **0** = CLEAR **1** = SET
8	CLEAR	**MVC1**: **0** = CLEAR **1** = SET
4	CLEAR	**SPKPROT1**: **0** = CLEAR **1** = SET
0	CLEAR	**OPE1**: **0** = CLEAR **1** = SET

7.7.4.3 Inter-Chip Sound (I2S) Control Registers

I2S_AXBAR_RX_ENABLE_0

Offset: 0x0
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**ENABLE**: **0** = FALSE **1** = TRUE

I2S_AXBAR_RX_SOFT_RESET_0

Offset: 0x4
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DISABLE	**SOFT_RESET**: **0** = DISABLE **1** = ENABLE

I2S_AXBAR_RX_STATUS_0

Offset: 0xc
Read/Write: RO
Parity Protection: N
Reset: 0x00000002 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x010)

Bit	Reset	Description
2	FALSE	**RXCIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
1	TRUE	**RXCIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	FALSE	**RX_ENABLED**: **0** = FALSE **1** = TRUE

I2S_AXBAR_RX_INT_STATUS_0

Offset: 0x10
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	FALSE	**RX_LOWER_WATERMARK**: **0** = FALSE **1** = TRUE
2	FALSE	**RX_NORMAL_WATERMARK**: **0** = FALSE **1** = TRUE
1	FALSE	**RXCIF_FIFO_UNDERRUN**: **0** = FALSE **1** = TRUE
0	FALSE	**RX_DONE**: **0** = FALSE **1** = TRUE

I2S_AXBAR_RX_INT_MASK_0

Offset: 0x14
Read/Write: RW
Parity Protection: N
Reset: 0x0000000f (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1111)

Bit	Reset	Description
3	MASK	**RX_LOWER_WATERMARK**: **0** = UNMASK **1** = MASK
2	MASK	**RX_NORMAL_WATERMARK**: **0** = UNMASK **1** = MASK
1	MASK	**RXCIF_FIFO_UNDERRUN**: **0** = UNMASK **1** = MASK
0	MASK	**RX_DONE**: **0** = UNMASK **1** = MASK

I2S_AXBAR_RX_INT_SET_0

Offset: 0x18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	FALSE	**RX_LOWER_WATERMARK**: **0** = FALSE **1** = TRUE
2	FALSE	**RX_NORMAL_WATERMARK**: **0** = FALSE **1** = TRUE
1	FALSE	**RXCIF_FIFO_UNDERRUN**: **0** = FALSE **1** = TRUE
0	FALSE	**RX_DONE**: **0** = FALSE **1** = TRUE

I2S_AXBAR_RX_INT_CLEAR_0

Offset: 0x1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	FALSE	**RX_LOWER_WATERMARK**: **0** = FALSE **1** = TRUE
2	FALSE	**RX_NORMAL_WATERMARK**: **0** = FALSE **1** = TRUE
1	FALSE	**RXCIF_FIFO_UNDERRUN**: **0** = FALSE **1** = TRUE
0	FALSE	**RX_DONE**: **0** = FALSE **1** = TRUE

I2S_AXBAR_RX_CIF_CTRL_0

Offset: 0x20
Read/Write: RW

Parity Protection: N
Reset: 0x00007700 (0bxx00,0000,0000,0000,x111,x111,0000,0000)

Bit	Reset	Description
29:24	0x0	**FIFO_THRESHOLD**
23:20	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32

Bit	Reset	Description
7:6	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	0x0	**STEREO_MONO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	0x0	**FIFO_SIZE_DOWNSHIFT**
1	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	0x0	**MONO_STEREO_CONV**: **0** = ZERO **1** = COPY

I2S_AXBAR_RX_CTRL_0

Offset: 0x24
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,x000,0000,0000,x000,x000)

Bit	Reset	Description
18:8	0x0	**DATA_OFFSET**
6:4	ZERO	**MASK_BITS**: **0** = ZERO **1** = ONE **2** = TWO **3** = THREE **4** = FOUR **5** = FIVE **6** = SIX **7** = SEVEN
2:1	NOHIGHZ	**HIGHZ_CTRL**: **0** = NOHIGHZ **1** = HIGHZ **2** = HIGHZ_ON_HALF_BIT_CLK
0	MSB_FIRST	**BIT_ORDER**: **0** = MSB_FIRST **1** = LSB_FIRST

I2S_AXBAR_RX_SLOT_CTRL_0

Offset: 0x28
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**SLOT_ENABLES**

I2S_AXBAR_RX_CLK_TRIM_0

Offset: 0x2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4:0	0x0	**TRIM_SEL**

I2S_AXBAR_TX_ENABLE_0

Offset: 0x40
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**ENABLE**: **0** = FALSE **1** = TRUE

I2S_AXBAR_TX_SOFT_RESET_0

Offset: 0x44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DISABLE	**SOFT_RESET**: **0** = DISABLE **1** = ENABLE

I2S_AXBAR_TX_STATUS_0

Offset: 0x4c
Read/Write: RO
Parity Protection: N
Reset: 0x00000002 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x010)

Bit	Reset	Description
2	FALSE	**TXCIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
1	TRUE	**TXCIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	FALSE	**TX_ENABLED**: **0** = FALSE **1** = TRUE

I2S_AXBAR_TX_INT_STATUS_0

Offset: 0x50
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	FALSE	**TX_UPPER_WATERMARK**: **0** = FALSE **1** = TRUE
2	FALSE	**TX_NORMAL_WATERMARK**: **0** = FALSE **1** = TRUE
1	FALSE	**TXCIF_FIFO_OVERRUN**: **0** = FALSE **1** = TRUE
0	FALSE	**TX_DONE**: **0** = FALSE **1** = TRUE

I2S_AXBAR_TX_INT_MASK_0

Offset: 0x54
Read/Write: RW

Parity Protection: N
Reset: 0x0000000f (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1111)

Bit	Reset	Description
3	MASK	**TX_UPPER_WATERMARK**: **0** = UNMASK **1** = MASK
2	MASK	**TX_NORMAL_WATERMARK**: **0** = UNMASK **1** = MASK
1	MASK	**TXCIF_FIFO_OVERRUN**: **0** = UNMASK **1** = MASK
0	MASK	**TX_DONE**: **0** = UNMASK **1** = MASK

I2S_AXBAR_TX_INT_SET_0

Offset: 0x58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	FALSE	**TX_UPPER_WATERMARK**: **0** = FALSE **1** = TRUE
2	FALSE	**TX_NORMAL_WATERMARK**: **0** = FALSE **1** = TRUE
1	FALSE	**TXCIF_FIFO_OVERRUN**: **0** = FALSE **1** = TRUE
0	FALSE	**TX_DONE**: **0** = FALSE **1** = TRUE

I2S_AXBAR_TX_INT_CLEAR_0

Offset: 0x5c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	FALSE	**TX_UPPER_WATERMARK**: **0** = FALSE **1** = TRUE
2	FALSE	**TX_NORMAL_WATERMARK**: **0** = FALSE **1** = TRUE
1	FALSE	**TXCIF_FIFO_OVERRUN**: **0** = FALSE **1** = TRUE
0	FALSE	**TX_DONE**: **0** = FALSE **1** = TRUE

I2S_AXBAR_TX_CIF_CTRL_0

Offset: 0x60
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0bxx00,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_MONO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_STEREO_CONV**: **0** = ZERO **1** = COPY

I2S_AXBAR_TX_CTRL_0

Offset: 0x64
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,x000,0000,0000,x000,xxx0)

Bit	Reset	Description
18:8	0x0	**DATA_OFFSET**
6:4	ZERO	**MASK_BITS**: **0** = ZERO **1** = ONE **2** = TWO **3** = THREE **4** = FOUR **5** = FIVE **6** = SIX **7** = SEVEN
0	MSB_FIRST	**BIT_ORDER**: **0** = MSB_FIRST **1** = LSB_FIRST

I2S_AXBAR_TX_SLOT_CTRL_0

Offset: 0x68
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**SLOT_ENABLES**

I2S_AXBAR_TX_CLK_TRIM_0

Offset: 0x6c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,xxx0,0000)

Bit	Reset	Description
12:8	0x0	**TRIM_SEL_MASTER**: Clock trimmer for pad->Rx pad macro flops in slave mode
4:0	0x0	**TRIM_SEL_SLAVE**: Clock trimmer for pad->Rx pad macro flops in slave mode

I2S_ENABLE_0

Offset: 0x80
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**ENABLE**: **0** = FALSE **1** = TRUE

I2S_SOFT_RESET_0

Offset: 0x84
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DISABLE	**SOFT_RESET**: **0** = DISABLE **1** = ENABLE

I2S_CG_0

Offset: 0x88
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	TRUE	**SLCG_ENABLE**: **0** = FALSE **1** = TRUE

I2S_STATUS_0

Offset: 0x8c
Read/Write: RO
Parity Protection: N
Reset: 0x00000202 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x010,xxxx,x010)

Bit	Reset	Description
12	FALSE	**SLCG_CLKEN**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
10	FALSE	**RXCIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
9	TRUE	**RXCIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
8	FALSE	**RX_ENABLED**: **0** = FALSE **1** = TRUE
2	FALSE	**TXCIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
1	TRUE	**TXCIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	FALSE	**TX_ENABLED**: **0** = FALSE **1** = TRUE

I2S_INT_STATUS_0

Offset: 0x90
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,xxxx,0000)

Bit	Reset	Description
11	FALSE	**RX_LOWER_WATERMARK**: **0** = FALSE **1** = TRUE
10	FALSE	**RX_NORMAL_WATERMARK**: **0** = FALSE **1** = TRUE
9	FALSE	**RXCIF_FIFO_UNDERRUN**: **0** = FALSE **1** = TRUE
8	FALSE	**RX_DONE**: **0** = FALSE **1** = TRUE
3	FALSE	**TX_UPPER_WATERMARK**: **0** = FALSE **1** = TRUE
2	FALSE	**TX_NORMAL_WATERMARK**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
1	FALSE	**TXCIF_FIFO_OVERRUN**: **0** = FALSE **1** = TRUE
0	FALSE	**TX_DONE**: **0** = FALSE **1** = TRUE

I2S_CTRL_0

Offset: 0xa0
Read/Write: See table below
Parity Protection: N
Reset: 0x00080000 (0b0000,0000,xxx0,1xxx,x000,x000,xxxx,x000)

Bit	R/W	Reset	Description
31:24	RW	0x0	**FSYNC_WIDTH**
20	RW	0x0	**EDGE_CTRL**: **0** = POS_EDGE **1** = NEG_EDGE
19	RO	ENABLE	**PIPE_MACRO_EN**: **0** = DISABLE **1** = ENABLE
14:12	RW	LRCK_MODE	**FRAME_FORMAT**: **0** = LRCK_MODE **1** = FSYNC_MODE
10	RW	DISABLE	**MASTER**: **0** = DISABLE **1** = ENABLE
9	RW	LOW	**LRCK_POLARITY**: **0** = LOW **1** = HIGH
8	RW	DISABLE	**LPBK**: **0** = DISABLE **1** = ENABLE
2:0	RW	BIT_SIZE_RSVD	**BIT_SIZE**: **0** = BIT_SIZE_RSVD **1** = BIT_SIZE_8 **2** = BIT_SIZE_12 **3** = BIT_SIZE_16 **4** = BIT_SIZE_20 **5** = BIT_SIZE_24 **6** = BIT_SIZE_28 **7** = BIT_SIZE_32

I2S_TIMING_0

Offset: 0xa4
Read/Write: RW
Parity Protection: N
Reset: 0x0000001f (0bxxxx,xxxx,xxxx,xxxx,xxx0,x000,0001,1111)

Bit	Reset	Description
12	DISABLE	**NON_SYM_EN**: **0** = DISABLE **1** = ENABLE
10:0	0x1f	**CHANNEL_BIT_CNT**

I2S_SLOT_CTRL_0

Offset: 0xa8
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	0x0	**TOTAL_SLOTS**

I2S_CLK_TRIM_0

Offset: 0xac
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,xxx0,0000)

Bit	Reset	Description
12:8	0x0	**SCLK_TRIM_SEL**: Clock trimmer for master mode i2s clock ouutpt to pad
4:0	0x0	**CORE_TRIM_SEL**: Clock trimmer for core i2s clock segment

I2S_CYA_0

Offset: 0xb0
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**CYA_BITS**: spare bits

7.7.4.4 Digital Microphone Control (DMIC) Registers

DMIC_AXBAR_TX_STATUS_0

Offset: 0xc
Read/Write: RO
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01)

Bit	Reset	Description
1	FALSE	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	TRUE	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

DMIC_AXBAR_TX_INT_STATUS_0

Offset: 0x10
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	CLEAR	**TX_UPPER_WATERMARK**: **0** = CLEAR **1** = SET
1	CLEAR	**TX_NORMAL_WATERMARK**: **0** = CLEAR **1** = SET
0	CLEAR	**TX_DONE**: **0** = CLEAR **1** = SET

DMIC_AXBAR_TX_INT_MASK_0

Offset: 0x14
Read/Write: RW

Parity Protection: N
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	MASK	**TX_UPPER_WATERMARK**: **0** = UNMASK **1** = MASK
1	MASK	**TX_NORMAL_WATERMARK**: **0** = UNMASK **1** = MASK
0	MASK	**TX_DONE**: **0** = UNMASK **1** = MASK

DMIC_AXBAR_TX_INT_SET_0

Offset: 0x18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	CLEAR	**TX_UPPER_WATERMARK**: **0** = CLEAR **1** = SET
1	CLEAR	**TX_NORMAL_WATERMARK**: **0** = CLEAR **1** = SET
0	CLEAR	**TX_DONE**: When the SW sets these fields, corresponding interrupts are generated.- **0** = CLEAR **1** = SET

DMIC_AXBAR_TX_INT_CLEAR_0

Offset: 0x1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	CLEAR	**TX_UPPER_WATERMARK**: **0** = CLEAR **1** = SET

Bit	Reset	Description
1	CLEAR	**TX_NORMAL_WATERMARK**: **0** = CLEAR **1** = SET
0	CLEAR	**TX_DONE**: **0** = CLEAR **1** = SET

DMIC_AXBAR_TX_CIF_CTRL_0

Offset: 0x20
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0bxx00,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD

Bit	R/W	Reset	Description
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

DMIC_ENABLE_0

Offset: 0x40
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**ENABLE**: **0** = FALSE **1** = TRUE

DMIC_SOFT_RESET_0

Offset: 0x44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**SOFT_RESET**: **0** = FALSE **1** = TRUE

DMIC_CG_0

Offset: 0x48
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	TRUE	**SLCG_EN**: Second level clock gating enable,for first 2 channels and global/common logic. **0** = FALSE **1** = TRUE

DMIC_STATUS_0

Offset: 0x4c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxx0,xxxx,xxx0)

Bit	Reset	Description
31	FALSE	**CNFG_ERR**: **0** = FALSE **1** = TRUE
8	FALSE	**SLCG_CLKEN**: **0** = FALSE **1** = TRUE
0	FALSE	**ENABLE_STATUS**: **0** = FALSE **1** = TRUE

DMIC_INT_STATUS_0

Offset: 0x50
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	CLEAR	**TX_UPPER_WATERMARK**: **0** = CLEAR **1** = SET
1	CLEAR	**TX_NORMAL_WATERMARK**: **0** = CLEAR **1** = SET
0	CLEAR	**TX_DONE**: **0** = CLEAR **1** = SET

DMIC_CTRL_0

Offset: 0x64
Read/Write: RW
Parity Protection: N
Reset: 0x00000301 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx11,xxx0,xx01)

Bit	Reset	Description
9:8	STEREO	**CHANNEL_SELECT**: **0** = NONE **1** = LEFT **2** = RIGHT **3** = STEREO
4	LEFT	**LRSEL_POLARITY**: **0** = LEFT **1** = RIGHT
1:0	OSR128	**OSR**: **0** = OSR64 **1** = OSR128 **2** = OSR256

DMIC_DBG_CTRL_0

Offset: 0x70
Read/Write: RW
Parity Protection: N
Reset: 0x00000002 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0010)

Bit	Reset	Description
3	DISABLE	**DCR_ENABLE**: **0** = DISABLE **1** = ENABLE
2	DISABLE	**LP_ENABLE**: **0** = DISABLE **1** = ENABLE
1	ENABLE	**SC_ENABLE**: **0** = DISABLE **1** = ENABLE
0	DISABLE	**BYPASS**: **0** = DISABLE **1** = ENABLE

DMIC_DCR_FILTER_GAIN_0

Offset: 0x74
Read/Write: RW
Parity Protection: N
Reset: 0x00800000 (0b0000,0000,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x800000	**GAIN_0**

DMIC_DCR_BIQUAD_0_COEF_<*j*>_0,

where <*j*> = 0, 1, 2, 3, 4.

DMIC_DCR_BIQUAD_0_COEF_0_0
Offset: 0x78

DMIC_DCR_BIQUAD_0_COEF_1_0
Offset: 0x7c

DMIC_DCR_BIQUAD_0_COEF_2_0
Offset: 0x80

DMIC_DCR_BIQUAD_0_COEF_3_0
Offset: 0x84

DMIC_DCR_BIQUAD_0_COEF_4_0
Offset: 0x88

Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**COEF**

DMIC_LP_FILTER_GAIN_0
Offset: 0x8c
Read/Write: RW
Parity Protection: N
Reset: 0x004c255a (0b0000,0000,0100,1100,0010,0101,0101,1010)

Bit	Reset	Description
31:0	0x4c255a	**GAIN_0**

DMIC_LP_BIQUAD_0_COEF_<*j*>_0,

where <*j*> = 0, 1, 2, 3, 4.

DMIC_LP_BIQUAD_0_COEF_0_0
Offset: 0x90

DMIC_LP_BIQUAD_0_COEF_1_0
Offset: 0x94

DMIC_LP_BIQUAD_0_COEF_2_0
Offset: 0x98

DMIC_LP_BIQUAD_0_COEF_3_0
Offset: 0x9c

DMIC_LP_BIQUAD_0_COEF_4_0
Offset: 0xa0

Read/Write: RW
Parity Protection: N
Reset: 0x00380f38 (0b0000,0000,0011,1000,0000,1111,0011,1000)

Bit	Reset	Description
31:0	0x380f38	**COEF**

DMIC_LP_BIQUAD_1_COEF_0_0
Offset: 0xa4
Read/Write: RW
Parity Protection: N
Reset: 0x00800000 (0b0000,0000,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x800000	**COEF_0**

DMIC_LP_BIQUAD_1_COEF_1_0

Offset: 0xa8
Read/Write: RW
Parity Protection: N
Reset: 0x00fe1178 (0b0000,0000,1111,1110,0001,0001,0111,1000)

Bit	Reset	Description
31:0	0xfe1178	**COEF_1**

DMIC_LP_BIQUAD_1_COEF_2_0

Offset: 0xac
Read/Write: RW
Parity Protection: N
Reset: 0x00800000 (0b0000,0000,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x800000	**COEF_2**

DMIC_LP_BIQUAD_1_COEF_3_0

Offset: 0xb0
Read/Write: RW
Parity Protection: N
Reset: 0x00e05f02 (0b0000,0000,1110,0000,0101,1111,0000,0010)

Bit	Reset	Description
31:0	0xe05f02	**COEF_3**

DMIC_LP_BIQUAD_1_COEF_4_0

Offset: 0xb4
Read/Write: RW
Parity Protection: N
Reset: 0x006fc80d (0b0000,0000,0110,1111,1100,1000,0000,1101)

Bit	Reset	Description
31:0	0x6fc80d	**COEF_4**

DMIC_CORRECTION_FILTER_GAIN_0

Offset: 0xb8
Read/Write: RW
Parity Protection: N
Reset: 0x010628f6 (0b0000,0001,0000,0110,0010,1000,1111,0110)

Bit	Reset	Description
31:0	0x10628f6	**GAIN_0**

DMIC_CORRECTION_BIQUAD_0_COEF_0_0

Offset: 0xbc
Read/Write: RW
Parity Protection: N
Reset: 0x00800000 (0b0000,0000,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x800000	**COEF_0**

DMIC_CORRECTION_BIQUAD_0_COEF_1_0

Offset: 0xc0
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**COEF_1**

DMIC_CORRECTION_BIQUAD_0_COEF_2_0

Offset: 0xc4
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**COEF_2**

DMIC_CORRECTION_BIQUAD_0_COEF_3_0

Offset: 0xc8
Read/Write: RW
Parity Protection: N
Reset: 0x0067ffff (0b0000,0000,0110,0111,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0x67ffff	**COEF_3**

DMIC_CORRECTION_BIQUAD_0_COEF_4_0

Offset: 0xcc
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**COEF_4**

DMIC_CORRECTION_BIQUAD_1_COEF_0_0

Offset: 0xd0
Read/Write: RW
Parity Protection: N
Reset: 0x00800000 (0b0000,0000,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x800000	**COEF_0**

DMIC_CORRECTION_BIQUAD_1_COEF_1_0

Offset: 0xd4
Read/Write: RW
Parity Protection: N
Reset: 0x0048f5c2 (0b0000,0000,0100,1000,1111,0101,1100,0010)

Bit	Reset	Description
31:0	0x48f5c2	**COEF_1**

DMIC_CORRECTION_BIQUAD_1_COEF_2_0

Offset: 0xd8
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**COEF_2**

DMIC_CORRECTION_BIQUAD_1_COEF_3_0

Offset: 0xdc
Read/Write: RW

Parity Protection: N
Reset: 0x00562394 (0b0000,0000,0101,0110,0010,0011,1001,0100)

Bit	Reset	Description
31:0	0x562394	**COEF_3**

DMIC_CORRECTION_BIQUAD_1_COEF_4_0

Offset: 0xe0
Read/Write: RW
Parity Protection: N
Reset: 0x00169446 (0b0000,0000,0001,0110,1001,0100,0100,0110)

Bit	Reset	Description
31:0	0x169446	**COEF_4**

7.7.4.5 Digital Speaker (DSPK) Control Registers

DSPK_AXBAR_RX_STATUS_0

Offset: 0xc
Read/Write: RO
Parity Protection: N
Reset: 0x00000002 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x010)

Bit	Reset	Description
2	FALSE	**RXCIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
1	TRUE	**RXCIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	FALSE	**RX_ENABLED**: **0** = FALSE **1** = TRUE

DSPK_AXBAR_RX_INT_STATUS_0

Offset: 0x10
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	FALSE	**RX_LOWER_WATERMARK**: **0** = FALSE **1** = TRUE
3	FALSE	**RX_NORMAL_WATERMARK**: **0** = FALSE **1** = TRUE
2	FALSE	**CODEC_CONFIG_DONE**: **0** = FALSE **1** = TRUE
1	FALSE	**RXCIF_FIFO_UNDERRUN**: **0** = FALSE **1** = TRUE
0	FALSE	**RX_DONE**: **0** = FALSE **1** = TRUE

DSPK_AXBAR_RX_INT_MASK_0

Offset: 0x14
Read/Write: RW
Parity Protection: N
Reset: 0x0000001f (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1,1111)

Bit	Reset	Description
4	MASK	**RX_LOWER_WATERMARK**: **0** = UNMASK **1** = MASK
3	MASK	**RX_NORMAL_WATERMARK**: **0** = UNMASK **1** = MASK
2	MASK	**CODEC_CONFIG_DONE**: **0** = UNMASK **1** = MASK
1	MASK	**RXCIF_FIFO_UNDERRUN**: **0** = UNMASK **1** = MASK
0	MASK	**RX_DONE**: **0** = UNMASK **1** = MASK

DSPK_AXBAR_RX_INT_SET_0

Offset: 0x18
Read/Write: RW

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	FALSE	**RX_LOWER_WATERMARK**: **0** = FALSE **1** = TRUE
3	FALSE	**RX_NORMAL_WATERMARK**: **0** = FALSE **1** = TRUE
2	FALSE	**CODEC_CONFIG_DONE**: **0** = FALSE **1** = TRUE
1	FALSE	**RXCIF_FIFO_UNDERRUN**: **0** = FALSE **1** = TRUE
0	FALSE	**RX_DONE**: **0** = FALSE **1** = TRUE

DSPK_AXBAR_RX_INT_CLEAR_0

Offset: 0x1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	FALSE	**RX_LOWER_WATERMARK**: **0** = FALSE **1** = TRUE
3	FALSE	**RX_NORMAL_WATERMARK**: **0** = FALSE **1** = TRUE
2	FALSE	**CODEC_CONFIG_DONE**: **0** = FALSE **1** = TRUE
1	FALSE	**RXCIF_FIFO_UNDERRUN**: **0** = FALSE **1** = TRUE
0	FALSE	**RX_DONE**: **0** = FALSE **1** = TRUE

DSPK_AXBAR_RX_CIF_CTRL_0

Offset: 0x20
Read/Write: RW
Parity Protection: N
Reset: 0x00007700 (0bxx00,0000,0000,0000,x111,x111,0000,0000)

Bit	Reset	Description
29:24	0x0	**FIFO_THRESHOLD**
23:20	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32

Bit	Reset	Description
7:6	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	0x0	**STEREO_MONO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	0x0	**FIFO_SIZE_DOWNSHIFT**
1	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	0x0	**MONO_STEREO_CONV**: **0** = ZERO **1** = COPY

DSPK_ENABLE_0

Offset: 0x40
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**ENABLE**: **0** = FALSE **1** = TRUE

DSPK_SOFT_RESET_0

Offset: 0x44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DISABLE	**SOFT_RESET**: **0** = DISABLE **1** = ENABLE

DSPK_CG_0

Offset: 0x48
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	TRUE	**SLCG_ENABLE**: **0** = FALSE **1** = TRUE

DSPK_STATUS_0

Offset: 0x4c
Read/Write: RO
Parity Protection: N
Reset: 0x00000200 (0bxxxx,xxxx,xxxx,xxxx,x0x0,x010,xxxx,xxxx)

Bit	Reset	Description
14	FALSE	**CODEC_CONFIG_DONE**: **0** = FALSE **1** = TRUE
12	FALSE	**SLCG_CLKEN**: **0** = FALSE **1** = TRUE
10	FALSE	**RXCIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
9	TRUE	**RXCIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
8	FALSE	**RX_ENABLED**: **0** = FALSE **1** = TRUE

DSPK_INT_STATUS_0

Offset: 0x50
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,xx00,xxxx,xxxx)

Bit	Reset	Description
14	FALSE	**RX_LOWER_WATERMARK**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
13	FALSE	**RX_NORMAL_WATERMARK**: **0** = FALSE **1** = TRUE
12	FALSE	**CODEC_CONFIG_DONE**: **0** = FALSE **1** = TRUE
9	FALSE	**RXCIF_FIFO_UNDERRUN**: **0** = FALSE **1** = TRUE
8	FALSE	**RX_DONE**: **0** = FALSE **1** = TRUE

DSPK_CORE_CTRL_0

Offset: 0x60
Read/Write: RW
Parity Protection: N
Reset: 0x00000310 (0bx000,x000,x000,xxx0,xxxx,xx11,xx01,xxx0)

Bit	Reset	Description
30:28	0x0	**STAGE1_GAIN**: stage-1 Gain
26:24	0x0	**STAGE2_GAIN**: stage-2 Gain
22:20	0x0	**STAGE3_GAIN**: stage-3 Gain
16	NORMAL_MODE	**LOWDELAY_FILTER_MODE**: stage-1 : Normal mode - 103 tap fir filter; lowdelay mode - 73 tap **0** = NORMAL_MODE **1** = LOW_DELAY_MODE
9:8	STEREO	**CHANNEL_SELECT**: for mono/stereo output **0** = NO_CHANNEL **1** = MONO_LEFT **2** = MONO_RIGHT **3** = STEREO
5:4	OSR64	**OSR**: Oversampling ratio **0** = OSR32 **1** = OSR64 **2** = OSR128 **3** = OSR256

Bit	Reset	Description
0	LEFT	**LRSEL_POLARITY**: LEFT= left followed by right **0** = LEFT **1** = RIGHT

DSPK_CODEC_CTRL_0

Offset: 0x64
Read/Write: RW
Parity Protection: N
Reset: 0x03000000 (0bxxxx,xx11,xxxx,xxx0,xxx0,xxx0,0000,0000)

Bit	Reset	Description
25:24	STEREO	**CHANNEL_SELECT**: for mono/stereo output **0** = NO_CHANNEL **1** = MONO_LEFT **2** = MONO_RIGHT **3** = STEREO
16	LSB	**BIT_ORDER**: codec control word bit order to transmit; if '0' then LSB first **0** = LSB **1** = MSB
12	DISABLE	**CODEC_CONFIG_MODE**: in this mode, dspk configures the codec **0** = DISABLE **1** = ENABLE
8:0	0x0	**CNFG_REP_NUM**: codec control word repetetion number

DSPK_CODEC_DATA_0

Offset: 0x68
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:8	0x0	**CH1_CONTROL_WORD**: CH0 codec control word data
7:0	0x0	**CH0_CONTROL_WORD**: CH1 codec control word data

DSPK_CODEC_ENABLE_0

Offset: 0x6c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DISABLE	**CONFIG_START**: dspk codec config start bit **0** = DISABLE **1** = ENABLE

DSPK_SDM_COEF_A_2_0

Offset: 0x74
Read/Write: RW
Parity Protection: N
Reset: 0x000013bb (0bxxxx,xxxx,xxxx,xxxx,0001,0011,1011,1011)

Bit	Reset	Description
15:0	0x13bb	**COEF_A_2**

DSPK_SDM_COEF_A_3_0

Offset: 0x78
Read/Write: RW
Parity Protection: N
Reset: 0x00001cbf (0bxxxx,xxxx,xxxx,xxxx,0001,1100,1011,1111)

Bit	Reset	Description
15:0	0x1cbf	**COEF_A_3**

DSPK_SDM_COEF_A_4_0

Offset: 0x7c
Read/Write: RW
Parity Protection: N
Reset: 0x000029d7 (0bxxxx,xxxx,xxxx,xxxx,0010,1001,1101,0111)

Bit	Reset	Description
15:0	0x29d7	**COEF_A_4**

DSPK_SDM_COEF_A_5_0

Offset: 0x80
Read/Write: RW
Parity Protection: N
Reset: 0x00003782 (0bxxxx,xxxx,xxxx,xxxx,0011,0111,1000,0010)

Bit	Reset	Description
15:0	0x3782	**COEF_A_5**

DSPK_SDM_COEF_C_1_0

Offset: 0x84
Read/Write: RW
Parity Protection: N
Reset: 0x000000a6 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,1010,0110)

Bit	Reset	Description
15:0	0xa6	**COEF_C_1**

DSPK_SDM_COEF_C_2_0

Offset: 0x88
Read/Write: RW
Parity Protection: N
Reset: 0x00001959 (0bxxxx,xxxx,xxxx,xxxx,0001,1001,0101,1001)

Bit	Reset	Description
15:0	0x1959	**COEF_C_2**

DSPK_SDM_COEF_C_3_0

Offset: 0x8c
Read/Write: RW
Parity Protection: N
Reset: 0x00002b9f (0bxxxx,xxxx,xxxx,xxxx,0010,1011,1001,1111)

Bit	Reset	Description
15:0	0x2b9f	**COEF_C_3**

DSPK_SDM_COEF_C_4_0

Offset: 0x90
Read/Write: RW
Parity Protection: N
Reset: 0x00004218 (0bxxxx,xxxx,xxxx,xxxx,0100,0010,0001,1000)

Bit	Reset	Description
15:0	0x4218	**COEF_C_4**

DSPK_SDM_COEF_G_1_0

Offset: 0x94
Read/Write: RW
Parity Protection: N
Reset: 0x00000074 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0111,0100)

Bit	Reset	Description
15:0	0x74	**COEF_G_1**

DSPK_SDM_COEF_G_2_0

Offset: 0x98
Read/Write: RW
Parity Protection: N
Reset: 0x0000007d (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0111,1101)

Bit	Reset	Description
15:0	0x7d	**COEF_G_2**

7.7.4.6 Audio Multiplexer (AMX) Control Registers

AMX_AXBAR_RX_STATUS_0

Offset: 0xc
Read/Write: RO

Parity Protection: N
Reset: 0x00000055 (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0101,0101)

Bit	Reset	Description
11	FALSE	**CH3_PROCESSING_ENABLED**: **0** = FALSE **1** = TRUE
10	FALSE	**CH2_PROCESSING_ENABLED**: **0** = FALSE **1** = TRUE
9	FALSE	**CH1_PROCESSING_ENABLED**: **0** = FALSE **1** = TRUE
8	FALSE	**CH0_PROCESSING_ENABLED**: **0** = FALSE **1** = TRUE
7	FALSE	**ACIF_FIFO4_FULL**: **0** = FALSE **1** = TRUE
6	TRUE	**ACIF_FIFO4_EMPTY**: **0** = FALSE **1** = TRUE
5	FALSE	**ACIF_FIFO3_FULL**: **0** = FALSE **1** = TRUE
4	TRUE	**ACIF_FIFO3_EMPTY**: **0** = FALSE **1** = TRUE
3	FALSE	**ACIF_FIFO2_FULL**: **0** = FALSE **1** = TRUE
2	TRUE	**ACIF_FIFO2_EMPTY**: **0** = FALSE **1** = TRUE
1	FALSE	**ACIF_FIFO1_FULL**: **0** = FALSE **1** = TRUE
0	TRUE	**ACIF_FIFO1_EMPTY**: **0** = FALSE **1** = TRUE

AMX_AXBAR_RX_INT_STATUS_0

Offset: 0x10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,xxxx,0000)

Bit	Reset	Description
11	CLEAR	**RX4_IDLE_CNT_EXPIRED**: **0** = CLEAR **1** = SET
10	CLEAR	**RX3_IDLE_CNT_EXPIRED**: **0** = CLEAR **1** = SET
9	CLEAR	**RX2_IDLE_CNT_EXPIRED**: **0** = CLEAR **1** = SET
8	CLEAR	**RX1_IDLE_CNT_EXPIRED**: **0** = CLEAR **1** = SET
3	CLEAR	**RX4_DONE**: **0** = CLEAR **1** = SET
2	CLEAR	**RX3_DONE**: **0** = CLEAR **1** = SET
1	CLEAR	**RX2_DONE**: **0** = CLEAR **1** = SET
0	CLEAR	**RX1_DONE**: **0** = CLEAR **1** = SET

AMX_AXBAR_RX_INT_MASK_0

Offset: 0x14
Read/Write: RW
Parity Protection: N
Reset: 0x00000f0f (0bxxxx,xxxx,xxxx,xxxx,xxxx,1111,xxxx,1111)

Bit	Reset	Description
11	MASK	**RX4_IDLE_CNT_EXPIRED**: **0** = UNMASK **1** = MASK
10	MASK	**RX3_IDLE_CNT_EXPIRED**: **0** = UNMASK **1** = MASK
9	MASK	**RX2_IDLE_CNT_EXPIRED**: **0** = UNMASK **1** = MASK

Bit	Reset	Description
8	MASK	**RX1_IDLE_CNT_EXPIRED**: **0** = UNMASK **1** = MASK
3	MASK	**RX4_DONE**: **0** = UNMASK **1** = MASK
2	MASK	**RX3_DONE**: **0** = UNMASK **1** = MASK
1	MASK	**RX2_DONE**: **0** = UNMASK **1** = MASK
0	MASK	**RX1_DONE**: **0** = UNMASK **1** = MASK

AMX_AXBAR_RX_INT_SET_0

Offset: 0x18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,xxxx,0000)

Bit	Reset	Description
11	CLEAR	**RX4_IDLE_CNT_EXPIRED**: **0** = CLEAR **1** = SET
10	CLEAR	**RX3_IDLE_CNT_EXPIRED**: **0** = CLEAR **1** = SET
9	CLEAR	**RX2_IDLE_CNT_EXPIRED**: **0** = CLEAR **1** = SET
8	CLEAR	**RX1_IDLE_CNT_EXPIRED**: **0** = CLEAR **1** = SET
3	CLEAR	**RX4_DONE**: This bit is auto cleared after an interrupt in generated **0** = CLEAR **1** = SET
2	CLEAR	**RX3_DONE**: **0** = CLEAR **1** = SET

Bit	Reset	Description
1	CLEAR	**RX2_DONE**: **0** = CLEAR **1** = SET
0	CLEAR	**RX1_DONE**: **0** = CLEAR **1** = SET

AMX_AXBAR_RX_INT_CLEAR_0

Offset: 0x1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,xxxx,0000)

Bit	Reset	Description
11	CLEAR	**RX4_IDLE_CNT_EXPIRED**: **0** = CLEAR **1** = SET
10	CLEAR	**RX3_IDLE_CNT_EXPIRED**: **0** = CLEAR **1** = SET
9	CLEAR	**RX2_IDLE_CNT_EXPIRED**: **0** = CLEAR **1** = SET
8	CLEAR	**RX1_IDLE_CNT_EXPIRED**: **0** = CLEAR **1** = SET
3	CLEAR	**RX4_DONE**: **0** = CLEAR **1** = SET
2	CLEAR	**RX3_DONE**: **0** = CLEAR **1** = SET
1	CLEAR	**RX2_DONE**: **0** = CLEAR **1** = SET
0	CLEAR	**RX1_DONE**: **0** = CLEAR **1** = SET

AMX_AXBAR_RX1_CIF_CTRL_0

Offset: 0x20
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0bxx00,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RO	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32

Bit	R/W	Reset	Description
7:6	RO	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RO	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RO	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RO	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

AMX_AXBAR_RX2_CIF_CTRL_0

Offset: 0x24
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0bxx00,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
19:16	RO	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RO	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RO	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RO	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RO	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

AMX_AXBAR_RX3_CIF_CTRL_0

Offset: 0x28
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0bxx00,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RO	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32

Bit	R/W	Reset	Description
10:8	RO	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RO	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RO	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RO	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RO	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

AMX_AXBAR_RX4_CIF_CTRL_0

Offset: 0x2c
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0bxx00,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
29:24	RO	0x0	**FIFO_THRESHOLD**

Bit	R/W	Reset	Description
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RO	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RO	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD

Bit	R/W	Reset	Description
5:4	RO	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RO	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RO	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

AMX_AXBAR_TX_STATUS_0

Offset: 0x4c
Read/Write: RO
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01)

Bit	Reset	Description
1	FALSE	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	TRUE	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

AMX_AXBAR_TX_INT_STATUS_0

Offset: 0x50
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	CLEAR	**TX_DONE**: **0** = CLEAR **1** = SET

AMX_AXBAR_TX_INT_MASK_0

Offset: 0x54
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**TX_DONE**: **0** = UNMASK **1** = MASK

AMX_AXBAR_TX_INT_SET_0

Offset: 0x58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	CLEAR	**TX_DONE**: This bit is auto cleared after an interrupt in generated **0** = CLEAR **1** = SET

AMX_AXBAR_TX_INT_CLEAR_0

Offset: 0x5c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	CLEAR	**TX_DONE**: **0** = CLEAR **1** = SET

AMX_AXBAR_TX_CIF_CTRL_0

Offset: 0x60
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0bxx00,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RO	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32

Bit	R/W	Reset	Description
7:6	RO	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RO	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RO	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RO	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

AMX_ENABLE_0

Offset: 0x80
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**ENABLE**: **0** = FALSE **1** = TRUE

AMX_SOFT_RESET_0

Offset: 0x84
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**SOFT_RESET**: **0** = FALSE **1** = TRUE

AMX_CG_0

Offset: 0x88
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	TRUE	**SLCG_EN**: Second level clock gating enable **0** = FALSE **1** = TRUE

AMX_STATUS_0

Offset: 0x8c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxx0,xxxx,xxx0)

Bit	Reset	Description
31	FALSE	**CNFG_ERR**: **0** = FALSE **1** = TRUE
8	FALSE	**CLKEN**: **0** = FALSE **1** = TRUE
0	FALSE	**ENABLE_STATUS**: this indicates AMX module Level status **0** = FALSE **1** = TRUE

AMX_INT_STATUS_0

Offset: 0x90
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,xxx0,0000)

Bit	Reset	Description
11	CLEAR	**RX4_IDLE_CNT_EXPIRED**: **0** = CLEAR **1** = SET
10	CLEAR	**RX3_IDLE_CNT_EXPIRED**: **0** = CLEAR **1** = SET
9	CLEAR	**RX2_IDLE_CNT_EXPIRED**: **0** = CLEAR **1** = SET
8	CLEAR	**RX1_IDLE_CNT_EXPIRED**: **0** = CLEAR **1** = SET
4	CLEAR	**TX_DONE**: **0** = CLEAR **1** = SET
3	CLEAR	**RX4_DONE**: **0** = CLEAR **1** = SET
2	CLEAR	**RX3_DONE**: **0** = CLEAR **1** = SET
1	CLEAR	**RX2_DONE**: **0** = CLEAR **1** = SET
0	CLEAR	**RX1_DONE**: **0** = CLEAR **1** = SET

AMX_CTRL_0

Offset: 0xa4
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,xxxx,0000)

Bit	Reset	Description
15:14	0x0	**MSTR_RX_NUM**: **0** = RX1 **1** = RX2 **2** = RX3 **3** = RX4
13:12	0x0	**RX_DEP**: **0** = WT_ON_ALL **1** = WT_ON_ANY **2** = RSVD

Bit	Reset	Description
11	0x0	**RX4_FORCE_DISABLE**: **0** = DISABLE **1** = ENABLE
10	0x0	**RX3_FORCE_DISABLE**: **0** = DISABLE **1** = ENABLE
9	0x0	**RX2_FORCE_DISABLE**: **0** = DISABLE **1** = ENABLE
8	0x0	**RX1_FORCE_DISABLE**: **0** = DISABLE **1** = ENABLE
3	0x0	**RX4_EN**: **0** = DISABLE **1** = ENABLE
2	0x0	**RX3_EN**: **0** = DISABLE **1** = ENABLE
1	0x0	**RX2_EN**: **0** = DISABLE **1** = ENABLE
0	0x0	**RX1_EN**: **0** = DISABLE **1** = ENABLE

AMX_OUT_BYTE_EN0_0

Offset: 0xa8
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**BYTE_EN**: Byte enables for bytes 0 to 31

AMX_OUT_BYTE_EN1_0

Offset: 0xac
Read/Write: RW

Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**BYTE_EN**: Byte enables for bytes 32 to 63

AMX_CYA_0

Offset: 0xb0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CYA**

AMX_AHUBRAMCTL_AMX_CTRL_0

Offset: 0xb8
Read/Write: See table below
Parity Protection: N
Reset: 0x00004000 (0b0xxx,xxxx,0000,0000,x100,xxx0,0000,0000)

Bit	R/W	Reset	Description
31	RO	DONE	**READ_BUSY**: **0** = DONE **1** = BUSY
23:16	RW	0x0	**SEQ_READ_COUNT**
14	RW	WRITE	**RW**: **0** = READ **1** = WRITE
13	RW	DISABLE	**ADDR_INIT_EN**: **0** = DISABLE **1** = ENABLE
12	RW	DISABLE	**SEQ_ACCESS_EN**: **0** = DISABLE **1** = ENABLE
8:0	RW	0x0	**RAM_ADDR**

AMX_AHUBRAMCTL_AMX_DATA_0

Offset: 0xbc
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

AMX_RX1_CTRL_FRAME_PERIOD_0

Program the maximum frame idle count value interms of ahub clocks; if a frame is NOT received within this idle count clock cycles, automatically the channel gets disabled.

Offset: 0xc0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**IDLE_CNT**

AMX_RX2_CTRL_FRAME_PERIOD_0

Offset: 0xc4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**IDLE_CNT**

AMX_RX3_CTRL_FRAME_PERIOD_0

Offset: 0xc8
Read/Write: R/W

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**IDLE_CNT**

AMX_RX4_CTRL_FRAME_PERIOD_0

Offset: 0xcc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**IDLE_CNT**

AMX_RX1_LAST_FRAME_PERIOD_0

This register logs the last frame idle period interms of number of ahub clock.

Offset: 0xd0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**IDLE_CNT**

AMX_RX2_LAST_FRAME_PERIOD_0

Offset: 0xd4
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**IDLE_CNT**

AMX_RX3_LAST_FRAME_PERIOD_0
Offset: 0xd8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**IDLE_CNT**

AMX_RX4_LAST_FRAME_PERIOD_0
Offset: 0xdc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**IDLE_CNT**

7.7.4.7 Audio De-Multiplexer (ADX) Control Registers

ADX_AXBAR_RX_STATUS_0

Offset: 0xc
Read/Write: RO
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01)

Bit	Reset	Description
1	FALSE	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	TRUE	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

ADX_AXBAR_RX_INT_STATUS_0

Offset: 0x10
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	CLEAR	**RX_DONE**: **0** = CLEAR **1** = SET

ADX_AXBAR_RX_INT_MASK_0

Offset: 0x14
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**RX_DONE**: **0** = UNMASK **1** = MASK

ADX_AXBAR_RX_INT_SET_0

Offset: 0x18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	CLEAR	**RX_DONE**: This bit is auto cleared after an interrupt in generated **0** = CLEAR **1** = SET

ADX_AXBAR_RX_INT_CLEAR_0

Offset: 0x1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	CLEAR	**RX_DONE**: **0** = CLEAR **1** = SET

ADX_AXBAR_RX_CIF_CTRL_0

Offset: 0x20
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0bxx00,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
19:16	RO	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RO	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RO	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RO	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RO	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

```
---------------------------------------
 ADX Tx path (w.r.to AXBAR) Registers
---------------------------------------
```

ADX_AXBAR_TX_STATUS_0

Offset: 0x4c
Read/Write: RO
Parity Protection: N
Reset: 0x00000055 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0101,0101)

Bit	Reset	Description
7	FALSE	**ACIF_FIFO4_FULL**: **0** = FALSE **1** = TRUE
6	TRUE	**ACIF_FIFO4_EMPTY**: **0** = FALSE **1** = TRUE
5	FALSE	**ACIF_FIFO3_FULL**: **0** = FALSE **1** = TRUE
4	TRUE	**ACIF_FIFO3_EMPTY**: **0** = FALSE **1** = TRUE
3	FALSE	**ACIF_FIFO2_FULL**: **0** = FALSE **1** = TRUE
2	TRUE	**ACIF_FIFO2_EMPTY**: **0** = FALSE **1** = TRUE
1	FALSE	**ACIF_FIFO1_FULL**: **0** = FALSE **1** = TRUE
0	TRUE	**ACIF_FIFO1_EMPTY**: **0** = FALSE **1** = TRUE

ADX_AXBAR_TX_INT_STATUS_0

Offset: 0x50
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	CLEAR	**TX4_DONE**: **0** = CLEAR **1** = SET
2	CLEAR	**TX3_DONE**: **0** = CLEAR **1** = SET

Bit	Reset	Description
1	CLEAR	**TX2_DONE**: **0** = CLEAR **1** = SET
0	CLEAR	**TX1_DONE**: **0** = CLEAR **1** = SET

ADX_AXBAR_TX_INT_MASK_0

Offset: 0x54
Read/Write: RW
Parity Protection: N
Reset: 0x0000000f (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1111)

Bit	Reset	Description
3	MASK	**TX4_DONE**: **0** = UNMASK **1** = MASK
2	MASK	**TX3_DONE**: **0** = UNMASK **1** = MASK
1	MASK	**TX2_DONE**: **0** = UNMASK **1** = MASK
0	MASK	**TX1_DONE**: **0** = UNMASK **1** = MASK

ADX_AXBAR_TX_INT_SET_0

Offset: 0x58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	CLEAR	**TX4_DONE**: This bit is auto cleared after an interrupt in generated **0** = CLEAR **1** = SET
2	CLEAR	**TX3_DONE**: **0** = CLEAR **1** = SET

Bit	Reset	Description
1	CLEAR	**TX2_DONE**: **0** = CLEAR **1** = SET
0	CLEAR	**TX1_DONE**: **0** = CLEAR **1** = SET

ADX_AXBAR_TX_INT_CLEAR_0

Offset: 0x5c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	CLEAR	**TX4_DONE**: **0** = CLEAR **1** = SET
2	CLEAR	**TX3_DONE**: **0** = CLEAR **1** = SET
1	CLEAR	**TX2_DONE**: **0** = CLEAR **1** = SET
0	CLEAR	**TX1_DONE**: **0** = CLEAR **1** = SET

ADX_AXBAR_TX1_CIF_CTRL_0

Offset: 0x60
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0bxx00,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
29:24	RO	0x0	**FIFO_THRESHOLD**

Bit	R/W	Reset	Description
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RO	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RO	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD

Bit	R/W	Reset	Description
5:4	RO	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RO	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RO	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADX_AXBAR_TX2_CIF_CTRL_0

Offset: 0x64
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0bxx00,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
19:16	RO	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RO	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RO	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RO	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RO	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADX_AXBAR_TX3_CIF_CTRL_0

Offset: 0x68
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0bxx00,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RO	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32

Bit	R/W	Reset	Description
7:6	RO	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RO	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RO	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RO	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADX_AXBAR_TX4_CIF_CTRL_0

Offset: 0x6c
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0bxx00,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
19:16	RO	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RO	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RO	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RO	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RO	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADX_ENABLE_0

Offset: 0x80
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**ENABLE**: **0** = FALSE **1** = TRUE

ADX_SOFT_RESET_0

Offset: 0x84
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADX_CG_0

Offset: 0x88
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	TRUE	**SLCG_EN**: Second level clock gating enable,for first 2 channels and global/common logic. **0** = FALSE **1** = TRUE

ADX_STATUS_0

Offset: 0x8c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxx0,xxxx,xxx0)

Bit	Reset	Description
31	FALSE	**CNFG_ERR**: **0** = FALSE **1** = TRUE
8	FALSE	**CLKEN**: **0** = FALSE **1** = TRUE
0	FALSE	**ENABLE_STATUS**: **0** = FALSE **1** = TRUE

ADX_INT_STATUS_0

Offset: 0x90
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	CLEAR	**RX_DONE**: **0** = CLEAR **1** = SET
3	CLEAR	**TX4_DONE**: **0** = CLEAR **1** = SET
2	CLEAR	**TX3_DONE**: **0** = CLEAR **1** = SET
1	CLEAR	**TX2_DONE**: **0** = CLEAR **1** = SET
0	CLEAR	**TX1_DONE**: **0** = CLEAR **1** = SET

ADX_CTRL_0

Offset: 0xa4
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,xxxx,0000)

Bit	Reset	Description
11	0x0	**TX4_FORCE_DISABLE**: **0** = DISABLE **1** = ENABLE
10	0x0	**TX3_FORCE_DISABLE**: **0** = DISABLE **1** = ENABLE
9	0x0	**TX2_FORCE_DISABLE**: **0** = DISABLE **1** = ENABLE
8	0x0	**TX1_FORCE_DISABLE**: **0** = DISABLE **1** = ENABLE
3	0x0	**TX4_EN**: **0** = DISABLE **1** = ENABLE
2	0x0	**TX3_EN**: **0** = DISABLE **1** = ENABLE
1	0x0	**TX2_EN**: **0** = DISABLE **1** = ENABLE
0	0x0	**TX1_EN**: **0** = DISABLE **1** = ENABLE

ADX_IN_BYTE_EN0_0

Offset: 0xa8
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**BYTE_EN**: Byte enables for bytes 0 to 31

ADX_IN_BYTE_EN1_0

Offset: 0xac
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**BYTE_EN**: Byte enables for bytes 32 to 63

ADX_AHUBRAMCTL_ADX_CTRL_0

Offset: 0xb8
Read/Write: See table below
Parity Protection: N
Reset: 0x00004000 (0b0xxx,xxxx,0000,0000,x100,xxx0,0000,0000)

Bit	R/W	Reset	Description
31	RO	DONE	**READ_BUSY**: **0** = DONE **1** = BUSY
23:16	RW	0x0	**SEQ_READ_COUNT**
14	RW	WRITE	**RW**: **0** = READ **1** = WRITE
13	RW	DISABLE	**ADDR_INIT_EN**: **0** = DISABLE **1** = ENABLE
12	RW	DISABLE	**SEQ_ACCESS_EN**: **0** = DISABLE **1** = ENABLE
8:0	RW	0x0	**RAM_ADDR**

ADX_AHUBRAMCTL_ADX_DATA_0

Offset: 0xbc
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

7.7.4.8 Mixer Control Registers

MIXER_AXBAR_RX<*i*>_SOFT_RESET_0,

where *i* = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

MIXER_AXBAR_RX1_SOFT_RESET_0
Offset: 0x00004

MIXER_AXBAR_RX2_SOFT_RESET_0
Offset: 0x00044

MIXER_AXBAR_RX3_SOFT_RESET_0
Offset: 0x00084

MIXER_AXBAR_RX4_SOFT_RESET_0
Offset: 0x000c4

MIXER_AXBAR_RX5_SOFT_RESET_0
Offset: 0x00104

MIXER_AXBAR_RX6_SOFT_RESET_0
Offset: 0x00144

MIXER_AXBAR_RX7_SOFT_RESET_0
Offset: 0x00184

MIXER_AXBAR_RX8_SOFT_RESET_0
Offset: 0x001c4

MIXER_AXBAR_RX9_SOFT_RESET_0
Offset: 0x00204

MIXER_AXBAR_RX10_SOFT_RESET_0
Offset: 0x00244

Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**SOFT_RESET**: **0** = FALSE **1** = TRUE

MIXER_AXBAR_RX<*i*>_STATUS_0,

where *i* = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

MIXER_AXBAR_RX1_STATUS_0
Offset: 0x00010

MIXER_AXBAR_RX2_STATUS_0
Offset: 0x00050

MIXER_AXBAR_RX3_STATUS_0

Offset: 0x00090

MIXER_AXBAR_RX4_STATUS_0
Offset: 0x000d0

MIXER_AXBAR_RX5_STATUS_0
Offset: 0x00110

MIXER_AXBAR_RX6_STATUS_0
Offset: 0x00150

MIXER_AXBAR_RX7_STATUS_0
Offset: 0x00190

MIXER_AXBAR_RX8_STATUS_0
Offset: 0x001d0

MIXER_AXBAR_RX9_STATUS_0
Offset: 0x00210

MIXER_AXBAR_RX10_STATUS_0
Offset: 0x00250

Read/Write: RO
Parity Protection: N
Reset: 0x00000001 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxx0,xxxx,xx01)

Bit	Reset	Description
8	FALSE	**CONFIG_BUSY**: **0** = FALSE **1** = TRUE
1	FALSE	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	TRUE	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

MIXER_AXBAR_RX<*i*>_CIF_CTRL_0,

where *i* = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

MIXER_AXBAR_RX1_CIF_CTRL_0
Offset: 0x00024

MIXER_AXBAR_RX2_CIF_CTRL_0
Offset: 0x00064

MIXER_AXBAR_RX3_CIF_CTRL_0
Offset: 0x000a4

MIXER_AXBAR_RX4_CIF_CTRL_0
Offset: 0x000e4

MIXER_AXBAR_RX5_CIF_CTRL_0
Offset: 0x00124

MIXER_AXBAR_RX6_CIF_CTRL_0
Offset: 0x00164

MIXER_AXBAR_RX7_CIF_CTRL_0
Offset: 0x001a4

MIXER_AXBAR_RX8_CIF_CTRL_0
Offset: 0x001e4

MIXER_AXBAR_RX9_CIF_CTRL_0
Offset: 0x00224

MIXER_AXBAR_RX10_CIF_CTRL_0
Offset: 0x00264

Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0bxx00,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
29:24	RO	0x0	**FIFO_THRESHOLD**

Bit	R/W	Reset	Description
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD

Bit	R/W	Reset	Description
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

MIXER_AXBAR_RX<*i*>_CTRL_0,

where *i* = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

MIXER_AXBAR_RX1_CTRL_0
Offset: 0x00028

MIXER_AXBAR_RX2_CTRL_0
Offset: 0x00068

MIXER_AXBAR_RX3_CTRL_0
Offset: 0x000a8

MIXER_AXBAR_RX4_CTRL_0
Offset: 0x000e8

MIXER_AXBAR_RX5_CTRL_0
Offset: 0x00128

MIXER_AXBAR_RX6_CTRL_0
Offset: 0x00168

MIXER_AXBAR_RX7_CTRL_0
Offset: 0x001a8

MIXER_AXBAR_RX8_CTRL_0
Offset: 0x001e8

MIXER_AXBAR_RX9_CTRL_0
Offset: 0x00228

MIXER_AXBAR_RX10_CTRL_0
Offset: 0x00268

Read/Write: RW
Parity Protection: N
Reset: 0x00010823 (0b0xxx,xxx0,xxx0,0001,0000,1000,0010,0011)

Bit	Reset	Description
24	FALSE	**SAMPLE_COUNTER_RESET**: **0** = FALSE **1** = TRUE
20:16	0x1	**ACT_THRESHOLD**
15:0	0x823	**DEACT_THRESHOLD**

MIXER_AXBAR_RX<*i*>_PEAK_CTRL_0,

where *i* = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

MIXER_AXBAR_RX1_PEAK_CTRL_0
Offset: 0x0002c

MIXER_AXBAR_RX2_PEAK_CTRL_0
Offset: 0x0006c

MIXER_AXBAR_RX3_PEAK_CTRL_0
Offset: 0x000ac

MIXER_AXBAR_RX4_PEAK_CTRL_0
Offset: 0x000ec

MIXER_AXBAR_RX5_PEAK_CTRL_0
Offset: 0x0012c

MIXER_AXBAR_RX6_PEAK_CTRL_0
Offset: 0x0016c

MIXER_AXBAR_RX7_PEAK_CTRL_0
Offset: 0x001ac

MIXER_AXBAR_RX8_PEAK_CTRL_0
Offset: 0x001ec

MIXER_AXBAR_RX9_PEAK_CTRL_0
Offset: 0x0022c

MIXER_AXBAR_RX10_PEAK_CTRL_0
Offset: 0x0026c

Read/Write: RW
Parity Protection: N
Reset: 0x000012c0 (0b0000,0000,0000,0000,0001,0010,1100,0000)

Bit	Reset	Description
31	WINDOW_BASED	**PEAK_TYPE**: **0** = WINDOW_BASED **1** = RESET_ON_READ
30:0	0x12c0	**PEAK_WIN**

MIXER_AXBAR_RX<*i*>_SAMPLE_COUNT_0,

where *i* = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

MIXER_AXBAR_RX1_SAMPLE_COUNT_0
Offset: 0x00030

MIXER_AXBAR_RX2_SAMPLE_COUNT_0
Offset: 0x00070

MIXER_AXBAR_RX3_SAMPLE_COUNT_0
Offset: 0x000b0

MIXER_AXBAR_RX4_SAMPLE_COUNT_0
Offset: 0x000f0

MIXER_AXBAR_RX5_SAMPLE_COUNT_0
Offset: 0x00130

MIXER_AXBAR_RX6_SAMPLE_COUNT_0
Offset: 0x00170

MIXER_AXBAR_RX7_SAMPLE_COUNT_0
Offset: 0x001b0

MIXER_AXBAR_RX8_SAMPLE_COUNT_0
Offset: 0x001f0

MIXER_AXBAR_RX9_SAMPLE_COUNT_0
Offset: 0x00230

MIXER_AXBAR_RX10_SAMPLE_COUNT_0
Offset: 0x00270

Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SAMPLE_CNT**

MIXER_AXBAR_TX<*j*>_ENABLE_0,

where *j* = 1, 2, 3, 4, 5.

MIXER_AXBAR_TX1_ENABLE_0

Offset: 0x00280

MIXER_AXBAR_TX2_ENABLE_0
Offset: 0x002c0

MIXER_AXBAR_TX3_ENABLE_0
Offset: 0x30000

MIXER_AXBAR_TX4_ENABLE_0
Offset: 0x00340

MIXER_AXBAR_TX5_ENABLE_0
Offset: 0x00380

Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**ENABLE**: **0** = FALSE **1** = TRUE

MIXER_AXBAR_TX<*j*>_SOFT_RESET_0,

where *j* = 1, 2, 3, 4, 5.

MIXER_AXBAR_TX1_SOFT_RESET_0
Offset: 0x00284

MIXER_AXBAR_TX2_SOFT_RESET_0
Offset: 0x002c4

MIXER_AXBAR_TX3_SOFT_RESET_0
Offset: 0x00304

MIXER_AXBAR_TX4_SOFT_RESET_0
Offset: 0x00344

MIXER_AXBAR_TX5_SOFT_RESET_0
Offset: 0x00384

Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**SOFT_RESET**: **0** = FALSE **1** = TRUE

MIXER_AXBAR_TX<*j*>_STATUS_0,

where *j* = 1, 2, 3, 4, 5.

MIXER_AXBAR_TX1_STATUS_0
Offset: 0x00290

MIXER_AXBAR_TX2_STATUS_0
Offset: 0x002d0

MIXER_AXBAR_TX3_STATUS_0
Offset: 0x00310

MIXER_AXBAR_TX4_STATUS_0
Offset: 0x00350

MIXER_AXBAR_TX5_STATUS_0
Offset: 0x00390

Read/Write: RO
Parity Protection: N
Reset: 0x00000002 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x010)

Bit	Reset	Description
2	FALSE	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
1	TRUE	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	FALSE	**ENABLE_STATUS**: **0** = FALSE **1** = TRUE

MIXER_AXBAR_TX<*j*>_INT_STATUS_0,

where *j* = 1, 2, 3, 4, 5.

MIXER_AXBAR_TX1_INT_STATUS_0
Offset: 0x00294

MIXER_AXBAR_TX2_INT_STATUS_0
Offset: 0x002d4

MIXER_AXBAR_TX3_INT_STATUS_0
Offset: 0x00314

MIXER_AXBAR_TX4_INT_STATUS_0
Offset: 0x00354

MIXER_AXBAR_TX5_INT_STATUS_0
Offset: 0x00394

Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	CLEAR	**TX_DONE**: **0** = CLEAR **1** = SET

MIXER_AXBAR_TX<*j*>_INT_MASK_0,

where *j* = 1, 2, 3, 4, 5.

MIXER_AXBAR_TX1_INT_MASK_0
Offset: 0x00298

MIXER_AXBAR_TX2_INT_MASK_0
Offset: 0x002d8

MIXER_AXBAR_TX3_INT_MASK_0
Offset: 0x00318

MIXER_AXBAR_TX4_INT_MASK_0
Offset: 0x00358

MIXER_AXBAR_TX5_INT_MASK_0
Offset: 0x00398

Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**TX_DONE**: **0** = UNMASK **1** = MASK

MIXER_AXBAR_TX<*j*>_INT_SET_0,

where *j* = 1, 2, 3, 4, 5.

MIXER_AXBAR_TX1_INT_SET_0
Offset: 0x0029c

MIXER_AXBAR_TX2_INT_SET_0
Offset: 0x002dc

MIXER_AXBAR_TX3_INT_SET_0
Offset: 0x0031c

MIXER_AXBAR_TX4_INT_SET_0
Offset: 0x0035c

MIXER_AXBAR_TX5_INT_SET_0
Offset: 0x0039c

Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	CLEAR	**TX_DONE**: **0** = CLEAR **1** = SET

MIXER_AXBAR_TX<*j*>_INT_CLEAR_0,

where *j* = 1, 2, 3, 4, 5.

MIXER_AXBAR_TX1_INT_CLEAR_0
Offset: 0x002a0

MIXER_AXBAR_TX2_INT_CLEAR_0
Offset: 0x002e0

MIXER_AXBAR_TX3_INT_CLEAR_0
Offset: 0x00320

MIXER_AXBAR_TX4_INT_CLEAR_0
Offset: 0x00360

MIXER_AXBAR_TX5_INT_CLEAR_0
Offset: 0x003a0

Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	CLEAR	**TX_DONE**: **0** = CLEAR **1** = SET

MIXER_AXBAR_TX<*j*>_CIF_CTRL_0,

where j = 1, 2, 3, 4, 5.

MIXER_AXBAR_TX1_CIF_CTRL_0
Offset: 0x002a4

MIXER_AXBAR_TX2_CIF_CTRL_0
Offset: 0x002e4

MIXER_AXBAR_TX3_CIF_CTRL_0
Offset: 0x00324

MIXER_AXBAR_TX4_CIF_CTRL_0
Offset: 0x00364

MIXER_AXBAR_TX5_CIF_CTRL_0
Offset: 0x003a4

Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0bxx00,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

MIXER_AXBAR_TX<*j*>_ADDER_CONFIG_0,

where j = 1, 2, 3, 4, 5.

MIXER_AXBAR_TX1_ADDER_CONFIG_0
Offset: 0x002a8

MIXER_AXBAR_TX2_ADDER_CONFIG_0
Offset: 0x002e8

MIXER_AXBAR_TX3_ADDER_CONFIG_0
Offset: 0x00328

MIXER_AXBAR_TX4_ADDER_CONFIG_0
Offset: 0x00368

MIXER_AXBAR_TX5_ADDER_CONFIG_0
Offset: 0x003a8

Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9	DISABLE	**RX10_INPUT_ENABLE**: **0** = DISABLE **1** = ENABLE
8	DISABLE	**RX9_INPUT_ENABLE**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**RX8_INPUT_ENABLE**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**RX7_INPUT_ENABLE**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**RX6_INPUT_ENABLE**: **0** = DISABLE **1** = ENABLE
4	DISABLE	**RX5_INPUT_ENABLE**: **0** = DISABLE **1** = ENABLE
3	DISABLE	**RX4_INPUT_ENABLE**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
2	DISABLE	**RX3_INPUT_ENABLE**: **0** = DISABLE **1** = ENABLE
1	DISABLE	**RX2_INPUT_ENABLE**: **0** = DISABLE **1** = ENABLE
0	DISABLE	**RX1_INPUT_ENABLE**: **0** = DISABLE **1** = ENABLE

MIXER_ENABLE_0

Offset: 0x400
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**ENABLE**: MIXER global enable bit, When disabled, MIXER does not disable till all the data in the datapipe has been processed and sent out. the "datapipe" here means, mixer core block and TXCIF, not including RXCIF. MIXER core will finish current frame and go bank to IDLE. It will not pop data from RXCIF any more, even there are still some data in RXCIF fifo. Read back of the register will also give module enable status **0** = FALSE **1** = TRUE

MIXER_SOFT_RESET_0

Offset: 0x404
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**SOFT_RESET**: Soft Reset is a self-clearing bit. Soft Reset resets all FSM, flushes flow control FIFO and resets the state registers. It also bring the module back to disabled state (without flushing data in the pipe) **0** = FALSE **1** = TRUE

MIXER_CG_0

Offset: 0x408
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	TRUE	**SLCG_EN**: Second level clock gating enable **0** = FALSE **1** = TRUE

MIXER_STATUS_0

Offset: 0x410
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	FALSE	**SLCG_CLKEN**: **0** = FALSE **1** = TRUE
0	FALSE	**ENABLE_STATUS**: **0** = FALSE **1** = TRUE

MIXER_INT_STATUS_0

Offset: 0x414
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxx0,0000,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
20	CLEAR	**TX5_DONE**: **0** = CLEAR **1** = SET
19	CLEAR	**TX4_DONE**: **0** = CLEAR **1** = SET
18	CLEAR	**TX3_DONE**: **0** = CLEAR **1** = SET
17	CLEAR	**TX2_DONE**: **0** = CLEAR **1** = SET
16	CLEAR	**TX1_DONE**: **0** = CLEAR **1** = SET
0	CLEAR	**MIXER_TX_DONE**: **0** = CLEAR **1** = SET

MIXER_AHUBRAMCTL_GAIN_CONFIG_RAM_CTRL_0

Offset: 0x42c
Read/Write: See table below
Parity Protection: N
Reset: 0x00004000 (0b0xxx,xxxx,0000,0000,x100,xxx0,0000,0000)

Bit	R/W	Reset	Description
31	RO	DONE	**READ_BUSY**: **0** = DONE **1** = BUSY
23:16	RW	0x0	**SEQ_READ_COUNT**
14	RW	WRITE	**RW**: **0** = READ **1** = WRITE
13	RW	DISABLE	**ADDR_INIT_EN**: **0** = DISABLE **1** = ENABLE

Bit	R/W	Reset	Description
12	RW	DISABLE	**SEQ_ACCESS_EN**: **0** = DISABLE **1** = ENABLE
8:0	RW	0x0	**RAM_ADDR**

MIXER_AHUBRAMCTL_GAIN_CONFIG_RAM_DATA_0

Offset: 0x430
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

MIXER_AHUBRAMCTL_PEAKM_RAM_CTRL_0

Offset: 0x434
Read/Write: See table below
Parity Protection: N
Reset: 0x00004000 (0b0xxx,xxxx,0000,0000,x100,xxx0,0000,0000)

Bit	R/W	Reset	Description
31	RO	DONE	**READ_BUSY**: **0** = DONE **1** = BUSY
23:16	RW	0x0	**SEQ_READ_COUNT**
14	RW	WRITE	**RW**: **0** = READ **1** = WRITE
13	RW	DISABLE	**ADDR_INIT_EN**: **0** = DISABLE **1** = ENABLE
12	RW	DISABLE	**SEQ_ACCESS_EN**: **0** = DISABLE **1** = ENABLE
8:0	RW	0x0	**RAM_ADDR**

MIXER_AHUBRAMCTL_PEAKM_RAM_DATA_0

Offset: 0x438
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

MIXER_CTRL_0

Offset: 0x43c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DISABLE	**BYPASS**: MIXER BYPASS mode. When enabled, five adders will all work at BYPASS mode. POP the first associated input stream and PUSH it out. **0** = DISABLE **1** = ENABLE

7.7.4.9 Master Volume Control (MVC) Registers

MVC_AXBAR_RX_STATUS_0

Offset: 0xc
Read/Write: RO
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01)

Bit	Reset	Description
1	FALSE	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	TRUE	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

MVC_AXBAR_RX_INT_STATUS_0

Offset: 0x10
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	CLEAR	**RX_DONE**: **0** = CLEAR **1** = SET

MVC_AXBAR_RX_INT_MASK_0

Offset: 0x14
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**RX_DONE**: **0** = UNMASK **1** = MASK

MVC_AXBAR_RX_INT_SET_0

Offset: 0x18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**RX_DONE**: This bit is auto cleared after an interrupt in generated **0** = FALSE **1** = TRUE

MVC_AXBAR_RX_INT_CLEAR_0

Offset: 0x1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**RX_DONE**: **0** = FALSE **1** = TRUE

MVC_AXBAR_RX_CIF_CTRL_0

Offset: 0x20
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0bxx00,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8
19:16	RO	0x0	**CLIENT_CHANNELS**: RO, equal to RX_CIF_CTRL AXBAR_CHANNELS **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32

Bit	R/W	Reset	Description
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

MVC_AXBAR_TX_STATUS_0

Offset: 0x4c
Read/Write: RO
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01)

Bit	Reset	Description
1	FALSE	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	TRUE	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

MVC_AXBAR_TX_INT_STATUS_0

Offset: 0x50
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	CLEAR	**TX_DONE**: **0** = CLEAR **1** = SET

MVC_AXBAR_TX_INT_MASK_0

Offset: 0x54
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**TX_DONE**: **0** = UNMASK **1** = MASK

MVC_AXBAR_TX_INT_SET_0

Offset: 0x58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**TX_DONE**: This bit is auto cleared after an interrupt in generated. **0** = FALSE **1** = TRUE

MVC_AXBAR_TX_INT_CLEAR_0

Offset: 0x5c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**TX_DONE**: **0** = FALSE **1** = TRUE

MVC_AXBAR_TX_CIF_CTRL_0

Offset: 0x60
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0bxx00,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RO	0x0	**AXBAR_CHANNELS**: RO, equal to RX_CIF_CTRL AXBAR_CHANNELS **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8
19:16	RO	0x0	**CLIENT_CHANNELS**: RO, equal to RX_CIF_CTRL AXBAR_CHANNELS **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP

Bit	R/W	Reset	Description
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

MVC_ENABLE_0

Offset: 0x80
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**ENABLE**: When disabled, MVC does not disable till all the data in the datapipe has been processed and sent out Read back of the register will also give module enable status **0** = FALSE **1** = TRUE

MVC_SOFT_RESET_0

Offset: 0x84
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**SOFT_RESET**: Soft Reset is a self-clearing bit. Soft Reset resets all FSM, flushes flow control FIFO and resets the state registers. It also bring the module back to disabled state (without flushing data in the pipe) **0** = FALSE **1** = TRUE

MVC_CG_0

Offset: 0x88
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	TRUE	**SLCG_EN**: Second level clock gating enable **0** = FALSE **1** = TRUE

MVC_STATUS_0

Offset: 0x90
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxx0,xxxx,xxx0)

Bit	Reset	Description
8	FALSE	**SLCG_CLKEN**: Status of SLCG **0** = FALSE **1** = TRUE
0	FALSE	**ENABLE_STATUS**: Status of MVC enabling. **0** = FALSE **1** = TRUE

MVC_INT_STATUS_0

Offset: 0x94
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	CLEAR	**TX_DONE**: When MVC is disabled, and whole pipe data (RXCIF, MVC CORE and TXCIF) is flushed, TX_DONE will be issued. **0** = CLEAR **1** = SET
0	CLEAR	**RX_DONE**: When MVC is disabled, RXCIF will stop receiving data from AXBAR, RX_DONE will be issue after all the data in TXCIF FIFO is popped out **0** = CLEAR **1** = SET

MVC_CTRL_0

Offset: 0xa8
Read/Write: RW
Parity Protection: N
Reset: 0x40000001 (0b01xx,xxxx,xxxx,xxxx,0000,0000,xxxx,xx01)

Bit	Reset	Description
15:8	0x0	**MUTE_UNMUTE**: Mute UnMute control bit. Eight bits for eight channels. bit[8]->ch0, bit[9]->ch1 ... bit[15]->ch7 If PER_CHAN_CTRL_EN is DISABLE, all the channels use the same MUTE_UNMUTE control: ch0->MUTE_UNMUTE[0] If Software program this register, must be followed by VOLUME_SWITCH trigger to let Hardware know new volume parameters ready. Hardware will take the new target volume (mute value or previous unmute target volume) after finishing current frame If Software programs new target volume when system is under MUTE, this bit will be set to UNMUTE by Hardware. Usually, this case is not allowed: Software programming both MUTE and new target volume at the same time. But if it is by accident, system will go to MUTE. **0** = UNMUTE **255** = MUTE
1	0x0	**CURVE_TYPE**: curve type, 0 for POLYNOMIAL Curve, 1 for LINEAR RAMP curve Curve type switching cannot do on the fly Software reset or disable/enable is needed to switch between two curves. **0** = POLYNOMIAL **1** = LINEAR_RAMP
0	UNBIASED	**ROUNDING_TYPE**: Rounding option, 1 for unbiased rounding (RNE), 0 for biased rounding **0** = BIASED **1** = UNBIASED

MVC_SWITCH_0

Offset: 0xac
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	CLEAR	**VOLUME_SWITCH**: Software should use a write to this register - to prompt the Hardware to start using a new set of volume for the volume related parameters (TARGET_VOL AND MUTE_UNMUTE) 1 bits for eight channels. Hardware will detect which channel has new target and do volume ramping After Software programming new TARGET_VOL or changing MUTE_UNMUTE status, Software needs to program this field to let Hardware know new volume is ready for using. Hardware does this switch as soon as its done processing the current frame This bit will remain 1(SET) till the switch hasn't happened, and is cleared to 0 by Hardware as an ack that the configuration has been switched to the new one. During Write - The field does not matter, only a Write Access to this field to generate a trigger pulse During Read - The field indicates whether polynomial coefficients switch clearing has happened or not **0** = CLEAR **1** = SET
1	CLEAR	**COEFF_SWITCH**: Software should use a write to this register - to prompt the Hardware to start using a new set of configuration for the polynomial coefficients After Software programming new polynomial coefficients, Software needs to program this field to let Hardware know new volume is ready for using. Hardware does this switch as soon as its done processing the current frame This bit will remain 1(SET) till the switch hasn't happened, and is cleared to 0 by Hardware as an ack that the configuration has been switched to the new one During Write - The field does not matter, only a Write Access to this field to generate a trigger pulse During Read - The field indicates whether polynomial coefficients switch clearing has happened or not there is interdependent between this and duration switch, which means Software cannot issue this switch until both coefficient switch and duration switch are cleared **0** = CLEAR **1** = SET

Bit	Reset	Description
0	CLEAR	**DURATION_SWITCH**: Software should use a write to this register - to prompt the Hardware to start using a new set of configuration for the duration related parameters (DURATION, DURATION_INV, POLY_N1 AND POLY_N2) After Software programming new duration parameters, Software needs to program this field to let Hardware know new volume is ready for using. Hardware does this switch as soon as its done processing the current frame This bit will remain 1(SET) until the switch hasn't happened, and is cleared to 0 by Hardware as an ack that the configuration has been switched to the new one During Write - The field does not matter, only a Write Access to this field to generate a trigger pulse During Read - The field indicates whether duration parameters switch clearing has happened or not The purpose of putting these three switch bits in one register is for the case of Software wants to do switching for all the three parameters (volume, duration, and coefficients) at the same time. Software just programs the three bits and write this register, Hardware will take the new set of volume, duration, and coefficient simultaneously when proceed new frame samples. There is interdependent between this and coefficient switch, which means Software cannot issue this switch until both coefficient switch and duration switch are cleared **0** = CLEAR **1** = SET

MVC_INIT_VOL_0

Register array of 8 entries.
Offset: 0xb0,..,0xcc
Read/Write: RW
Parity Protection: N
Reset: 0x00800000 (0b0000,0000,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x800000	**INIT_VOL**: Eight initial volumes for eight channels. If PER_CHAN_CTRL_EN is DISABLE, all the channels use the same initial volume: ch0->INIT_VOL_0 This register is shared for Volume Initialization for two curve types: Polynomial: default curve type, all 32 bits of the registers are used. The volume is provided in linear and should be in signed Q8.24 format. default: 0x80_0000 (0.5, Q8.24) range : 0x0 ~ 0x6400_0000 (0 ~ 100 @Q8.24) positive value only, cannot be set with negative number Linear Ramp: For this curve type, only bottom 16 bits are used. The volume is provided in dB and should be in signed Q8.8 format. default: 0xffff_fa00 (-6dB @Q8.8) range : 0xffff_8800 ~ 0x2800 (-120dB ~ 40dB @Q8.8)

MVC_TARGET_VOL_0

Register array of 8 entries.
Offset: 0xd0,..,0xec
Read/Write: RW
Parity Protection: N
Reset: 0x00800000 (0b0000,0000,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x800000	**TARGET_VOL**: Eight target volumes for eight channels. If PER_CHAN_CTRL_EN is DISABLE, all the channels use the same target volume: ch0->TARGET_VOL_0 This register is shared for Target Volume for two curve types: Polynomial: For this curve type, all 32 bits of the registers are used. The volume is provided in linear and should be in signed 8.24 format default: 0x80_0000 (0.5, Q8.24) range : 0x0 ~ 0x6400_0000 (0 ~ 100 @Q8.24) positive value only, cannot be set with negative number Linear Ramp: For this curve type, only bottom 16 bits are used. The volume is provided in dB and should be in signed 8.8 format default: 0xffff_fa00 (-6dB @Q8.8) range : 0xffff_8800 ~ 0x2800 (-120dB ~ 40dB @Q8.8) If Software program this register, must be followed by VOLUME_SWITCH trigger to let Hardware know new volume parameters ready. Hardware will take the new target volume after finishing current frame Usually this case is not allowed: Software programming both MUTE and new target volume at the same time. But if it is by accident, system will go to MUTE, ignore the target volume set it with RWTO, which Hardware can update this reg to be MUTE value when Software programming mute flag

MVC_DURATION_0

Offset: 0xf0
Read/Write: RW
Parity Protection: N
Reset: 0x000012c0 (0bxxxx,xxxx,0000,0000,0001,0010,1100,0000)

Bit	Reset	Description
23:0	0x12c0	**DURATION**: Number of samples to reach the target volume default: 0x12C0 (100ms @ 48 kHz) If Software program this register, must be followed by DURATION_SWITCH trigger to let Hardware know new duration parameters ready. Hardware will take the new duration parameter after finishing current frame

MVC_DURATION_INV_0

Offset: 0xf4
Read/Write: RW
Parity Protection: N
Reset: 0x0006d3a0 (0b0000,0000,0000,0110,1101,0011,1010,0000)

Bit	Reset	Description
31:0	0x6d3a0	**DURATION_INV**: inverse ramp duration in samples If Software program this register, must be followed by DURATION_SWITCH trigger to let Hardware know new duration parameters ready. Hardware will take the new inv_duration parameter after finishing current frame

MVC_POLY_N1_0

Offset: 0xf8
Read/Write: RW
Parity Protection: N
Reset: 0x0000007d (0bxxxx,xxxx,0000,0000,0000,0000,0111,1101)

Bit	Reset	Description
23:0	0x7d	**POLY_N1**: The first splitting points for the three segments POLY_N1, POLY_N2 and DURATION programming should meet POLY_N1 <= POLY_N2 <=DURATION For two segments case: POLY_N2 == DURATION. Programming coefficients for the first two segments: p1_0/1/2 and p2_0/1/2 For one segment case: POLY_N1 == POLY_N2 == DURATION. Programming coefficients for the first segment: p1_0/1/2 If SW program this register, must be followed by DURATION_SWITCH trigger to let Hardware know new duration parameters ready. Hardware will take the new poly_n1 parameter after finishing current frame Note: POLY_N1 cannot be changed without DURATION changing.

MVC_POLY_N2_0

Offset: 0xfc
Read/Write: RW
Parity Protection: N
Reset: 0x00000271 (0bxxxx,xxxx,0000,0000,0000,0010,0111,0001)

Bit	Reset	Description
23:0	0x271	**POLY_N2**: The second splitting points for the three segments POLY_N1, POLY_N2 and DURATION programming should meet POLY_N1 <= POLY_N2 <=DURATION For two segments case: POLY_N2 == DURATION. Programming coefficients for the first two segments: p1_0/1/2 and p2_0/1/2 For one segment case: POLY_N1 == POLY_N2 == DURATION. Programming coefficients for the first segment: p1_0/1/2 If SW program this register, must be followed by DURATION_SWITCH trigger to let Hardware know new duration parameters ready. Hardware will take the new poly_n2 parameter after finishing current frame Note: POLY_N2 cannot be changed without DURATION changing.

MVC_PEAK_CTRL_0

Offset: 0x100
Read/Write: RW
Parity Protection: N
Reset: 0x000012c0 (0b0xxx,xxxx,0000,0000,0001,0010,1100,0000)

Bit	Reset	Description
31	0x0	**PEAK_TYPE**: (default:0 ->window base): peak metering type: 0 for window based, 1 for reset-on-read **0** = WINPEAK **1** = RORPEAK
23:0	0x12c0	**PEAK_WIN**: It is for Peak Metering window size, as no of sample Only available when peak metering type is window based peak metering default:0x12C0 (100ms @ 48 kHz)

MVC_AHUBRAMCTL_CONFIG_RAM_CTRL_0

Offset: 0x104
Read/Write: See table below
Parity Protection: N
Reset: 0x00004000 (0b0xxx,xxxx,0000,0000,x100,xxx0,0000,0000)

Bit	R/W	Reset	Description
31	RO	DONE	**READ_BUSY**: **0** = DONE **1** = BUSY
23:16	RW	0x0	**SEQ_READ_COUNT**

Bit	R/W	Reset	Description
14	RW	WRITE	**RW**: **0** = READ **1** = WRITE
13	RW	DISABLE	**ADDR_INIT_EN**: **0** = DISABLE **1** = ENABLE
12	RW	DISABLE	**SEQ_ACCESS_EN**: **0** = DISABLE **1** = ENABLE
8:0	RW	0x0	**RAM_ADDR**

MVC_AHUBRAMCTL_CONFIG_RAM_DATA_0

Offset: 0x108
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

MVC_PEAK_VALUE_0

This is an array of eight identical register entries; the register fields below apply to each entry.
Offset: 0x10c..0x128
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**PEAK_VALUE**: peak value for sw read, 8 for 7.1 audio format

MVC_CONFIG_ERR_TYPE_0

Offset: 0x12c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	FALSE	**VOLUME_CONFIG_ERROR**: Two sources: 1. Software programming target volume during Switching is SET 2. Wrong programming: a. INIT_VOL > max volume, or INIT_VOL < min volume (refer to range of INIT_VOL) b. TARGET_VOL > max volume, or TARGET_VOL < min volume (refer to range of TARGET_VOL) **0** = FALSE **1** = TRUE
1	FALSE	**COEFF_CONFIG_ERROR**: Software programming coefficient buffer during Switching is SET **0** = FALSE **1** = TRUE
0	FALSE	**DURATION_CONFIG_ERROR**: Two sources: 1. Software programming duration parameters(DURATION, DURATION_INV, POLY_N1 and POLY_N2) during Switching is SET 2. Wrong programming: a. POLY_N1 > POLY_N2 or POLY_N1 > DURATION or POLY_N2 > DURATION b. BOTH DURATION and DURATION_INV should be changed simultaneously. Only one of them changed will result in error config **0** = FALSE **1** = TRUE

7.7.4.10 Arbitrary Sample Rate Control (ASRC) Registers

ASRC_STREAM1_CONFIG_0

Offset: 0x0
Read/Write: RW
Parity Protection: N
Reset: 0x00000010 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,0001,xxx0)

Bit	Reset	Description
31	0x0	**ENABLE_HW_RATIO_COMP**
7:4	0x1	**LANE_ID**
0	0x0	**RATIO_TYPE**: **0** = FROM_ARAD **1** = FROM_SW

ASRC_STREAM1_RATIO_INTEGER_PART_0

Offset: 0x4
Read/Write: RW

Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0001)

Bit	Reset	Description
4:0	0x1	**RATIO**

ASRC_STREAM1_RATIO_FRAC_PART_0

Offset: 0x8
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**RATIO**

ASRC_STREAM1_RATIO_LOCK_STATUS_0

Offset: 0xc
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**LOCK**: **0** = UNLOCK **1** = LOCK

ASRC_STREAM1_MUTE_UNMUTE_DURATION_0

Offset: 0x10
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000400 (0bxxxx,xxxx,xxxx,xxxx,0000,0100,0000,0000)

Bit	Reset	Description
15:0	0x400	**DURATION**

ASRC_STREAM1_TX_THRESHOLD_0

Offset: 0x14
Read/Write: RW
Parity Protection: N
Reset: 0x00201002 (0bxxxx,xxxx,0010,xxxx,0001,xxxx,xxxx,xx10)

Bit	Reset	Description
23:20	0x2	**NORMAL_WMARK**
15:12	0x1	**LOWER_WMARK**
1:0	0x2	**WORD_CNT**

ASRC_STREAM1_RX_THRESHOLD_0

Offset: 0x18
Read/Write: RW
Parity Protection: N
Reset: 0x00406002 (0bxxxx,xxxx,0100,xxxx,0110,xxxx,xxxx,xx10)

Bit	Reset	Description
23:20	0x4	**NORMAL_WMARK**
15:12	0x6	**UPPER_WMARK**
1:0	0x2	**WORD_CNT**

ASRC_STREAM1_RATIO_COMP_0

Offset: 0x1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**COMP_VALUE**

ASRC_STREAM1_RX_STATUS_0

Offset: 0x20
Read/Write: RO
Parity Protection: N
Reset: 0x0000000d (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,1101)

Bit	Reset	Description
1	FALSE	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	TRUE	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

ASRC_STREAM1_RX_CIF_CTRL_0

Offset: 0x24
Read/Write: See table below
Parity Protection: N
Reset: 0x00007500 (0b0000,0000,0000,0000,x111,x101,0000,x000)

Bit	R/W	Reset	Description
31	RO	0x0	**PACK8_ENABLE**
30	RO	0x0	**PACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RO	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x5	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RO	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RO	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
2	RO	0x0	**REPLICATE**: **0** = FALSE **1** = TRUE
1	RO	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RO	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ASRC_STREAM1_TX_STATUS_0

Offset: 0x2c
Read/Write: RO
Parity Protection: N
Reset: 0x0000000d (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,1101)

Bit	Reset	Description
1	FALSE	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	TRUE	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

ASRC_STREAM1_TX_CIF_CTRL_0

Offset: 0x30
Read/Write: See table below
Parity Protection: N
Reset: 0x00007500 (0b0000,0000,0000,0000,x111,x101,0000,x000)

Bit	R/W	Reset	Description
31	RO	0x0	**UNPACK8_ENABLE**
30	RO	0x0	**UNPACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RO	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RO	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x5	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RO	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RO	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
2	RO	0x0	**REPLICATE**: **0** = FALSE **1** = TRUE
1	RO	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RO	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ASRC_STREAM1_ENABLE_0

Offset: 0x38
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ASRC_STREAM1_SOFT_RESET_0

Offset: 0x3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ASRC_STREAM1_STATUS_0

Offset: 0x4c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
31	FALSE	**CONFIG_ERROR**: **0** = FALSE **1** = TRUE
0	FALSE	**ENABLE_STATUS**: **0** = FALSE **1** = TRUE

ASRC_STREAM1_STATEBUF_ADDR_0

Offset: 0x5c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ADDR**

ASRC_STREAM1_STATEBUF_CONFIG_0

Offset: 0x60
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000445 (0bxxxx,xxxx,xxxx,xxxx,0000,0100,0100,0101)

Bit	Reset	Description
15:0	0x445	**SIZE**

ASRC_STREAM1_INSAMPLEBUF_ADDR_0

Offset: 0x64
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ADDR**

ASRC_STREAM1_INSAMPLEBUF_CONFIG_0

Offset: 0x68
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000064 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0110,0100)

Bit	Reset	Description
15:0	0x64	**SIZE**

ASRC_STREAM1_OUTSAMPLEBUF_ADDR_0

Offset: 0x6c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000004b0 (0b0000,0000,0000,0000,0000,0100,1011,0000)

Bit	Reset	Description
31:0	0x4b0	**ADDR**

ASRC_STREAM1_OUTSAMPLEBUF_CONFIG_0

Offset: 0x70
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000064 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0110,0100)

Bit	Reset	Description
15:0	0x64	**SIZE**

ASRC_STREAM2_CONFIG_0

Offset: 0x80
Read/Write: RW
Parity Protection: N
Reset: 0x00000020 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,0010,xxx0)

Bit	Reset	Description
31	0x0	**ENABLE_HW_RATIO_COMP**
7:4	0x2	**LANE_ID**
0	0x0	**RATIO_TYPE**: **0** = FROM_ARAD **1** = FROM_SW

ASRC_STREAM2_RATIO_INTEGER_PART_0

Offset: 0x84
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0001)

Bit	Reset	Description
4:0	0x1	**RATIO**

ASRC_STREAM2_RATIO_FRAC_PART_0

Offset: 0x88
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**RATIO**

ASRC_STREAM2_RATIO_LOCK_STATUS_0

Offset: 0x8c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**LOCK**: **0** = UNLOCK **1** = LOCK

ASRC_STREAM2_TX_THRESHOLD_0

Offset: 0x94
Read/Write: RW
Parity Protection: N
Reset: 0x00201002 (0bxxxx,xxxx,0010,xxxx,0001,xxxx,xxxx,xx10)

Bit	Reset	Description
23:20	0x2	**NORMAL_WMARK**
15:12	0x1	**LOWER_WMARK**
1:0	0x2	**WORD_CNT**

ASRC_STREAM2_RX_THRESHOLD_0

Offset: 0x98
Read/Write: RW
Parity Protection: N
Reset: 0x00406002 (0bxxxx,xxxx,0100,xxxx,0110,xxxx,xxxx,xx10)

Bit	Reset	Description
23:20	0x4	**NORMAL_WMARK**
15:12	0x6	**UPPER_WMARK**
1:0	0x2	**WORD_CNT**

ASRC_STREAM2_RATIO_COMP_0

Offset: 0x9c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**COMP_VALUE**

ASRC_STREAM2_RX_STATUS_0

Offset: 0xa0
Read/Write: RO
Parity Protection: N
Reset: 0x0000000d (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,1101)

Bit	Reset	Description
1	FALSE	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	TRUE	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

ASRC_STREAM2_RX_CIF_CTRL_0

Offset: 0xa4
Read/Write: See table below
Parity Protection: N
Reset: 0x00007500 (0b0000,0000,0000,0000,x111,x101,0000,x000)

Bit	R/W	Reset	Description
31	RO	0x0	**PACK8_ENABLE**
30	RO	0x0	**PACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**

Bit	R/W	Reset	Description
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RO	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x5	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RO	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD

Bit	R/W	Reset	Description
5:4	RO	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
2	RO	0x0	**REPLICATE**: **0** = FALSE **1** = TRUE
1	RO	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RO	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ASRC_STREAM2_TX_STATUS_0

Offset: 0xac
Read/Write: RO
Parity Protection: N
Reset: 0x0000000d (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,1101)

Bit	Reset	Description
1	FALSE	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	TRUE	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

ASRC_STREAM2_TX_CIF_CTRL_0

Offset: 0xb0
Read/Write: See table below
Parity Protection: N
Reset: 0x00007500 (0b0000,0000,0000,0000,x111,x101,0000,x000)

Bit	R/W	Reset	Description
31	RO	0x0	**UNPACK8_ENABLE**
30	RO	0x0	**UNPACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**

Bit	R/W	Reset	Description
23:20	RO	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RO	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x5	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RO	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD

Bit	R/W	Reset	Description
5:4	RO	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
2	RO	0x0	**REPLICATE**: **0** = FALSE **1** = TRUE
1	RO	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RO	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ASRC_STREAM2_ENABLE_0

Offset: 0xb8
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ASRC_STREAM2_SOFT_RESET_0

Offset: 0xbc
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ASRC_STREAM2_STATUS_0

Offset: 0xcc
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
31	FALSE	**CONFIG_ERROR**: **0** = FALSE **1** = TRUE
0	FALSE	**ENABLE_STATUS**: **0** = FALSE **1** = TRUE

ASRC_STREAM3_CONFIG_0

Offset: 0x100
Read/Write: RW
Parity Protection: N
Reset: 0x00000030 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,0011,xxx0)

Bit	Reset	Description
31	0x0	**ENABLE_HW_RATIO_COMP**
7:4	0x3	**LANE_ID**
0	0x0	**RATIO_TYPE**: **0** = FROM_ARAD **1** = FROM_SW

ASRC_STREAM3_RATIO_INTEGER_PART_0

Offset: 0x104
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0001)

Bit	Reset	Description
4:0	0x1	**RATIO**

ASRC_STREAM3_RATIO_FRAC_PART_0

Offset: 0x108
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**RATIO**

ASRC_STREAM3_RATIO_LOCK_STATUS_0

Offset: 0x10c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**LOCK**: **0** = UNLOCK **1** = LOCK

ASRC_STREAM3_TX_THRESHOLD_0

Offset: 0x114
Read/Write: RW
Parity Protection: N
Reset: 0x00201002 (0bxxxx,xxxx,0010,xxxx,0001,xxxx,xxxx,xx10)

Bit	Reset	Description
23:20	0x2	**NORMAL_WMARK**
15:12	0x1	**LOWER_WMARK**
1:0	0x2	**WORD_CNT**

ASRC_STREAM3_RX_THRESHOLD_0

Offset: 0x118
Read/Write: RW
Parity Protection: N
Reset: 0x00406002 (0bxxxx,xxxx,0100,xxxx,0110,xxxx,xxxx,xx10)

Bit	Reset	Description
23:20	0x4	**NORMAL_WMARK**
15:12	0x6	**UPPER_WMARK**
1:0	0x2	**WORD_CNT**

ASRC_STREAM3_RATIO_COMP_0

Offset: 0x11c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**COMP_VALUE**

ASRC_STREAM3_RX_STATUS_0

Offset: 0x120
Read/Write: RO
Parity Protection: N
Reset: 0x0000000d (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,1101)

Bit	Reset	Description
1	FALSE	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	TRUE	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

ASRC_STREAM3_RX_CIF_CTRL_0

Offset: 0x124
Read/Write: See table below
Parity Protection: N
Reset: 0x00007500 (0b0000,0000,0000,0000,x111,x101,0000,x000)

Bit	R/W	Reset	Description
31	RO	0x0	**PACK8_ENABLE**
30	RO	0x0	**PACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**

Bit	R/W	Reset	Description
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RO	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x5	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RO	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD

Bit	R/W	Reset	Description
5:4	RO	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
2	RO	0x0	**REPLICATE**: **0** = FALSE **1** = TRUE
1	RO	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RO	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ASRC_STREAM3_TX_STATUS_0

Offset: 0x12c
Read/Write: RO
Parity Protection: N
Reset: 0x0000000d (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,1101)

Bit	Reset	Description
1	FALSE	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	TRUE	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

ASRC_STREAM3_TX_CIF_CTRL_0

Offset: 0x130
Read/Write: See table below
Parity Protection: N
Reset: 0x00007500 (0b0000,0000,0000,0000,x111,x101,0000,x000)

Bit	R/W	Reset	Description
31	RO	0x0	**UNPACK8_ENABLE**
30	RO	0x0	**UNPACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**

Bit	R/W	Reset	Description
23:20	RO	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RO	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x5	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RO	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD

Bit	R/W	Reset	Description
5:4	RO	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
2	RO	0x0	**REPLICATE**: **0** = FALSE **1** = TRUE
1	RO	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RO	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ASRC_STREAM3_ENABLE_0

Offset: 0x138
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ASRC_STREAM3_SOFT_RESET_0

Offset: 0x13c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ASRC_STREAM3_STATUS_0

Offset: 0x14c
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
31	FALSE	**CONFIG_ERROR**: **0** = FALSE **1** = TRUE
0	FALSE	**ENABLE_STATUS**: **0** = FALSE **1** = TRUE

ASRC_STREAM4_CONFIG_0

Offset: 0x180
Read/Write: RW
Parity Protection: N
Reset: 0x00000040 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,0100,xxx0)

Bit	Reset	Description
31	0x0	**ENABLE_HW_RATIO_COMP**
7:4	0x4	**LANE_ID**
0	0x0	**RATIO_TYPE**: **0** = FROM_ARAD **1** = FROM_SW

ASRC_STREAM4_RATIO_INTEGER_PART_0

Offset: 0x184
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0001)

Bit	Reset	Description
4:0	0x1	**RATIO**

ASRC_STREAM4_RATIO_FRAC_PART_0

Offset: 0x188
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**RATIO**

ASRC_STREAM4_RATIO_LOCK_STATUS_0

Offset: 0x18c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**LOCK**: **0** = UNLOCK **1** = LOCK

ASRC_STREAM4_TX_THRESHOLD_0

Offset: 0x194
Read/Write: RW
Parity Protection: N
Reset: 0x00201002 (0bxxxx,xxxx,0010,xxxx,0001,xxxx,xxxx,xx10)

Bit	Reset	Description
23:20	0x2	**NORMAL_WMARK**
15:12	0x1	**LOWER_WMARK**
1:0	0x2	**WORD_CNT**

ASRC_STREAM4_RX_THRESHOLD_0

Offset: 0x198
Read/Write: RW
Parity Protection: N
Reset: 0x00406002 (0bxxxx,xxxx,0100,xxxx,0110,xxxx,xxxx,xx10)

Bit	Reset	Description
23:20	0x4	**NORMAL_WMARK**
15:12	0x6	**UPPER_WMARK**
1:0	0x2	**WORD_CNT**

ASRC_STREAM4_RATIO_COMP_0

Offset: 0x19c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**COMP_VALUE**

ASRC_STREAM4_RX_STATUS_0

Offset: 0x1a0
Read/Write: RO
Parity Protection: N
Reset: 0x0000000d (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,1101)

Bit	Reset	Description
1	FALSE	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	TRUE	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

ASRC_STREAM4_RX_CIF_CTRL_0

Offset: 0x1a4
Read/Write: See table below
Parity Protection: N
Reset: 0x00007500 (0b0000,0000,0000,0000,x111,x101,0000,x000)

Bit	R/W	Reset	Description
31	RO	0x0	**PACK8_ENABLE**
30	RO	0x0	**PACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**

Bit	R/W	Reset	Description
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RO	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x5	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RO	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD

Bit	R/W	Reset	Description
5:4	RO	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
2	RO	0x0	**REPLICATE**: **0** = FALSE **1** = TRUE
1	RO	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RO	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ASRC_STREAM4_TX_STATUS_0

Offset: 0x1ac
Read/Write: RO
Parity Protection: N
Reset: 0x0000000d (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,1101)

Bit	Reset	Description
1	FALSE	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	TRUE	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

ASRC_STREAM4_TX_CIF_CTRL_0

Offset: 0x1b0
Read/Write: See table below
Parity Protection: N
Reset: 0x00007500 (0b0000,0000,0000,0000,x111,x101,0000,x000)

Bit	R/W	Reset	Description
31	RO	0x0	**UNPACK8_ENABLE**
30	RO	0x0	**UNPACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**

Bit	R/W	Reset	Description
23:20	RO	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RO	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x5	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RO	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD

Bit	R/W	Reset	Description
5:4	RO	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
2	RO	0x0	**REPLICATE**: **0** = FALSE **1** = TRUE
1	RO	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RO	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ASRC_STREAM4_ENABLE_0

Offset: 0x1b8
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ASRC_STREAM4_SOFT_RESET_0

Offset: 0x1bc
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ASRC_STREAM4_STATUS_0

Offset: 0x1cc
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
31	FALSE	**CONFIG_ERROR**: **0** = FALSE **1** = TRUE
0	FALSE	**ENABLE_STATUS**: **0** = FALSE **1** = TRUE

ASRC_STREAM5_CONFIG_0

Offset: 0x200
Read/Write: RW
Parity Protection: N
Reset: 0x00000050 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,0101,xxx0)

Bit	Reset	Description
31	0x0	**ENABLE_HW_RATIO_COMP**
7:4	0x5	**LANE_ID**
0	0x0	**RATIO_TYPE**: **0** = FROM_ARAD **1** = FROM_SW

ASRC_STREAM5_RATIO_INTEGER_PART_0

Offset: 0x204
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0001)

Bit	Reset	Description
4:0	0x1	**RATIO**

ASRC_STREAM5_RATIO_FRAC_PART_0

Offset: 0x208
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**RATIO**

ASRC_STREAM5_RATIO_LOCK_STATUS_0

Offset: 0x20c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**LOCK**: **0** = UNLOCK **1** = LOCK

ASRC_STREAM5_TX_THRESHOLD_0

Offset: 0x214
Read/Write: RW
Parity Protection: N
Reset: 0x00201002 (0bxxxx,xxxx,0010,xxxx,0001,xxxx,xxxx,xx10)

Bit	Reset	Description
23:20	0x2	**NORMAL_WMARK**
15:12	0x1	**LOWER_WMARK**
1:0	0x2	**WORD_CNT**

ASRC_STREAM5_RX_THRESHOLD_0

Offset: 0x218
Read/Write: RW
Parity Protection: N
Reset: 0x00406002 (0bxxxx,xxxx,0100,xxxx,0110,xxxx,xxxx,xx10)

Bit	Reset	Description
23:20	0x4	**NORMAL_WMARK**
15:12	0x6	**UPPER_WMARK**
1:0	0x2	**WORD_CNT**

ASRC_STREAM5_RATIO_COMP_0

Offset: 0x21c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**COMP_VALUE**

ASRC_STREAM5_RX_STATUS_0

Offset: 0x220
Read/Write: RO
Parity Protection: N
Reset: 0x0000000d (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,1101)

Bit	Reset	Description
1	FALSE	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	TRUE	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

ASRC_STREAM5_RX_CIF_CTRL_0

Offset: 0x224
Read/Write: See table below
Parity Protection: N
Reset: 0x00007500 (0b0000,0000,0000,0000,x111,x101,0000,x000)

Bit	R/W	Reset	Description
31	RO	0x0	**PACK8_ENABLE**
30	RO	0x0	**PACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**

Bit	R/W	Reset	Description
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RO	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x5	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RO	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD

Bit	R/W	Reset	Description
5:4	RO	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
2	RO	0x0	**REPLICATE**: **0** = FALSE **1** = TRUE
1	RO	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RO	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ASRC_STREAM5_TX_STATUS_0

Offset: 0x22c
Read/Write: RO
Parity Protection: N
Reset: 0x0000000d (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,1101)

Bit	Reset	Description
1	FALSE	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	TRUE	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

ASRC_STREAM5_TX_CIF_CTRL_0

Offset: 0x230
Read/Write: See table below
Parity Protection: N
Reset: 0x00007500 (0b0000,0000,0000,0000,x111,x101,0000,x000)

Bit	R/W	Reset	Description
31	RO	0x0	**UNPACK8_ENABLE**
30	RO	0x0	**UNPACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**

Bit	R/W	Reset	Description
23:20	RO	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RO	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x5	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RO	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD

Bit	R/W	Reset	Description
5:4	RO	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
2	RO	0x0	**REPLICATE**: **0** = FALSE **1** = TRUE
1	RO	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RO	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ASRC_STREAM5_ENABLE_0

Offset: 0x238
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ASRC_STREAM5_SOFT_RESET_0

Offset: 0x23c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ASRC_STREAM5_STATUS_0

Offset: 0x24c
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
31	FALSE	**CONFIG_ERROR**: **0** = FALSE **1** = TRUE
0	FALSE	**ENABLE_STATUS**: **0** = FALSE **1** = TRUE

ASRC_STREAM6_CONFIG_0

Offset: 0x280
Read/Write: RW
Parity Protection: N
Reset: 0x00000060 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,0110,xxx0)

Bit	Reset	Description
31	0x0	**ENABLE_HW_RATIO_COMP**
7:4	0x6	**LANE_ID**
0	0x0	**RATIO_TYPE**: **0** = FROM_ARAD **1** = FROM_SW

ASRC_STREAM6_RATIO_INTEGER_PART_0

Offset: 0x284
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0001)

Bit	Reset	Description
4:0	0x1	**RATIO**

ASRC_STREAM6_RATIO_FRAC_PART_0

Offset: 0x288
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**RATIO**

ASRC_STREAM6_RATIO_LOCK_STATUS_0

Offset: 0x28c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**LOCK**: **0** = UNLOCK **1** = LOCK

ASRC_STREAM6_TX_THRESHOLD_0

Offset: 0x294
Read/Write: RW
Parity Protection: N
Reset: 0x00201002 (0bxxxx,xxxx,0010,xxxx,0001,xxxx,xxxx,xx10)

Bit	Reset	Description
23:20	0x2	**NORMAL_WMARK**
15:12	0x1	**LOWER_WMARK**
1:0	0x2	**WORD_CNT**

ASRC_STREAM6_RX_THRESHOLD_0

Offset: 0x298
Read/Write: RW
Parity Protection: N
Reset: 0x00406002 (0bxxxx,xxxx,0100,xxxx,0110,xxxx,xxxx,xx10)

Bit	Reset	Description
23:20	0x4	**NORMAL_WMARK**
15:12	0x6	**UPPER_WMARK**
1:0	0x2	**WORD_CNT**

ASRC_STREAM6_RATIO_COMP_0

Offset: 0x29c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**COMP_VALUE**

ASRC_STREAM6_RX_STATUS_0

Offset: 0x2a0
Read/Write: RO
Parity Protection: N
Reset: 0x0000000d (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,1101)

Bit	Reset	Description
1	FALSE	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	TRUE	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

ASRC_STREAM6_RX_CIF_CTRL_0

Offset: 0x2a4
Read/Write: See table below
Parity Protection: N
Reset: 0x00007500 (0b0000,0000,0000,0000,x111,x101,0000,x000)

Bit	R/W	Reset	Description
31	RO	0x0	**PACK8_ENABLE**
30	RO	0x0	**PACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**

Bit	R/W	Reset	Description
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RO	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x5	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RO	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD

Bit	R/W	Reset	Description
5:4	RO	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
2	RO	0x0	**REPLICATE**: **0** = FALSE **1** = TRUE
1	RO	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RO	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ASRC_STREAM6_TX_STATUS_0

Offset: 0x2ac
Read/Write: RO
Parity Protection: N
Reset: 0x0000000d (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,1101)

Bit	Reset	Description
1	FALSE	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	TRUE	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

ASRC_STREAM6_TX_CIF_CTRL_0

Offset: 0x2b0
Read/Write: See table below
Parity Protection: N
Reset: 0x00007500 (0b0000,0000,0000,0000,x111,x101,0000,x000)

Bit	R/W	Reset	Description
31	RO	0x0	**UNPACK8_ENABLE**
30	RO	0x0	**UNPACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**

Bit	R/W	Reset	Description
23:20	RO	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RO	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x5	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RO	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD

Bit	R/W	Reset	Description
5:4	RO	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
2	RO	0x0	**REPLICATE**: **0** = FALSE **1** = TRUE
1	RO	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RO	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ASRC_STREAM6_ENABLE_0

Offset: 0x2b8
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ASRC_STREAM6_SOFT_RESET_0

Offset: 0x2bc
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ASRC_STREAM6_STATUS_0

Offset: 0x2cc
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
31	FALSE	**CONFIG_ERROR**: **0** = FALSE **1** = TRUE
0	FALSE	**ENABLE_STATUS**: **0** = FALSE **1** = TRUE

ASRC_GLOBAL_ENB_0

Offset: 0x2f4
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: This bit must be set to "1" for enabling the processing of streams **0** = FALSE **1** = TRUE

ASRC_GLOBAL_SOFT_RESET_0

Offset: 0x2f8
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Self clearing soft reset **0** = FALSE **1** = TRUE

ASRC_GLOBAL_CG_0

Offset: 0x2fc
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	0x1	**SLCG_ENABLE**: Second level clock gating enable for ASRC **0** = FALSE **1** = TRUE

ASRC_GLOBAL_CONFIG_0

Offset: 0x300
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**RATIO_PREC_CNTRL**: **0** = FRAC_28BIT_PRECISION **1** = FRAC_32BIT_PRECISION

ASRC_GLOBAL_SCRATCH_ADDR_0

Offset: 0x304
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ADDR**

ASRC_GLOBAL_SCRATCH_CONFIG_0

SCRATCH_CONFIG register
SIZE: Max Size of ARAM in Bytes; RTL should not cross this Size
BLOCK_SIZE: ASRC processes "BLOCKSIZE" samples per channel before switching to another channel. low value. Normally the default value will cover all the worst case scenarios. Software may use BLOCK_SIZE field to change it.
MAX_CHANNELS: MAX_CHANNELS is used by ASRC hardware to partition the allocated buffer space between states and samples. By default, it is 12. Software may change it if it requires support for more channels from the hardware at the expense of guaranteed performance.

Offset: 0x308
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0c207980 (0b0000,1100,0010,0000,0111,1001,1000,0000)

Bit	R/W	Reset	Description
31:24	RW	0xc	**MAX_CHANNELS**
23:16	RW	0x20	**BLOCK_SIZE**
15:0	RO	0x7980	**SIZE**

ASRC_RATIO_UPD_RX_CIF_CTRL_0

Offset: 0x30c
Read/Write: See table below
Parity Protection: N
Reset: 0x00115500 (0b0000,0000,0001,0001,x101,x101,0000,x000)

Bit	R/W	Reset	Description
31	RO	0x0	**UNPACK8_ENABLE**
30	RO	0x0	**UNPACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x1	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
19:16	RO	0x1	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x5	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x5	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RO	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RO	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
2	RO	0x0	**REPLICATE**: **0** = FALSE **1** = TRUE
1	RO	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RO	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ASRC_RATIO_UPD_RX_STATUS_0

Offset: 0x310
Read/Write: RO
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01)

Bit	Reset	Description
1	FALSE	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	TRUE	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

ASRC_GLOBAL_STATUS_0

Offset: 0x314
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xx00,0000,0000,0000,xxx0)

Bit	Reset	Description
17	FALSE	**GLOBAL_ENABLED**: **0** = FALSE **1** = TRUE
7:4	0x0	**CURRENT_PROCESSING_STREAMID**
0	FALSE	**SLCG_CLKEN**: **0** = FALSE **1** = TRUE

ASRC_GLOBAL_STREAM_ENABLE_STATUS_0

Offset: 0x318
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**STREAM6_ENABLED**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
4	0x0	**STREAM5_ENABLED**: **0** = FALSE **1** = TRUE
3	0x0	**STREAM4_ENABLED**: **0** = FALSE **1** = TRUE
2	0x0	**STREAM3_ENABLED**: **0** = FALSE **1** = TRUE
1	0x0	**STREAM2_ENABLED**: **0** = FALSE **1** = TRUE
0	0x0	**STREAM1_ENABLED**: **0** = FALSE **1** = TRUE

ASRC_GLOBAL_INT_STATUS_0

Offset: 0x324
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xx00,0000,xxxx,xxx0)

Bit	Reset	Description
13	0x0	**STREAM6_RATIO_ERROR**: **0** = FALSE **1** = TRUE
12	0x0	**STREAM5_RATIO_ERROR**: **0** = FALSE **1** = TRUE
11	0x0	**STREAM4_RATIO_ERROR**: **0** = FALSE **1** = TRUE
10	0x0	**STREAM3_RATIO_ERROR**: **0** = FALSE **1** = TRUE
9	0x0	**STREAM2_RATIO_ERROR**: **0** = FALSE **1** = TRUE
8	0x0	**STREAM1_RATIO_ERROR**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ERROR**: **0** = FALSE **1** = TRUE

ASRC_GLOBAL_INT_MASK_0

Offset: 0x328
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xx00,0000,xxxx,xxx0)

Bit	Reset	Description
13	0x0	**STREAM6_RATIO_ERROR**: **0** = FALSE **1** = TRUE
12	0x0	**STREAM5_RATIO_ERROR**: **0** = FALSE **1** = TRUE
11	0x0	**STREAM4_RATIO_ERROR**: **0** = FALSE **1** = TRUE
10	0x0	**STREAM3_RATIO_ERROR**: **0** = FALSE **1** = TRUE
9	0x0	**STREAM2_RATIO_ERROR**: **0** = FALSE **1** = TRUE
8	0x0	**STREAM1_RATIO_ERROR**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ERROR**: **0** = FALSE **1** = TRUE

ASRC_GLOBAL_INT_SET_0

Offset: 0x32c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xx00,0000,xxxx,xxx0)

Bit	Reset	Description
13	0x0	**STREAM6_RATIO_ERROR**: **0** = FALSE **1** = TRUE
12	0x0	**STREAM5_RATIO_ERROR**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
11	0x0	**STREAM4_RATIO_ERROR**: **0** = FALSE **1** = TRUE
10	0x0	**STREAM3_RATIO_ERROR**: **0** = FALSE **1** = TRUE
9	0x0	**STREAM2_RATIO_ERROR**: **0** = FALSE **1** = TRUE
8	0x0	**STREAM1_RATIO_ERROR**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ERROR**: **0** = FALSE **1** = TRUE

ASRC_GLOBAL_INT_CLEAR_0

Offset: 0x330
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xx00,0000,xxxx,xxx0)

Bit	Reset	Description
13	0x0	**STREAM6_RATIO_ERROR**: **0** = FALSE **1** = TRUE
12	0x0	**STREAM5_RATIO_ERROR**: **0** = FALSE **1** = TRUE
11	0x0	**STREAM4_RATIO_ERROR**: **0** = FALSE **1** = TRUE
10	0x0	**STREAM3_RATIO_ERROR**: **0** = FALSE **1** = TRUE
9	0x0	**STREAM2_RATIO_ERROR**: **0** = FALSE **1** = TRUE
8	0x0	**STREAM1_RATIO_ERROR**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ERROR**: **0** = FALSE **1** = TRUE

ASRC_GLOBAL_TRANSFER_ERROR_LOG_0

Read/write response status and corresponding channel ID. An interrupt is generated on receiving an error response for read/write request made to ARAM/DRAM.

Offset: 0x334
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xx00,0000,xx00,xx00,0000,xx00)

Bit	Reset	Description
21:16	0x0	**RID**
13:12	0x0	**RRESP**: **0** = OKAY **1** = EXOKAY **2** = SLVERR **3** = DECERR
9:4	0x0	**BID**
1:0	0x0	**BRESP**: **0** = OKAY **1** = EXOKAY **2** = SLVERR **3** = DECERR

ASRC_GLOBAL_APR_CTRL_0

Offset: 0x1000
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,00x0)

Bit	Reset	Description
3:2	0x0	**AST_REGION_ID**
0	0x0	**APR_EN**

ASRC_GLOBAL_APR_CTRL_ACCESS_CTRL_0

Offset: 0x1004
Read/Write: RW

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**DIS_ACCESS**

ASRC_GLOBAL_DISARM_APR_0

Offset: 0x1008
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**DISARM_STREAM_6_APR**
4	0x0	**DISARM_STREAM_5_APR**
3	0x0	**DISARM_STREAM_4_APR**
2	0x0	**DISARM_STREAM_3_APR**
1	0x0	**DISARM_STREAM_2_APR**
0	0x0	**DISARM_STREAM_1_APR**

ASRC_GLOBAL_DISARM_APR_ACCESS_CTRL_0

Offset: 0x100c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**DIS_ACCESS**

ASRC_GLOBAL_RATIO_WR_ACCESS_0

Offset: 0x1010
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**STREAM_6_WR_DIS**
4	0x0	**STREAM_5_WR_DIS**
3	0x0	**STREAM_4_WR_DIS**
2	0x0	**STREAM_3_WR_DIS**
1	0x0	**STREAM_2_WR_DIS**
0	0x0	**STREAM_1_WR_DIS**

ASRC_GLOBAL_RATIO_WR_ACCESS_CTRL_0

Offset: 0x1014
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**DIS_ACCESS**

ASRC_CYA_0

This is an array of four identical register entries; the register fields below apply to each entry.
Full register list is: ASRC_CYA_<i>, among which <i> belongs to <0..3>.
Offset: 0x1018,..,0x1024
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CYA**

7.7.4.11 Audio Sampling Rate Detector (ARAD) Control Registers

ARAD_LANE_ENABLE_0

Offset: 0x0
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	FALSE	**LANE6_ENABLE**: **0** = FALSE **1** = TRUE
4	FALSE	**LANE5_ENABLE**: **0** = FALSE **1** = TRUE
3	FALSE	**LANE4_ENABLE**: **0** = FALSE **1** = TRUE
2	FALSE	**LANE3_ENABLE**: **0** = FALSE **1** = TRUE
1	FALSE	**LANE2_ENABLE**: **0** = FALSE **1** = TRUE
0	FALSE	**LANE1_ENABLE**: **0** = FALSE **1** = TRUE

ARAD_LANE_STATUS_0

Offset: 0x4
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xx00,0000,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
21	UNLOCKED	**LANE6_LOCK_STATE**: **0** = UNLOCKED **1** = LOCKED
20	UNLOCKED	**LANE5_LOCK_STATE**: **0** = UNLOCKED **1** = LOCKED
19	UNLOCKED	**LANE4_LOCK_STATE**: **0** = UNLOCKED **1** = LOCKED
18	UNLOCKED	**LANE3_LOCK_STATE**: **0** = UNLOCKED **1** = LOCKED
17	UNLOCKED	**LANE2_LOCK_STATE**: **0** = UNLOCKED **1** = LOCKED
16	UNLOCKED	**LANE1_LOCK_STATE**: **0** = UNLOCKED **1** = LOCKED

Bit	Reset	Description
5	FALSE	**LANE6_ENABLED**: **0** = FALSE **1** = TRUE
4	FALSE	**LANE5_ENABLED**: **0** = FALSE **1** = TRUE
3	FALSE	**LANE4_ENABLED**: **0** = FALSE **1** = TRUE
2	FALSE	**LANE3_ENABLED**: **0** = FALSE **1** = TRUE
1	FALSE	**LANE2_ENABLED**: **0** = FALSE **1** = TRUE
0	FALSE	**LANE1_ENABLED**: **0** = FALSE **1** = TRUE

ARAD_LANE_SOFT_RESET_0

Offset: 0x8
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	FALSE	**LANE6_SOFT_RESET**: **0** = FALSE **1** = TRUE
4	FALSE	**LANE5_SOFT_RESET**: **0** = FALSE **1** = TRUE
3	FALSE	**LANE4_SOFT_RESET**: **0** = FALSE **1** = TRUE
2	FALSE	**LANE3_SOFT_RESET**: **0** = FALSE **1** = TRUE
1	FALSE	**LANE2_SOFT_RESET**: **0** = FALSE **1** = TRUE
0	FALSE	**LANE1_SOFT_RESET**: **0** = FALSE **1** = TRUE

ARAD_LANE_INT_STATUS_0

Offset: 0xc
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xx00,0000,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
21	FALSE	**LANE6_RATIO_CHANGED**: **0** = FALSE **1** = TRUE
20	FALSE	**LANE5_RATIO_CHANGED**: **0** = FALSE **1** = TRUE
19	FALSE	**LANE4_RATIO_CHANGED**: **0** = FALSE **1** = TRUE
18	FALSE	**LANE3_RATIO_CHANGED**: **0** = FALSE **1** = TRUE
17	FALSE	**LANE2_RATIO_CHANGED**: **0** = FALSE **1** = TRUE
16	FALSE	**LANE1_RATIO_CHANGED**: **0** = FALSE **1** = TRUE
5	FALSE	**LANE6_UNLOCKED**: **0** = FALSE **1** = TRUE
4	FALSE	**LANE5_UNLOCKED**: **0** = FALSE **1** = TRUE
3	FALSE	**LANE4_UNLOCKED**: **0** = FALSE **1** = TRUE
2	FALSE	**LANE3_UNLOCKED**: **0** = FALSE **1** = TRUE
1	FALSE	**LANE2_UNLOCKED**: **0** = FALSE **1** = TRUE
0	FALSE	**LANE1_UNLOCKED**: **0** = FALSE **1** = TRUE

ARAD_LANE_INT_MASK_0

Offset: 0x10
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xx00,0000,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
21	FALSE	**LANE6_RATIO_CHANGED**: **0** = FALSE **1** = TRUE
20	FALSE	**LANE5_RATIO_CHANGED**: **0** = FALSE **1** = TRUE
19	FALSE	**LANE4_RATIO_CHANGED**: **0** = FALSE **1** = TRUE
18	FALSE	**LANE3_RATIO_CHANGED**: **0** = FALSE **1** = TRUE
17	FALSE	**LANE2_RATIO_CHANGED**: **0** = FALSE **1** = TRUE
16	FALSE	**LANE1_RATIO_CHANGED**: **0** = FALSE **1** = TRUE
5	FALSE	**LANE6_UNLOCKED**: **0** = FALSE **1** = TRUE
4	FALSE	**LANE5_UNLOCKED**: **0** = FALSE **1** = TRUE
3	FALSE	**LANE4_UNLOCKED**: **0** = FALSE **1** = TRUE
2	FALSE	**LANE3_UNLOCKED**: **0** = FALSE **1** = TRUE
1	FALSE	**LANE2_UNLOCKED**: **0** = FALSE **1** = TRUE
0	FALSE	**LANE1_UNLOCKED**: **0** = FALSE **1** = TRUE

ARAD_LANE_INT_SET_0

Offset: 0x14
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xx00,0000,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
21	FALSE	**LANE6_RATIO_CHANGED**: **0** = FALSE **1** = TRUE
20	FALSE	**LANE5_RATIO_CHANGED**: **0** = FALSE **1** = TRUE
19	FALSE	**LANE4_RATIO_CHANGED**: **0** = FALSE **1** = TRUE
18	FALSE	**LANE3_RATIO_CHANGED**: **0** = FALSE **1** = TRUE
17	FALSE	**LANE2_RATIO_CHANGED**: **0** = FALSE **1** = TRUE
16	FALSE	**LANE1_RATIO_CHANGED**: **0** = FALSE **1** = TRUE
5	FALSE	**LANE6_UNLOCKED**: **0** = FALSE **1** = TRUE
4	FALSE	**LANE5_UNLOCKED**: **0** = FALSE **1** = TRUE
3	FALSE	**LANE4_UNLOCKED**: **0** = FALSE **1** = TRUE
2	FALSE	**LANE3_UNLOCKED**: **0** = FALSE **1** = TRUE
1	FALSE	**LANE2_UNLOCKED**: **0** = FALSE **1** = TRUE
0	FALSE	**LANE1_UNLOCKED**: **0** = FALSE **1** = TRUE

ARAD_LANE_INT_CLEAR_0

Offset: 0x18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xx00,0000,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
21	FALSE	**LANE6_RATIO_CHANGED**: **0** = FALSE **1** = TRUE
20	FALSE	**LANE5_RATIO_CHANGED**: **0** = FALSE **1** = TRUE
19	FALSE	**LANE4_RATIO_CHANGED**: **0** = FALSE **1** = TRUE
18	FALSE	**LANE3_RATIO_CHANGED**: **0** = FALSE **1** = TRUE
17	FALSE	**LANE2_RATIO_CHANGED**: **0** = FALSE **1** = TRUE
16	FALSE	**LANE1_RATIO_CHANGED**: **0** = FALSE **1** = TRUE
5	FALSE	**LANE6_UNLOCKED**: **0** = FALSE **1** = TRUE
4	FALSE	**LANE5_UNLOCKED**: **0** = FALSE **1** = TRUE
3	FALSE	**LANE4_UNLOCKED**: **0** = FALSE **1** = TRUE
2	FALSE	**LANE3_UNLOCKED**: **0** = FALSE **1** = TRUE
1	FALSE	**LANE2_UNLOCKED**: **0** = FALSE **1** = TRUE
0	FALSE	**LANE1_UNLOCKED**: **0** = FALSE **1** = TRUE

ARAD_GLOBAL_SOFT_RESET_0

Offset: 0x1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**SOFTRESET**: **0** = FALSE **1** = TRUE

ARAD_CG_0

Offset: 0x20
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	TRUE	**SLCG_ENABLE**: **0** = FALSE **1** = TRUE

ARAD_STATUS_0

Offset: 0x24
Read/Write: RO
Parity Protection: N
Reset: 0x00000004 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x100)

Bit	Reset	Description
2	TRUE	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
1	FALSE	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	FALSE	**SLCG_ENABLED**: **0** = FALSE **1** = TRUE

ARAD_SEND_RATIO_0

Offset: 0x28
Read/Write: RW

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**SEND**: This bit is auto cleared after an interrupt in generated **0** = FALSE **1** = TRUE

ARAD_CYA_GLOBAL_0

Offset: 0x2c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CYA_BITS**

ARAD_LANE1_NUMERATOR_MUX_SEL_0

Offset: 0x40
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,0000,xxxx,xxxx,0000,xxxx,xx00,0000)

Bit	Reset	Description
27	FALSE	**IQC2_SEL**: **0** = FALSE **1** = TRUE
26	FALSE	**IQC1_SEL**: **0** = FALSE **1** = TRUE
25	FALSE	**DSPK2_SEL**: **0** = FALSE **1** = TRUE
24	FALSE	**DSPK1_SEL**: **0** = FALSE **1** = TRUE
15	FALSE	**DMIC4_SEL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
14	FALSE	**DMIC3_SEL**: **0** = FALSE **1** = TRUE
13	FALSE	**DMIC2_SEL**: **0** = FALSE **1** = TRUE
12	FALSE	**DMIC1_SEL**: **0** = FALSE **1** = TRUE
5	FALSE	**I2S6_SEL**: **0** = FALSE **1** = TRUE
4	FALSE	**I2S5_SEL**: **0** = FALSE **1** = TRUE
3	FALSE	**I2S4_SEL**: **0** = FALSE **1** = TRUE
2	FALSE	**I2S3_SEL**: **0** = FALSE **1** = TRUE
1	FALSE	**I2S2_SEL**: **0** = FALSE **1** = TRUE
0	FALSE	**I2S1_SEL**: **0** = FALSE **1** = TRUE

ARAD_LANE1_NUMERATOR_PRESCALAR_0

Offset: 0x44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PRESCALAR**

ARAD_LANE1_DENOMINATOR_MUX_SEL_0

Offset: 0x48
Read/Write: RW

Parity Protection: N
Reset: 0x00000000 (0bxxxx,0000,xxxx,xxxx,0000,xxxx,xx00,0000)

Bit	Reset	Description
27	FALSE	**IQC2_SEL**: **0** = FALSE **1** = TRUE
26	FALSE	**IQC1_SEL**: **0** = FALSE **1** = TRUE
25	0x0	**DSPK2_SEL**: **0** = FALSE **1** = TRUE
24	0x0	**DSPK1_SEL**: **0** = FALSE **1** = TRUE
15	0x0	**DMIC4_SEL**: **0** = FALSE **1** = TRUE
14	0x0	**DMIC3_SEL**: **0** = FALSE **1** = TRUE
13	0x0	**DMIC2_SEL**: **0** = FALSE **1** = TRUE
12	0x0	**DMIC1_SEL**: **0** = FALSE **1** = TRUE
5	0x0	**I2S6_SEL**: **0** = FALSE **1** = TRUE
4	0x0	**I2S5_SEL**: **0** = FALSE **1** = TRUE
3	0x0	**I2S4_SEL**: **0** = FALSE **1** = TRUE
2	0x0	**I2S3_SEL**: **0** = FALSE **1** = TRUE
1	0x0	**I2S2_SEL**: **0** = FALSE **1** = TRUE
0	0x0	**I2S1_SEL**: **0** = FALSE **1** = TRUE

ARAD_LANE1_DENOMINATOR_PRESCALAR_0

Offset: 0x4c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PRESCALAR**

ARAD_LANE1_RATIO_INTEGER_PART_0

Offset: 0x50
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VALUE**

ARAD_LANE1_RATIO_FRACTIONAL_PART_0

Offset: 0x54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VALUE**

ARAD_LANE1_PERIOD_COUNT_0

Offset: 0x58
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**DENOMINATOR**

Bit	Reset	Description
15:0	0x0	**NUMERATOR**

ARAD_LANE1_SERVO_LOOP_CONFIG_0

Offset: 0x5c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000a5b7 (0bxxxx,xxxx,xxxx,xxxx,1010,0101,1011,0111)

Bit	Reset	Description
15:12	0xa	**K2_SHIFT_VALUE_SLOW_MODE**
11:8	0x5	**K2_SHIFT_VALUE_FAST_MODE**
7:4	0xb	**K1_SHIFT_VALUE_SLOW_MODE**
3:0	0x7	**K1_SHIFT_VALUE_FAST_MODE**

ARAD_LANE_1_LOCK_UNLOCK_DETECTOR_CONFIG_0

Offset: 0x60
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00340500 (0bxxxx,xxxx,0011,0100,xxx0,0101,xxxx,xxx0)

Bit	Reset	Description
23:20	0x3	**ERROR_SPREAD_WINDOW_SIZE**
19:16	0x4	**K3_SHIFT_VALUE**
12:8	0x5	**CONVERGENCE_TIMER_CNT**
0	TIME_BASED	**LOCK_DETECT_METHOD**: **0** = TIME_BASED **1** = ERROR_THRESHOLD_BASED

ARAD_LANE_1_ERROR_LOCK_THRESHOLD_0

Offset: 0x64
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00040000 (0b0000,0000,0000,0100,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x40000	**ERROR_LOCK_THRESHOLD**

ARAD_LANE_1_ERROR_UNLOCK_THRESHOLD_0

Offset: 0x68
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000a0000 (0b0000,0000,0000,1010,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0xa0000	**ERROR_UNLOCK_THRESHOLD**

ARAD_LANE_1_RATIO_CALCULATOR_CONFIG_0

Offset: 0x6c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0f020006 (0b0000,1111,0000,0010,xxxx,xxx0,xxx0,x110)

Bit	Reset	Description
31:24	0xf	**ACCEPTABLE_RATIO_SPREAD_FACTOR**
23:16	0x2	**PPM_SENSITIVITY_SHIFT_VALUE**
8	DENOMINATOR_CLOCK	**RATIO_CALCULATOR_CLOCK**: **0** = DENOMINATOR_CLOCK **1** = NUMERATOR_CLOCK
4	AVERAGE_RATIO_VALUES	**AVERAGING_METHOD**: **0** = AVERAGE_RATIO_VALUES **1** = AVERAGE_PERIOD_COUNTS
2:0	0x6	**AVERAGE_SHIFT_VALUE**

ARAD_LANE_1_CYA_0

Offset: 0x70
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CYA_BITS**

ARAD_LANE2_NUMERATOR_MUX_SEL_0

Offset: 0x78
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,0000,xxxx,xxxx,0000,xxxx,xx00,0000)

Bit	Reset	Description
27	FALSE	**IQC2_SEL**: **0** = FALSE **1** = TRUE
26	FALSE	**IQC1_SEL**: **0** = FALSE **1** = TRUE
25	FALSE	**DSPK2_SEL**: **0** = FALSE **1** = TRUE
24	FALSE	**DSPK1_SEL**: **0** = FALSE **1** = TRUE
15	FALSE	**DMIC4_SEL**: **0** = FALSE **1** = TRUE
14	FALSE	**DMIC3_SEL**: **0** = FALSE **1** = TRUE
13	FALSE	**DMIC2_SEL**: **0** = FALSE **1** = TRUE
12	FALSE	**DMIC1_SEL**: **0** = FALSE **1** = TRUE
5	FALSE	**I2S6_SEL**: **0** = FALSE **1** = TRUE
4	FALSE	**I2S5_SEL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
3	FALSE	**I2S4_SEL**: **0** = FALSE **1** = TRUE
2	FALSE	**I2S3_SEL**: **0** = FALSE **1** = TRUE
1	FALSE	**I2S2_SEL**: **0** = FALSE **1** = TRUE
0	FALSE	**I2S1_SEL**: **0** = FALSE **1** = TRUE

ARAD_LANE2_NUMERATOR_PRESCALAR_0

Offset: 0x7c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PRESCALAR**

ARAD_LANE2_DENOMINATOR_MUX_SEL_0

Offset: 0x80
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,0000,xxxx,xxxx,0000,xxxx,xx00,0000)

Bit	Reset	Description
27	FALSE	**IQC2_SEL**: **0** = FALSE **1** = TRUE
26	FALSE	**IQC1_SEL**: **0** = FALSE **1** = TRUE
25	0x0	**DSPK2_SEL**: **0** = FALSE **1** = TRUE
24	0x0	**DSPK1_SEL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
15	0x0	**DMIC4_SEL**: **0** = FALSE **1** = TRUE
14	0x0	**DMIC3_SEL**: **0** = FALSE **1** = TRUE
13	0x0	**DMIC2_SEL**: **0** = FALSE **1** = TRUE
12	0x0	**DMIC1_SEL**: **0** = FALSE **1** = TRUE
5	0x0	**I2S6_SEL**: **0** = FALSE **1** = TRUE
4	0x0	**I2S5_SEL**: **0** = FALSE **1** = TRUE
3	0x0	**I2S4_SEL**: **0** = FALSE **1** = TRUE
2	0x0	**I2S3_SEL**: **0** = FALSE **1** = TRUE
1	0x0	**I2S2_SEL**: **0** = FALSE **1** = TRUE
0	0x0	**I2S1_SEL**: **0** = FALSE **1** = TRUE

ARAD_LANE2_DENOMINATOR_PRESCALAR_0

Offset: 0x84
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PRESCALAR**

ARAD_LANE2_RATIO_INTEGER_PART_0

Offset: 0x88
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VALUE**

ARAD_LANE2_RATIO_FRACTIONAL_PART_0

Offset: 0x8c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VALUE**

ARAD_LANE3_NUMERATOR_MUX_SEL_0

Offset: 0xb0
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,0000,xxxx,xxxx,0000,xxxx,xx00,0000)

Bit	Reset	Description
27	FALSE	**IQC2_SEL**: **0** = FALSE **1** = TRUE
26	FALSE	**IQC1_SEL**: **0** = FALSE **1** = TRUE
25	FALSE	**DSPK2_SEL**: **0** = FALSE **1** = TRUE
24	FALSE	**DSPK1_SEL**: **0** = FALSE **1** = TRUE
15	FALSE	**DMIC4_SEL**: **0** = FALSE **1** = TRUE
14	FALSE	**DMIC3_SEL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
13	FALSE	**DMIC2_SEL**: **0** = FALSE **1** = TRUE
12	FALSE	**DMIC1_SEL**: **0** = FALSE **1** = TRUE
5	FALSE	**I2S6_SEL**: **0** = FALSE **1** = TRUE
4	FALSE	**I2S5_SEL**: **0** = FALSE **1** = TRUE
3	FALSE	**I2S4_SEL**: **0** = FALSE **1** = TRUE
2	FALSE	**I2S3_SEL**: **0** = FALSE **1** = TRUE
1	FALSE	**I2S2_SEL**: **0** = FALSE **1** = TRUE
0	FALSE	**I2S1_SEL**: **0** = FALSE **1** = TRUE

ARAD_LANE3_NUMERATOR_PRESCALAR_0

Offset: 0xb4
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PRESCALAR**

ARAD_LANE3_DENOMINATOR_MUX_SEL_0

Offset: 0xb8
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,0000,xxxx,xxxx,0000,xxxx,xx00,0000)

Bit	Reset	Description
27	FALSE	**IQC2_SEL**: **0** = FALSE **1** = TRUE
26	FALSE	**IQC1_SEL**: **0** = FALSE **1** = TRUE
25	0x0	**DSPK2_SEL**: **0** = FALSE **1** = TRUE
24	0x0	**DSPK1_SEL**: **0** = FALSE **1** = TRUE
15	0x0	**DMIC4_SEL**: **0** = FALSE **1** = TRUE
14	0x0	**DMIC3_SEL**: **0** = FALSE **1** = TRUE
13	0x0	**DMIC2_SEL**: **0** = FALSE **1** = TRUE
12	0x0	**DMIC1_SEL**: **0** = FALSE **1** = TRUE
5	0x0	**I2S6_SEL**: **0** = FALSE **1** = TRUE
4	0x0	**I2S5_SEL**: **0** = FALSE **1** = TRUE
3	0x0	**I2S4_SEL**: **0** = FALSE **1** = TRUE
2	0x0	**I2S3_SEL**: **0** = FALSE **1** = TRUE
1	0x0	**I2S2_SEL**: **0** = FALSE **1** = TRUE
0	0x0	**I2S1_SEL**: **0** = FALSE **1** = TRUE

ARAD_LANE3_DENOMINATOR_PRESCALAR_0

Offset: 0xbc
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PRESCALAR**

ARAD_LANE3_RATIO_INTEGER_PART_0

Offset: 0xc0
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VALUE**

ARAD_LANE3_RATIO_FRACTIONAL_PART_0

Offset: 0xc4
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VALUE**

ARAD_LANE4_NUMERATOR_MUX_SEL_0

Offset: 0xe8
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,0000,xxxx,xxxx,0000,xxxx,xx00,0000)

Bit	Reset	Description
27	FALSE	**IQC2_SEL**: **0** = FALSE **1** = TRUE
26	FALSE	**IQC1_SEL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
25	FALSE	**DSPK2_SEL**: **0** = FALSE **1** = TRUE
24	FALSE	**DSPK1_SEL**: **0** = FALSE **1** = TRUE
15	FALSE	**DMIC4_SEL**: **0** = FALSE **1** = TRUE
14	FALSE	**DMIC3_SEL**: **0** = FALSE **1** = TRUE
13	FALSE	**DMIC2_SEL**: **0** = FALSE **1** = TRUE
12	FALSE	**DMIC1_SEL**: **0** = FALSE **1** = TRUE
5	FALSE	**I2S6_SEL**: **0** = FALSE **1** = TRUE
4	FALSE	**I2S5_SEL**: **0** = FALSE **1** = TRUE
3	FALSE	**I2S4_SEL**: **0** = FALSE **1** = TRUE
2	FALSE	**I2S3_SEL**: **0** = FALSE **1** = TRUE
1	FALSE	**I2S2_SEL**: **0** = FALSE **1** = TRUE
0	FALSE	**I2S1_SEL**: **0** = FALSE **1** = TRUE

ARAD_LANE4_NUMERATOR_PRESCALAR_0

Offset: 0xec
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PRESCALAR**

ARAD_LANE4_DENOMINATOR_MUX_SEL_0

Offset: 0xf0
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,0000,xxxx,xxxx,0000,xxxx,xx00,0000)

Bit	Reset	Description
27	FALSE	**IQC2_SEL**: **0** = FALSE **1** = TRUE
26	FALSE	**IQC1_SEL**: **0** = FALSE **1** = TRUE
25	0x0	**DSPK2_SEL**: **0** = FALSE **1** = TRUE
24	0x0	**DSPK1_SEL**: **0** = FALSE **1** = TRUE
15	0x0	**DMIC4_SEL**: **0** = FALSE **1** = TRUE
14	0x0	**DMIC3_SEL**: **0** = FALSE **1** = TRUE
13	0x0	**DMIC2_SEL**: **0** = FALSE **1** = TRUE
12	0x0	**DMIC1_SEL**: **0** = FALSE **1** = TRUE
5	0x0	**I2S6_SEL**: **0** = FALSE **1** = TRUE
4	0x0	**I2S5_SEL**: **0** = FALSE **1** = TRUE
3	0x0	**I2S4_SEL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
2	0x0	**I2S3_SEL**: **0** = FALSE **1** = TRUE
1	0x0	**I2S2_SEL**: **0** = FALSE **1** = TRUE
0	0x0	**I2S1_SEL**: **0** = FALSE **1** = TRUE

ARAD_LANE4_DENOMINATOR_PRESCALAR_0

Offset: 0xf4
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PRESCALAR**

ARAD_LANE4_RATIO_INTEGER_PART_0

Offset: 0xf8
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VALUE**

ARAD_LANE4_RATIO_FRACTIONAL_PART_0

Offset: 0xfc
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VALUE**

ARAD_LANE5_NUMERATOR_MUX_SEL_0

Offset: 0x120
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,0000,xxxx,xxxx,0000,xxxx,xx00,0000)

Bit	Reset	Description
27	FALSE	**IQC2_SEL**: **0** = FALSE **1** = TRUE
26	FALSE	**IQC1_SEL**: **0** = FALSE **1** = TRUE
25	FALSE	**DSPK2_SEL**: **0** = FALSE **1** = TRUE
24	FALSE	**DSPK1_SEL**: **0** = FALSE **1** = TRUE
15	FALSE	**DMIC4_SEL**: **0** = FALSE **1** = TRUE
14	FALSE	**DMIC3_SEL**: **0** = FALSE **1** = TRUE
13	FALSE	**DMIC2_SEL**: **0** = FALSE **1** = TRUE
12	FALSE	**DMIC1_SEL**: **0** = FALSE **1** = TRUE
5	FALSE	**I2S6_SEL**: **0** = FALSE **1** = TRUE
4	FALSE	**I2S5_SEL**: **0** = FALSE **1** = TRUE
3	FALSE	**I2S4_SEL**: **0** = FALSE **1** = TRUE
2	FALSE	**I2S3_SEL**: **0** = FALSE **1** = TRUE
1	FALSE	**I2S2_SEL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	FALSE	**I2S1_SEL**: **0** = FALSE **1** = TRUE

ARAD_LANE5_NUMERATOR_PRESCALAR_0

Offset: 0x124
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PRESCALAR**

ARAD_LANE5_DENOMINATOR_MUX_SEL_0

Offset: 0x128
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,0000,xxxx,xxxx,0000,xxxx,xx00,0000)

Bit	Reset	Description
27	FALSE	**IQC2_SEL**: **0** = FALSE **1** = TRUE
26	FALSE	**IQC1_SEL**: **0** = FALSE **1** = TRUE
25	0x0	**DSPK2_SEL**: **0** = FALSE **1** = TRUE
24	0x0	**DSPK1_SEL**: **0** = FALSE **1** = TRUE
15	0x0	**DMIC4_SEL**: **0** = FALSE **1** = TRUE
14	0x0	**DMIC3_SEL**: **0** = FALSE **1** = TRUE
13	0x0	**DMIC2_SEL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
12	0x0	**DMIC1_SEL**: **0** = FALSE **1** = TRUE
5	0x0	**I2S6_SEL**: **0** = FALSE **1** = TRUE
4	0x0	**I2S5_SEL**: **0** = FALSE **1** = TRUE
3	0x0	**I2S4_SEL**: **0** = FALSE **1** = TRUE
2	0x0	**I2S3_SEL**: **0** = FALSE **1** = TRUE
1	0x0	**I2S2_SEL**: **0** = FALSE **1** = TRUE
0	0x0	**I2S1_SEL**: **0** = FALSE **1** = TRUE

ARAD_LANE5_DENOMINATOR_PRESCALAR_0

Offset: 0x12c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PRESCALAR**

ARAD_LANE5_RATIO_INTEGER_PART_0

Offset: 0x130
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VALUE**

ARAD_LANE5_RATIO_FRACTIONAL_PART_0

Offset: 0x134
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VALUE**

ARAD_LANE6_NUMERATOR_MUX_SEL_0

Offset: 0x158
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,0000,xxxx,xxxx,0000,xxxx,xx00,0000)

Bit	Reset	Description
27	FALSE	**IQC2_SEL**: **0** = FALSE **1** = TRUE
26	FALSE	**IQC1_SEL**: **0** = FALSE **1** = TRUE
25	FALSE	**DSPK2_SEL**: **0** = FALSE **1** = TRUE
24	FALSE	**DSPK1_SEL**: **0** = FALSE **1** = TRUE
15	FALSE	**DMIC4_SEL**: **0** = FALSE **1** = TRUE
14	FALSE	**DMIC3_SEL**: **0** = FALSE **1** = TRUE
13	FALSE	**DMIC2_SEL**: **0** = FALSE **1** = TRUE
12	FALSE	**DMIC1_SEL**: **0** = FALSE **1** = TRUE
5	FALSE	**I2S6_SEL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
4	FALSE	**I2S5_SEL**: **0** = FALSE **1** = TRUE
3	FALSE	**I2S4_SEL**: **0** = FALSE **1** = TRUE
2	FALSE	**I2S3_SEL**: **0** = FALSE **1** = TRUE
1	FALSE	**I2S2_SEL**: **0** = FALSE **1** = TRUE
0	FALSE	**I2S1_SEL**: **0** = FALSE **1** = TRUE

ARAD_LANE6_NUMERATOR_PRESCALAR_0

Offset: 0x15c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PRESCALAR**

ARAD_LANE6_DENOMINATOR_MUX_SEL_0

Offset: 0x160
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,0000,xxxx,xxxx,0000,xxxx,xx00,0000)

Bit	Reset	Description
27	FALSE	**IQC2_SEL**: **0** = FALSE **1** = TRUE
26	FALSE	**IQC1_SEL**: **0** = FALSE **1** = TRUE
25	0x0	**DSPK2_SEL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
24	0x0	**DSPK1_SEL**: **0** = FALSE **1** = TRUE
15	0x0	**DMIC4_SEL**: **0** = FALSE **1** = TRUE
14	0x0	**DMIC3_SEL**: **0** = FALSE **1** = TRUE
13	0x0	**DMIC2_SEL**: **0** = FALSE **1** = TRUE
12	0x0	**DMIC1_SEL**: **0** = FALSE **1** = TRUE
5	0x0	**I2S6_SEL**: **0** = FALSE **1** = TRUE
4	0x0	**I2S5_SEL**: **0** = FALSE **1** = TRUE
3	0x0	**I2S4_SEL**: **0** = FALSE **1** = TRUE
2	0x0	**I2S3_SEL**: **0** = FALSE **1** = TRUE
1	0x0	**I2S2_SEL**: **0** = FALSE **1** = TRUE
0	0x0	**I2S1_SEL**: **0** = FALSE **1** = TRUE

ARAD_LANE6_DENOMINATOR_PRESCALAR_0

Offset: 0x164
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**PRESCALAR**

ARAD_LANE6_RATIO_INTEGER_PART_0

Offset: 0x168
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VALUE**

ARAD_LANE6_RATIO_FRACTIONAL_PART_0

Offset: 0x16c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VALUE**

ARAD_TX_CIF_CTRL_0

Offset: 0x190
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00115500 (0bxx00,0000,0001,0001,x101,x101,0000,0000)

Bit	R/W	Reset	Description
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x1	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
19:16	RO	0x1	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x5	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x5	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RO	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RO	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**: **0** = FALSE **1** = TRUE
1	RO	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RO	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

7.7.4.12 Audio Flow Control (AFC) Registers

AFC_AXBAR_RX_STATUS_0

Offset: 0xc
Read/Write: RO
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01)

Bit	Reset	Description
1	FALSE	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	TRUE	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

AFC_AXBAR_RX_CIF_CTRL_0

Offset: 0x20
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0bxx00,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32

Bit	R/W	Reset	Description
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

AFC_AXBAR_TX_STATUS_0

Offset: 0x4c
Read/Write: RO
Parity Protection: N
Reset: 0x00000002 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x010)

Bit	Reset	Description
2	FALSE	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
1	TRUE	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	FALSE	**TX_ENABLED**: **0** = FALSE **1** = TRUE

AFC_AXBAR_TX_INT_STATUS_0

Offset: 0x50
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	CLEAR	**TX_DONE**: **0** = CLEAR **1** = SET

AFC_AXBAR_TX_INT_MASK_0

Offset: 0x54
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**TX_DONE**: **0** = UNMASK **1** = MASK

AFC_AXBAR_TX_INT_SET_0

Offset: 0x58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	CLEAR	**TX_DONE**: **0** = CLEAR **1** = SET

AFC_AXBAR_TX_INT_CLEAR_0

Offset: 0x5c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	CLEAR	**TX_DONE**: **0** = CLEAR **1** = SET

AFC_AXBAR_TX_CIF_CTRL_0

Offset: 0x60
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0bxx00,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2
19:16	RO	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32

Bit	R/W	Reset	Description
10:8	RO	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RO	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RO	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RO	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RO	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

AFC_ENABLE_0

Offset: 0x80
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**ENABLE**: **0** = FALSE **1** = TRUE

AFC_SOFT_RESET_0

Offset: 0x84
Read/Write: RW

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**SOFT_RESET**: **0** = FALSE **1** = TRUE

AFC_CG_0

Offset: 0x88
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	TRUE	**SLCG_EN**: Second level clock gating enable,for first 2 channels and global/common logic. **0** = FALSE **1** = TRUE

AFC_STATUS_0

Offset: 0x8c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,x0x0)

Bit	Reset	Description
31	FALSE	**CNFG_ERR**: **0** = FALSE **1** = TRUE
5	FALSE	**INTERP_RATE_SLOW**: **0** = FALSE **1** = TRUE
4	FALSE	**DECIM_RATE_SLOW**: **0** = FALSE **1** = TRUE
2	FALSE	**CLKEN**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	FALSE	**TX_ENABLED**: **0** = FALSE **1** = TRUE

AFC_INT_STATUS_0

Offset: 0x90
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	CLEAR	**TX_DONE**: **0** = CLEAR **1** = SET
0	CLEAR	**CLK_PPM_DIFF_ERROR**: **0** = CLEAR **1** = SET

AFC_INT_MASK_0

Offset: 0x94
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**CLK_PPM_DIFF_ERROR**: **0** = UNMASK **1** = MASK

AFC_INT_SET_0

Offset: 0x98
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	CLEAR	**CLK_PPM_DIFF_ERROR**: **0** = CLEAR **1** = SET

AFC_INT_CLEAR_0

Offset: 0x9c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	CLEAR	**CLK_PPM_DIFF_ERROR**: **0** = CLEAR **1** = SET

AFC_DEST_OUTPUT_MODULE_PARAMS_0

Offset: 0xa4
Read/Write: RW
Parity Protection: N
Reset: 0x01190e0c (0bxxx0,0001,x001,1001,x000,1110,x000,1100)

Bit	Reset	Description
28:27	0x0	**MODULE_SELECT**: **0** = I2S **1** = DSPK
26:24	0x1	**MODULE_ID**: I2S1 or DSPK1 = 1, I2S2 or DSPK2 = 2 and so on.
22:16	0x19	**FIFO_HIGH_THRESHOLD**
14:8	0xe	**FIFO_START_THRESHOLD**
6:0	0xc	**FIFO_LOW_THRESHOLD**

AFC_TXCIF_FIFO_PARAMS_0

Offset: 0xa8
Read/Write: RW

Parity Protection: N
Reset: 0x00190e0c (0bxxxx,xxxx,x001,1001,x000,1110,x000,1100)

Bit	Reset	Description
22:16	0x19	**FIFO_HIGH_THRESHOLD**
14:8	0xe	**FIFO_START_THRESHOLD**
6:0	0xc	**FIFO_LOW_THRESHOLD**

AFC_CLK_PPM_DIFF_0

Offset: 0xac
Read/Write: RW
Parity Protection: N
Reset: 0x0000001e (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0001,1110)

Bit	Reset	Description
15:0	0x1e	**VALUE**: We expect most CODECs to have +/- 10 ppm. So the max diff is 20 ppm.// A value of 30 is chosen to be higher than 20.

AFC_AXBAR_RX_STATUS_0

Offset: 0xc
Read/Write: RO
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01)

Bit	Reset	Description
1	FALSE	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	TRUE	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

AFC_AXBAR_RX_CIF_CTRL_0

Offset: 0x20
Read/Write: See table below

Parity Protection: N
Reset: 0x00007700 (0bxx00,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

AFC_AXBAR_TX_STATUS_0

Offset: 0x4c
Read/Write: RO
Parity Protection: N
Reset: 0x00000002 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x010)

Bit	Reset	Description
2	FALSE	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
1	TRUE	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	FALSE	**TX_ENABLED**: **0** = FALSE **1** = TRUE

AFC_AXBAR_TX_INT_STATUS_0

Offset: 0x50
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	CLEAR	**TX_DONE**: **0** = CLEAR **1** = SET

AFC_AXBAR_TX_INT_MASK_0

Offset: 0x54
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**TX_DONE**: **0** = UNMASK **1** = MASK

AFC_AXBAR_TX_INT_SET_0

Offset: 0x58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	CLEAR	**TX_DONE**: **0** = CLEAR **1** = SET

AFC_AXBAR_TX_INT_CLEAR_0

Offset: 0x5c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	CLEAR	**TX_DONE**: **0** = CLEAR **1** = SET

AFC_AXBAR_TX_CIF_CTRL_0

Offset: 0x60
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0bxx00,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2

Bit	R/W	Reset	Description
19:16	RO	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RO	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RO	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RO	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RO	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

AFC_ENABLE_0

Offset: 0x80
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**ENABLE**: **0** = FALSE **1** = TRUE

AFC_SOFT_RESET_0

Offset: 0x84
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**SOFT_RESET**: **0** = FALSE **1** = TRUE

AFC_CG_0

Offset: 0x88
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	TRUE	**SLCG_EN**: Second level clock gating enable for first two channels and global/ common logic. **0** = FALSE **1** = TRUE

AFC_STATUS_0

Offset: 0x8c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,x0x0)

Bit	Reset	Description
31	FALSE	**CNFG_ERR**: **0** = FALSE **1** = TRUE
5	FALSE	**INTERP_RATE_SLOW**: **0** = FALSE **1** = TRUE
4	FALSE	**DECIM_RATE_SLOW**: **0** = FALSE **1** = TRUE
2	FALSE	**CLKEN**: **0** = FALSE **1** = TRUE
0	FALSE	**TX_ENABLED**: **0** = FALSE **1** = TRUE

AFC_INT_STATUS_0

Offset: 0x90
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	CLEAR	**TX_DONE**: **0** = CLEAR **1** = SET
0	CLEAR	**CLK_PPM_DIFF_ERROR**: **0** = CLEAR **1** = SET

AFC_INT_MASK_0

Offset: 0x94
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**CLK_PPM_DIFF_ERROR**: **0** = UNMASK **1** = MASK

AFC_INT_SET_0

Offset: 0x98
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	CLEAR	**CLK_PPM_DIFF_ERROR**: **0** = CLEAR **1** = SET

AFC_INT_CLEAR_0

Offset: 0x9c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	CLEAR	**CLK_PPM_DIFF_ERROR**: **0** = CLEAR **1** = SET

AFC_DEST_OUTPUT_MODULE_PARAMS_0

Offset: 0xa4
Read/Write: RW
Parity Protection: N
Reset: 0x01190e0c (0bxxx0,0001,x001,1001,x000,1110,x000,1100)

Bit	Reset	Description
28:27	0x0	**MODULE_SELECT**: **0** = I2S **1** = DSPK

Bit	Reset	Description
26:24	0x1	**MODULE_ID**: I2S1 or DSPK1 = 1, I2S2 or DSPK2 = 2 and so on.
22:16	0x19	**FIFO_HIGH_THRESHOLD**
14:8	0xe	**FIFO_START_THRESHOLD**
6:0	0xc	**FIFO_LOW_THRESHOLD**

AFC_TXCIF_FIFO_PARAMS_0

Offset: 0xa8
Read/Write: RW
Parity Protection: N
Reset: 0x00190e0c (0bxxxx,xxxx,x001,1001,x000,1110,x000,1100)

Bit	Reset	Description
22:16	0x19	**FIFO_HIGH_THRESHOLD**
14:8	0xe	**FIFO_START_THRESHOLD**
6:0	0xc	**FIFO_LOW_THRESHOLD**

AFC_CLK_PPM_DIFF_0

Offset: 0xac
Read/Write: RW
Parity Protection: N
Reset: 0x0000001e (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0001,1110)

Bit	Reset	Description
15:0	0x1e	**VALUE**: We expect most CODECs to have +/- 10 ppm. So the maximum diff is 20 ppm. A value of 30 is chosen to be higher than 20.

AFC_LCOEF_1_4_0_0

The LCOEF_1_4 coefficients are {46, -1562, 8394, 28999, -3714, 480}

Offset: 0xc4
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x0000002e (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0010,1110)

Bit	Reset	Description
15:0	0x2e	**COEF**

AFC_LCOEF_1_4_1_0
Offset: 0xc8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000f9e6 (0bxxxx,xxxx,xxxx,xxxx,1111,1001,1110,0110)

Bit	Reset	Description
15:0	0xf9e6	**COEF**

AFC_LCOEF_1_4_2_0
Offset: 0xcc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000020ca (0bxxxx,xxxx,xxxx,xxxx,0010,0000,1100,1010)

Bit	Reset	Description
15:0	0x20ca	**COEF**

AFC_LCOEF_1_4_3_0
Offset: 0xd0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00007147 (0bxxxx,xxxx,xxxx,xxxx,0111,0001,0100,0111)

Bit	Reset	Description
15:0	0x7147	**COEF**

AFC_LCOEF_1_4_4_0
Offset: 0xd4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000f17e (0bxxxx,xxxx,xxxx,xxxx,1111,0001,0111,1110)

Bit	Reset	Description
15:0	0xf17e	**COEF**

AFC_LCOEF_1_4_5_0
Offset: 0xd8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000001e0 (0bxxxx,xxxx,xxxx,xxxx,0000,0001,1110,0000)

Bit	Reset	Description
15:0	0x1e0	**COEF**

AFC_LCOEF_2_4_0_0

The LCOEF_2_4 coefficients are {279, -3477, 19463, 19463, -3477, 279}; given the symmetry, we only use three registers.

Offset: 0xdc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000117 (0bxxxx,xxxx,xxxx,xxxx,0000,0001,0001,0111)

Bit	Reset	Description
15:0	0x117	**COEF**

AFC_LCOEF_2_4_1_0
Offset: 0xe0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000f26b (0bxxxx,xxxx,xxxx,xxxx,1111,0010,0110,1011)

Bit	Reset	Description
15:0	0xf26b	**COEF**

AFC_LCOEF_2_4_2_0
Offset: 0xe4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00004c07 (0bxxxx,xxxx,xxxx,xxxx,0100,1100,0000,0111)

Bit	Reset	Description
15:0	0x4c07	**COEF**

AFC_CYA_0

Setting this to '0' will make the threshold monitoring logic works as follows:
AFC starts with monitoring destination module FIFO threshold. if AFC TXCIF FIFO threshold > AFC_THRESHOLDS_AFC_0 then automatically AFC switches to monitor its own TXCIF FIFO threshold. Once threshold monitor logic switches to monitor AFC_THRESHOLDS_AFC_0, it does not come back to destination module threshold unless entire AFC is hardware reset.

Setting this to '1' will make the threshold monitoring logic works as follows:
AFC starts with monitoring destination module FIFO threshold. if AFC TXCIF FIFO threshold > AFC_THRESHOLDS_AFC_0 then automatically AFC switches to monitor its own TXCIF FIFO threshold. if module is soft reset or disabled, this switch goes back to monitor destination module FIFO threshold.

Offset: 0xe8
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000001 (0b0000,0000,0000,0000,0000,0000,0000,0001)

Bit	Reset	Description
31:0	0x1	**CYA_BITS**

7.7.4.13 Sampling Frequency Converter (SFC) Control Registers

SFC_AXBAR_RX_STATUS_0

Offset: 0xc
Read/Write: RO
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01)

Bit	Reset	Description
1	FALSE	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	TRUE	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

SFC_AXBAR_RX_INT_STATUS_0

Offset: 0x10
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**RX_DONE**: **0** = FALSE **1** = TRUE

SFC_AXBAR_RX_INT_MASK_0

Offset: 0x14
Read/Write: RW

Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**RX_DONE**: **0** = UNMASK **1** = MASK

SFC_AXBAR_RX_INT_SET_0

Offset: 0x18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**RX_DONE**: This bit is auto cleared after an interrupt in generated **0** = FALSE **1** = TRUE

SFC_AXBAR_RX_INT_CLEAR_0

Offset: 0x1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**RX_DONE**: **0** = FALSE **1** = TRUE

SFC_AXBAR_RX_CIF_CTRL_0

Offset: 0x20
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0bxx00,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

SFC_AXBAR_RX_FREQ_0

Offset: 0x24
Read/Write: RW

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	0x0	**FS_IN**: **0** = FS8 **1** = FS11_025 **2** = FS16 **3** = FS22_05 **4** = FS24 **5** = FS32 **6** = FS44_1 **7** = FS48 **8** = FS64 **9** = FS88_2 **10** = FS96 **11** = FS176_4 **12** = FS192

SFC_AXBAR_TX_STATUS_0

Offset: 0x4c
Read/Write: RO
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01)

Bit	Reset	Description
1	FALSE	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	TRUE	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

SFC_AXBAR_TX_INT_STATUS_0

Offset: 0x50
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**TX_DONE**: **0** = FALSE **1** = TRUE

SFC_AXBAR_TX_INT_MASK_0

Offset: 0x54
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**TX_DONE**: **0** = UNMASK **1** = MASK

SFC_AXBAR_TX_INT_SET_0

Offset: 0x58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**TX_DONE**: This bit is auto cleared after an interrupt in generated **0** = FALSE **1** = TRUE

SFC_AXBAR_TX_INT_CLEAR_0

Offset: 0x5c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**TX_DONE**: **0** = FALSE **1** = TRUE

SFC_AXBAR_TX_CIF_CTRL_0

Offset: 0x60
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0bxx00,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2
19:16	RO	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

SFC_AXBAR_TX_FREQ_0

Offset: 0x64
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	0x0	**FS_OUT**: **0** = FS8 **1** = FS11_025 **2** = FS16 **3** = FS22_05 **4** = FS24 **5** = FS32 **6** = FS44_1 **7** = FS48 **8** = FS64 **9** = FS88_2 **10** = FS96 **11** = FS176_4 **12** = FS192

SFC_ENABLE_0

Offset: 0x80
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**ENABLE**: **0** = FALSE **1** = TRUE

SFC_SOFT_RESET_0

Offset: 0x84
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**SOFT_RESET**: **0** = FALSE **1** = TRUE

SFC_CG_0

Offset: 0x88
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	TRUE	**SLCG_EN**: Second level clock gating enable,for first 2 channels and global/common logic. **0** = FALSE **1** = TRUE

SFC_STATUS_0

Offset: 0x8c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxx0,xxxx,xxx0)

Bit	Reset	Description
8	FALSE	**SLCG_CLKEN**: **0** = FALSE **1** = TRUE
0	FALSE	**ENABLE_STATUS**: **0** = FALSE **1** = TRUE

SFC_INT_STATUS_0

Offset: 0x90
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	FALSE	**TX_DONE**: **0** = FALSE **1** = TRUE
0	FALSE	**RX_DONE**: **0** = FALSE **1** = TRUE

SFC_COEF_RAM_0

Offset: 0xbc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DISABLE	**COEF_RAM_EN**: **0** = DISABLE **1** = ENABLE

SFC_AHUBRAMCTL_SFC_CTRL_0

Offset: 0xc0
Read/Write: See table below
Parity Protection: N
Reset: 0x00004000 (0b0xxx,xxxx,0000,0000,x100,xxx0,0000,0000)

Bit	R/W	Reset	Description
31	RO	DONE	**READ_BUSY**: **0** = DONE **1** = BUSY
23:16	RW	0x0	**SEQ_READ_COUNT**
14	RW	WRITE	**RW**: **0** = READ **1** = WRITE
13	RW	DISABLE	**ADDR_INIT_EN**: **0** = DISABLE **1** = ENABLE
12	RW	DISABLE	**SEQ_ACCESS_EN**: **0** = DISABLE **1** = ENABLE

Bit	R/W	Reset	Description
8:0	RW	0x0	**RAM_ADDR**

SFC_AHUBRAMCTL_SFC_DATA_0

Offset: 0xc4
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

7.7.4.14 Output Processing Engine (OPE) Control Registers

OPE_AXBAR_RX_STATUS_0

Offset: 0xc
Read/Write: RO
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01)

Bit	Reset	Description
1	FALSE	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	TRUE	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

OPE_AXBAR_RX_INT_STATUS_0

Offset: 0x10
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	CLEAR	**RX_DONE**: **0** = CLEAR **1** = SET

OPE_AXBAR_RX_INT_MASK_0

Offset: 0x14
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**RX_DONE**: **0** = UNMASK **1** = MASK

OPE_AXBAR_RX_INT_SET_0

Offset: 0x18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	CLEAR	**RX_DONE**: This bit is auto cleared after an interrupt in generated **0** = CLEAR **1** = SET

OPE_AXBAR_RX_INT_CLEAR_0

Offset: 0x1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	CLEAR	**RX_DONE**: **0** = CLEAR **1** = SET

OPE_AXBAR_RX_CIF_CTRL_0

Offset: 0x20
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0bxx00,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8
19:16	RO	0x0	**CLIENT_CHANNELS**: Read only = RXCIF_XBAR_CHANNELS; HW is tied to what is programmed for RXCIF XBAR Channels **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RO	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RO	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RO	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP

Bit	R/W	Reset	Description
0	RO	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

OPE_AXBAR_TX_STATUS_0

Offset: 0x4c
Read/Write: RO
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01)

Bit	Reset	Description
1	FALSE	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	TRUE	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

OPE_AXBAR_TX_INT_STATUS_0

Offset: 0x50
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	CLEAR	**TX_DONE**: **0** = CLEAR **1** = SET

OPE_AXBAR_TX_INT_MASK_0

Offset: 0x54
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**TX_DONE**: **0** = UNMASK **1** = MASK

OPE_AXBAR_TX_INT_SET_0

Offset: 0x58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	CLEAR	**TX_DONE**: This bit is auto cleared after an interrupt in generated **0** = CLEAR **1** = SET

OPE_AXBAR_TX_INT_CLEAR_0

Offset: 0x5c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	CLEAR	**TX_DONE**: **0** = CLEAR **1** = SET

OPE_AXBAR_TX_CIF_CTRL_0

Offset: 0x60
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0bxx00,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RO	0x0	**AXBAR_CHANNELS**: Read only = RXCIF_XBAR_CHANNELS; HW is tied to what is programmed for RXCIF XBAR Channels **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8

Bit	R/W	Reset	Description
19:16	RO	0x0	**CLIENT_CHANNELS**: Read only = RXCIF_XBAR_CHANNELS; HW is tied to what is programmed for RXCIF XBAR Channels **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RO	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RO	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RO	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RO	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RO	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

OPE_ENABLE_0

Offset: 0x80
Read/Write: RW

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**ENABLE**: **0** = FALSE **1** = TRUE

OPE_SOFT_RESET_0

Offset: 0x84
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**SOFT_RESET**: **0** = FALSE **1** = TRUE

OPE_CG_0

Offset: 0x88
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	TRUE	**SLCG_EN**: Second level clock gating enable,for first 2 channels and global/common logic. **0** = FALSE **1** = TRUE

OPE_STATUS_0

Offset: 0x8c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxx0,xxxx,xxx0)

Bit	Reset	Description
8	FALSE	**SLCG_CLKEN**: **0** = FALSE **1** = TRUE
0	FALSE	**ENABLE_STATUS**: **0** = FALSE **1** = TRUE

OPE_INT_STATUS_0

Offset: 0x90
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	CLEAR	**TX_DONE**: **0** = CLEAR **1** = SET
0	CLEAR	**RX_DONE**: **0** = CLEAR **1** = SET

OPE_DIRECTION_0

Offset: 0x94
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**DIR**: 0 --> (data flows MBDRC to PEQ) - default 1 --> (data flows PEQ to MBDRC)

7.7.4.15 Parametric Equalizer (PEQ) Control Registers

PEQ_SOFT_RST_0

Offset: 0x0
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	DISABLE	**SOFT_RESET**: This bit is Auto Cleared. Resets PEQ logic - FSM and control states **0** = DISABLE :SW Should wait until this bit is cleared **1** = ENABLE

PEQ_CG_0

Offset: 0x4
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**SLCG_EN**: PEQ Idle Detection Enable - To Enable Second Level Clock Gating **0** = FALSE **1** = TRUE

PEQ_STATUS_0

Offset: 0x8
Read/Write: RO
Parity Protection: N
Reset: 0x00000001 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	TRUE	**SLCG_CLKEN**: Idle Detect Enable **0** = FALSE **1** = TRUE

PEQ_CONFIG_0

Offset: 0xc
Read/Write: RW
Parity Protection: N
Reset: 0x00000013 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0011)

Bit	Reset	Description
5:2	0x4	**BIQUAD_STAGES**: Number of BiQuad Stages in PEQ Chain (Maximum of 12 - for current arch) This number is N-1, example for (max) 12 - it should be 'd11
1	UNBIAS	**BIAS_UNBIAS**: Rounding Option across all Rounding Operations of MBDRC Biased = Round to plus infinity (both positive and negative num) **1** = UNBIAS :Unbias = Round to plus infinity (positive num) and minus infinity (negative num) **0** = BIAS
0	ACTIVE	**MODE**: Select Mode - when 0 - Bypass, when 1 - Active - will do PreGain, BiQuad, PostGain **0** = BYPASS : **1** = ACTIVE

PEQ_AHUBRAMCTL_PEQ_CTRL_0

Offset: 0x10
Read/Write: See table below
Parity Protection: N
Reset: 0x00004000 (0b0xxx,xxxx,0000,0000,x100,xxx0,0000,0000)

Bit	R/W	Reset	Description
31	RO	DONE	**READ_BUSY**: **0** = DONE **1** = BUSY
23:16	RW	0x0	**SEQ_READ_COUNT**
14	RW	WRITE	**RW**: **0** = READ **1** = WRITE
13	RW	DISABLE	**ADDR_INIT_EN**: **0** = DISABLE **1** = ENABLE
12	RW	DISABLE	**SEQ_ACCESS_EN**: **0** = DISABLE **1** = ENABLE
8:0	RW	0x0	**RAM_ADDR**

PEQ_AHUBRAMCTL_PEQ_DATA_0

Offset: 0x14
Read/Write: RW

Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

PEQ_AHUBRAMCTL_SHIFT_CTRL_0

Offset: 0x18
Read/Write: See table below
Parity Protection: N
Reset: 0x00004000 (0b0xxx,xxxx,0000,0000,x100,xxx0,0000,0000)

Bit	R/W	Reset	Description
31	RO	DONE	**READ_BUSY**: **0** = DONE **1** = BUSY
23:16	RW	0x0	**SEQ_READ_COUNT**
14	RW	WRITE	**RW**: **0** = READ **1** = WRITE
13	RW	DISABLE	**ADDR_INIT_EN**: **0** = DISABLE **1** = ENABLE
12	RW	DISABLE	**SEQ_ACCESS_EN**: **0** = DISABLE **1** = ENABLE
8:0	RW	0x0	**RAM_ADDR**

PEQ_AHUBRAMCTL_SHIFT_DATA_0

Offset: 0x1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

7.7.4.16 Multiband Dynamic Range Compression (MBDRC) Control Registers

MBDRC_SOFT_RESET_0

MBDRC soft reset cannot be stand-alone (will affect whole data flow of OPE) So to avoid issues, SW should not use this register. But simply use OPE soft reset register to affect both MBDRC and PEQ as well as CIFs.

Offset: 0x4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**SOFT_RESET**: Soft Reset is a self-clearing bit. Soft Reset resets all FSM, flushes flow control FIFO and resets the state registers. It also bring the module back to disabled state (without flushing data in the pipe) **0** = FALSE **1** = TRUE

MBDRC_CG_0

Offset: 0x8
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**SLCG_EN**: Second level clock gating enable. **0** = FALSE **1** = TRUE

MBDRC_STATUS_0

Offset: 0xc
Read/Write: RO

Parity Protection: N
Reset: 0x00000001 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	TRUE	**SLCG_CLKEN**: **0** = FALSE **1** = TRUE

MBDRC_CONFIG_0

Offset: 0x28
Read/Write: RW
Parity Protection: N
Reset: 0x0030de51 (0bxxxx,xxx0,0011,0000,1101,1110,0101,xx01)

Bit	Reset	Description
24:16	0x30	**RMS_OFFSET**: RMS Offset used during RMS detection(-16dB to +15.9375dB) (Q5.4 format)
15	UNBIAS	**BIAS_UNBIAS**: Rounding Option across all Rounding Operations of MBDRC Biased = Round to plus infinity (both positive and negative num). **1** = UNBIAS: Unbias = Round to plus infinity (positive num) and minus infinity (negative num) **0** = BIAS
14	PEAK	**PEAK_RMS**: Mode Select for Peak Detection in SideChain. **0** = RMS **1** = PEAK
13	ALL_PASS_TREE	**FILTER_STRUC**: Selects between a custom Butterfly Structure with fixed BiQuad stages (ALL_PASS_TREE). **0** = ALL_PASS_TREE :or a programmable structure where per band biquads can be programmed (FLEX) **1** = FLEX
12:8	0x1e	**SHIFT_CTRL**: Shift Control for each BiQuad Stage - this depends on the format of coefficient programmed. Default to 30 - current coefficients are all with 2.30 format. Software can only change it when the format of coeffcients has changed. Otherwise, always keep 30.

Bit	Reset	Description
7:4	N32	**FRAME_SIZE**: Number of Samples per "frame" that will be used to compute the sidechain calculation. **0** = N1 :Has to be a power of 2 **1** = N2 :Maximum Frame size Hardware can support is 64(FRAME_SIZE=6) **2** = N4 :It will result in incorrect output if it is set bigger than 6 **3** = N8 **4** = N16 **5** = N32 **6** = N64
1:0	FULLBAND	**MBDRC_MODE**: MBDRC Mode of Operation. Filtering - LowPass, BandPass and HighPass, SideChain and Gain Application for the three bands. **0** = BYPASS :Bypass - No Filtering, No Side Chain, No Gain Application (DISABLE) **1** = FULLBAND :Full Band - No Filtering, But SideChain and Gain Application - for a single Band **2** = DUALBAND :Filtering - LowPass and HighPass, SideChain and Gain Application for the two Band **3** = MULTIBAND

MBDRC_CHAN_MASK_0

Offset: 0x2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0011)

Bit	Reset	Description
7:0	0x3	**CHAN_MASK_EN**: An N bit field - a "1" on field, indicates perform DRC on that channel. a "0" on field, indicates DRC will not be done on that channel, i.e., no SideChain Calculation, and Unity Gain will be applied. Bit(N-1) - 0 = Masks for Channels(N-1) - 0.

MBDRC_MASTER_VOLUME_0

Offset: 0x30
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MASTER_VOLUME**: Feedback of Volume Control to MBDRC (if Volume Control is placed before MBDRC) (Signed Q9.23 format) (Default is 0 dB).

MBDRC_FAST_FACTOR_0

Offset: 0x34
Read/Write: RW
Parity Protection: N
Reset: 0x30000800 (0b0011,0000,0000,0000,0000,1000,0000,0000)

Bit	Reset	Description
31:16	0x3000	**FR_FACTOR**: Gain Smoothing: FastRelease is applied if (CurrentGain/ PreviousGain < FRFactor). default: 12288 (0.375) (Q1.15 format) positive value only. Defining them as negative values will result in incorrect results.
15:0	0x800	**FA_FACTOR**: Gain Smoothing: FastAttack is applied if (PreviousLevel/ CurrentLevel < FAFactor). default: 2048 (0.0625) (Q1.15 format) positive value only. Defining them as negative values will result in incorrect results.

MBDRC_IIR_CONFIG_0

This is an array of three identical register entries; the register fields below apply to each entry.
Offset: 0x38,...0x40
Read/Write: RW
Parity Protection: N
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0101)

Bit	Reset	Description
3:0	0x5	**NUM_STAGES**: Number of Stages - Maximum of 6,8,6 - only applicable in ALL_PALL_TREE mode. Here number is actually N-1, for max 6 - this number should be 5.

MBDRC_INATTACK_0

This is an array of three identical register entries; the register fields below apply to each entry.
Offset: 0x44,..,0x4c
Read/Write: RW
Parity Protection: N
Reset: 0x3e48590c (0b0011,1110,0100,1000,0101,1001,0000,1100)

Bit	Reset	Description
31:0	0x3e48590c	**IN_ATTACK_TC**: Attack Time Constant for Input envelope detection (different for each band) (1 - exp((-1.0 * BufferSize) / (Fs * INattackTC))). default: 1044928780 (0.001sec @ 48 kHz with 32 BufferSize). positive value only. Defining them as negative values will result in incorrect results.

MBDRC_INRELEASE_0

This is an array of three identical register entries; the register fields below apply to each entry.
Offset: 0x50,..,0x58
Read/Write: RW
Parity Protection: N
Reset: 0x08414e9f (0b0000,1000,0100,0001,0100,1110,1001,1111)

Bit	Reset	Description
31:0	0x8414e9f	**IN_RELEASE_TC**: Release Time Constant for Input envelope detection (different for each band) (1 - exp((-1.0 * BufferSize) / (Fs * INrttackTC))) default: 138497695 (0.01sec @ 48 kHz with 32 BufferSize) positive value only. Defining them as negative values will result in incorrect results

MBDRC_FASTATTACK_0

This is an array of three identical register entries; the register fields below apply to each entry.
Offset: 0x5c,..,0x64
Read/Write: RW
Parity Protection: N
Reset: 0x7fffffff (0b0111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0x7fffffff	**FAST_ATTACK_TC**: Fast Attack Time Constant for Input envelope detection (different for each band) (1 - exp((-1.0 * BufferSize) / (Fs * FastAttackTC))) default: 2147483647 (0.00001sec @ 48 kHz with 32 BufferSize) positive value only. Defining them as negative values will result in incorrect results

MBDRC_IN_THRESH_0

This is an array of three identical register entries; the register fields below apply to each entry.
Offset: 0x68,..,0x70

Read/Write: RW
Parity Protection: N
Reset: 0x06145082 (0b0000,0110,0001,0100,0101,0000,1000,0010)

Bit	Reset	Description
31:24	0x6	**IN_THRESH_4TH**: (default: 6) Fourth Knee for the 5 gain segments, which is close to 0 dB (0dB to -127.5dB in 0.5dB step size) (After conversion Thresh -40 is provided as 40)
23:16	0x14	**IN_THRESH_3RD**: (default: 20) Third Knee for the 5 gain segments (0dB to -127.5dB in 0.5dB step size) (After conversion Thresh -40 is provided as 40)
15:8	0x50	**IN_THRESH_2ND**: (default: 80) Second Knee for the 5 gain segments (0dB to -127.5dB in 0.5dB step size) (After conversion Thresh -40 is provided as 40)
7:0	0x82	**IN_THRESH_1ST**: (default: 130) First Knee for the 5 gain segments, which is close to -127.5 dB (0dB to -127.5dB in 0.5dB step size) (After conversion Thresh -40 is provided as 40)

MBDRC_OUT_THRESH_0

This is an array of three identical register entries; the register fields below apply to each entry.
Offset: 0x74,..,0x7c
Read/Write: RW
Parity Protection: N
Reset: 0x060d379b (0b0000,0110,0000,1101,0011,0111,1001,1011)

Bit	Reset	Description
31:24	0x6	**OUT_THRESH_4TH**: (default: 6) Fourth output point corresponding to Fourth IN_THRESH, (0dB to -127.5dB in 0.5dB step size) (After conversion Thresh -40 is provided as 40)
23:16	0xd	**OUT_THRESH_3RD**: (default: 13) Third output point corresponding to Third IN_THRESH, (0dB to -127.5dB in 0.5dB step size) (After conversion Thresh -40 is provided as 40)
15:8	0x37	**OUT_THRESH_2ND**: (default: 55) Second output point corresponding to Second IN_THRESH, (0dB to -127.5dB in 0.5dB step size) (After conversion Thresh -40 is provided as 40)
7:0	0x9b	**OUT_THRESH_1ST**: (default: 155) First output point corresponding to First IN_THRESH, (0dB to -127.5dB in 0.5dB step size) (After conversion Thresh -40 is provided as 40)

MBDRC_RATIO_1ST_0

This is an array of three identical register entries; the register fields below apply to each entry.
Offset: 0x80,..,0x88
Read/Write: RW
Parity Protection: N
Reset: 0x0000a000 (0bxxxx,xxxx,xxxx,xxxx,1010,0000,0000,0000)

Bit	Reset	Description
15:0	0xa000	**RATIO_1ST**: (default: 40960) Output to Input ratio for first gain segment, which is close to -127.5 dB. (Q4.12 format).

MBDRC_RATIO_2ND_0

This is an array of three identical register entries; the register fields below apply to each entry.
Offset: 0x8c,..,0x94
Read/Write: RW
Parity Protection: N
Reset: 0x00002000 (0bxxxx,xxxx,xxxx,xxxx,0010,0000,0000,0000)

Bit	Reset	Description
15:0	0x2000	**RATIO_2ND**: (default: 8192) Output to Input ratio for second gain segment. (Q4.12 format).

MBDRC_RATIO_3RD_0

This is an array of three identical register entries; the register fields below apply to each entry.
Offset: 0x98,..,0xa0
Read/Write: RW
Parity Protection: N
Reset: 0x00000b33 (0bxxxx,xxxx,xxxx,xxxx,0000,1011,0011,0011)

Bit	Reset	Description
15:0	0xb33	**RATIO_3RD**: (default: 2867) Output to Input ratio for third gain segment. (Q4.12 format).

MBDRC_RATIO_4TH_0

This is an array of three identical register entries; the register fields below apply to each entry.
Offset: 0xa4,..,0xac
Read/Write: RW
Parity Protection: N
Reset: 0x00000800 (0bxxxx,xxxx,xxxx,xxxx,0000,1000,0000,0000)

Bit	Reset	Description
15:0	0x800	**RATIO_4TH**: (default: 2048) Output to Input ratio for fourth gain segment. (Q4.12 format).

MBDRC_RATIO_5TH_0

This is an array of three identical register entries; the register fields below apply to each entry.
Offset: 0xb0,..,0xb8
Read/Write: RW
Parity Protection: N
Reset: 0x0000019a (0bxxxx,xxxx,xxxx,xxxx,0000,0001,1001,1010)

Bit	Reset	Description
15:0	0x19a	**RATIO_5TH**: (default: 410) Output to Input ratio for fifth gain segment, which is close to 0 dB. (Q4.12 format).

MBDRC_MAKEUP_GAIN_0

This is an array of three identical register entries; the register fields below apply to each entry.
Offset: 0xbc,..,0xc4
Read/Write: RW
Parity Protection: N
Reset: 0x00000002 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0010)

Bit	Reset	Description
5:0	0x2	**MAKEUP_GAIN**: Makeup Gain (5.1 Format, -16dB to 15.5dB in 0.5dB step size).

MBDRC_INITGAIN_0

This is an array of three identical register entries; the register fields below apply to each entry.
Offset: 0xc8,..,0xd0
Read/Write: RW
Parity Protection: N
Reset: 0x00066666 (0b0000,0000,0000,0110,0110,0110,0110,0110)

Bit	Reset	Description
31:0	0x66666	**GAIN_INIT**: Starting value for the gain to be applied in linear (Gain range in dB = -127dB to 54dB). default: 419430 (0.1 ie. -20dB) (Q10.22 format).

MBDRC_GAINATTACK_0

This is an array of three identical register entries; the register fields below apply to each entry.
Offset: 0xd4,..,0xdc
Read/Write: RW
Parity Protection: N
Reset: 0x00d9ba0e (0b0000,0000,1101,1001,1011,1010,0000,1110)

Bit	Reset	Description
31:0	0xd9ba0e	**GAIN_ATTACK_TC**: Attack Time Constant for gain smoothing (different for each band) (1 - exp((-1.0 * BufferSize) / (Fs * GAINattackTC))). default: 14268942 (0.1sec @ 48 kHz with 32 BufferSize) (Q1.31 format) positive value only. Defining them as negative values will result in incorrect results.

MBDRC_GAINRELEASE_0

This is an array of three identical register entries; the register fields below apply to each entry.
Offset: 0xe0,..,0xe8
Read/Write: RW
Parity Protection: N
Reset: 0x3e48590c (0b0011,1110,0100,1000,0101,1001,0000,1100)

Bit	Reset	Description
31:0	0x3e48590c	**GAIN_RELEASE_TC**: Release Time Constant for gain smoothing (different for each band) (1 - exp((-1.0 * BufferSize) / (Fs * GAINreleaseTC))) default: 1044928780 (0.001sec @ 48 kHz with 32 BufferSize) (Q1.31 format) positive value only. Defining them as negative values will result in incorrect results

MBDRC_FASTRELEASE_0

This is an array of three identical register entries; the register fields below apply to each entry.
Offset: 0xec,..,0xf4
Read/Write: RW
Parity Protection: N
Reset: 0x7ffff26a (0b0111,1111,1111,1111,1111,0010,0110,1010)

Bit	Reset	Description
31:0	0x7ffff26a	**FAST_RELEASE_TC**: Fast Release Time Constant for gain smoothing (different for each band) (1 - exp((-1.0 * BufferSize) / (Fs * FastRelease))) default: 2147480170 (0.00005sec @ 48 kHz with 32 BufferSize) (Q1.31 format) positive value only. Defining them as negative values will result in incorrect results.

MBDRC_AHUBRAMCTL_MBDRC_CTRL_0

This is an array of three identical register entries; the register fields below apply to each entry.
Offset: 0xf8,..,0x100
Read/Write: See table below
Parity Protection: N
Reset: 0x00004000 (0b0xxx,xxxx,0000,0000,x100,xxx0,0000,0000)

Bit	R/W	Reset	Description
31	RO	DONE	**READ_BUSY**: **0** = DONE **1** = BUSY
23:16	RW	0x0	**SEQ_READ_COUNT**
14	RW	WRITE	**RW**: **0** = READ **1** = WRITE
13	RW	DISABLE	**ADDR_INIT_EN**: **0** = DISABLE **1** = ENABLE
12	RW	DISABLE	**SEQ_ACCESS_EN**: **0** = DISABLE **1** = ENABLE
8:0	RW	0x0	**RAM_ADDR**

MBDRC_AHUBRAMCTL_MBDRC_DATA_0

This is an array of three identical register entries; the register fields below apply to each entry.
Offset: 0x104,..,0x10c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

7.7.4.17 Audio Direct Memory Access Interface (ADMAIF) Control Registers

ADMAIF_AXBAR_RX1_ENABLE_0

Offset: 0x0
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX1_SOFT_RESET_0

Offset: 0x4
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX1_STATUS_0

Offset: 0xc
Read/Write: RO
Parity Protection: N
Reset: 0x0XXX00XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
25:16	X	**AVAILABLE_CREDITS**
7	X	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	X	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
5	X	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
4	X	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	X	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX1_INT_STATUS_0

Offset: 0x10
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX1_INT_MASK_0

Offset: 0x14
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_OVERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_RX1_INT_SET_0

Offset: 0x18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX1_INT_CLEAR_0

Offset: 0x1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX1_CIF_CTRL_0

Offset: 0x20
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PACK8_ENABLE**
30	RW	0x0	**PACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_RX1_FIFO_CTRL_0

Offset: 0x28
Read/Write: RW
Parity Protection: N
Reset: 0x00000300 (0b0xxx,xxxx,xxxx,xxxx,xx00,0011,xx00,0000)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
13:8	0x3	**DMA_FIFO_SIZE**
5:0	0x0	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_RX1_FIFO_READ_0

Offset: 0x2c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

ADMAIF_AXBAR_RX2_ENABLE_0

Offset: 0x40
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX2_SOFT_RESET_0

Offset: 0x44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX2_STATUS_0

Offset: 0x4c
Read/Write: RO
Parity Protection: N
Reset: 0x0XXX00XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
25:16	X	**AVAILABLE_CREDITS**
7	X	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	X	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	X	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
4	X	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	X	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX2_INT_STATUS_0

Offset: 0x50
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX2_INT_MASK_0

Offset: 0x54
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_OVERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_RX2_INT_SET_0

Offset: 0x58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX2_INT_CLEAR_0

Offset: 0x5c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX2_CIF_CTRL_0

Offset: 0x60
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PACK8_ENABLE**

Bit	R/W	Reset	Description
30	RW	0x0	**PACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32

Bit	R/W	Reset	Description
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_RX2_FIFO_CTRL_0

Offset: 0x68
Read/Write: RW
Parity Protection: N
Reset: 0x00000304 (0b0xxx,xxxx,xxxx,xxxx,xx00,0011,xx00,0100)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
13:8	0x3	**DMA_FIFO_SIZE**
5:0	0x4	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_RX2_FIFO_READ_0

Offset: 0x6c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

ADMAIF_AXBAR_RX3_ENABLE_0

Offset: 0x80
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX3_SOFT_RESET_0

Offset: 0x84
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX3_STATUS_0

Offset: 0x8c
Read/Write: RO
Parity Protection: N
Reset: 0x0XXX00XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
25:16	X	**AVAILABLE_CREDITS**
7	X	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	X	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	X	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
4	X	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	X	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX3_INT_STATUS_0

Offset: 0x90
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX3_INT_MASK_0

Offset: 0x94
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_OVERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_RX3_INT_SET_0

Offset: 0x98
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX3_INT_CLEAR_0

Offset: 0x9c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX3_CIF_CTRL_0

Offset: 0xa0
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PACK8_ENABLE**
30	RW	0x0	**PACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_RX3_FIFO_CTRL_0

Offset: 0xa8
Read/Write: RW
Parity Protection: N
Reset: 0x00000308 (0b0xxx,xxxx,xxxx,xxxx,xx00,0011,xx00,1000)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
13:8	0x3	**DMA_FIFO_SIZE**
5:0	0x8	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_RX3_FIFO_READ_0

Offset: 0xac
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

ADMAIF_AXBAR_RX4_ENABLE_0

Offset: 0xc0
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX4_SOFT_RESET_0

Offset: 0xc4
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX4_STATUS_0

Offset: 0xcc
Read/Write: RO
Parity Protection: N
Reset: 0x0XXX00XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
25:16	X	**AVAILABLE_CREDITS**
7	X	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	X	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	X	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
4	X	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	X	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX4_INT_STATUS_0

Offset: 0xd0
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX4_INT_MASK_0

Offset: 0xd4
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_OVERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_RX4_INT_SET_0

Offset: 0xd8
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX4_INT_CLEAR_0

Offset: 0xdc
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX4_CIF_CTRL_0

Offset: 0xe0
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PACK8_ENABLE**

Bit	R/W	Reset	Description
30	RW	0x0	**PACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32

Bit	R/W	Reset	Description
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_RX4_FIFO_CTRL_0

Offset: 0xe8
Read/Write: RW
Parity Protection: N
Reset: 0x0000030c (0b0xxx,xxxx,xxxx,xxxx,xx00,0011,xx00,1100)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
13:8	0x3	**DMA_FIFO_SIZE**
5:0	0xc	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_RX4_FIFO_READ_0

Offset: 0xec
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

ADMAIF_AXBAR_RX5_ENABLE_0

Offset: 0x100
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX5_SOFT_RESET_0

Offset: 0x104
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX5_STATUS_0

Offset: 0x10c
Read/Write: RO
Parity Protection: N
Reset: 0x0XXX00XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
25:16	X	**AVAILABLE_CREDITS**
7	X	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	X	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	X	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
4	X	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	X	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX5_INT_STATUS_0

Offset: 0x110
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX5_INT_MASK_0

Offset: 0x114
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_OVERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_RX5_INT_SET_0

Offset: 0x118
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX5_INT_CLEAR_0

Offset: 0x11c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX5_CIF_CTRL_0

Offset: 0x120
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PACK8_ENABLE**
30	RW	0x0	**PACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_RX5_FIFO_CTRL_0

Offset: 0x128
Read/Write: RW
Parity Protection: N
Reset: 0x00000210 (0b0xxx,xxxx,xxxx,xxxx,xx00,0010,xx01,0000)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x10	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_RX5_FIFO_READ_0

Offset: 0x12c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

ADMAIF_AXBAR_RX6_ENABLE_0

Offset: 0x140
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX6_SOFT_RESET_0

Offset: 0x144
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX6_STATUS_0

Offset: 0x14c
Read/Write: RO
Parity Protection: N
Reset: 0x0XXX00XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
25:16	X	**AVAILABLE_CREDITS**
7	X	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	X	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	X	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
4	X	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	X	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX6_INT_STATUS_0

Offset: 0x150
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX6_INT_MASK_0

Offset: 0x154
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_OVERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_RX6_INT_SET_0

Offset: 0x158
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX6_INT_CLEAR_0

Offset: 0x15c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX6_CIF_CTRL_0

Offset: 0x160
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PACK8_ENABLE**

Bit	R/W	Reset	Description
30	RW	0x0	**PACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32

Bit	R/W	Reset	Description
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_RX6_FIFO_CTRL_0

Offset: 0x168
Read/Write: RW
Parity Protection: N
Reset: 0x00000213 (0b0xxx,xxxx,xxxx,xxxx,xx00,0010,xx01,0011)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x13	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_RX6_FIFO_READ_0

Offset: 0x16c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

ADMAIF_AXBAR_RX7_ENABLE_0

Offset: 0x180
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX7_SOFT_RESET_0

Offset: 0x184
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX7_STATUS_0

Offset: 0x18c
Read/Write: RO
Parity Protection: N
Reset: 0x0XXX00XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
25:16	X	**AVAILABLE_CREDITS**
7	X	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	X	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	X	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
4	X	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	X	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX7_INT_STATUS_0

Offset: 0x190
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX7_INT_MASK_0

Offset: 0x194
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_OVERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_RX7_INT_SET_0

Offset: 0x198
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX7_INT_CLEAR_0

Offset: 0x19c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX7_CIF_CTRL_0

Offset: 0x1a0
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PACK8_ENABLE**
30	RW	0x0	**PACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_RX7_FIFO_CTRL_0

Offset: 0x1a8
Read/Write: RW
Parity Protection: N
Reset: 0x00000216 (0b0xxx,xxxx,xxxx,xxxx,xx00,0010,xx01,0110)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x16	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_RX7_FIFO_READ_0

Offset: 0x1ac
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

ADMAIF_AXBAR_RX8_ENABLE_0

Offset: 0x1c0
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX8_SOFT_RESET_0

Offset: 0x1c4
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX8_STATUS_0

Offset: 0x1cc
Read/Write: RO
Parity Protection: N
Reset: 0x0XXX00XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
25:16	X	**AVAILABLE_CREDITS**
7	X	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	X	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	X	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
4	X	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	X	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX8_INT_STATUS_0

Offset: 0x1d0
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX8_INT_MASK_0

Offset: 0x1d4
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_OVERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_RX8_INT_SET_0

Offset: 0x1d8
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX8_INT_CLEAR_0

Offset: 0x1dc
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX8_CIF_CTRL_0

Offset: 0x1e0
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PACK8_ENABLE**

Bit	R/W	Reset	Description
30	RW	0x0	**PACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32

Bit	R/W	Reset	Description
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_RX8_FIFO_CTRL_0

Offset: 0x1e8
Read/Write: RW
Parity Protection: N
Reset: 0x00000219 (0b0xxx,xxxx,xxxx,xxxx,xx00,0010,xx01,1001)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x19	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_RX8_FIFO_READ_0

Offset: 0x1ec
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

ADMAIF_AXBAR_RX9_ENABLE_0

Offset: 0x200
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX9_SOFT_RESET_0

Offset: 0x204
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX9_STATUS_0

Offset: 0x20c
Read/Write: RO
Parity Protection: N
Reset: 0x0XXX00XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
25:16	X	**AVAILABLE_CREDITS**
7	X	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	X	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	X	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
4	X	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	X	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX9_INT_STATUS_0

Offset: 0x210
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX9_INT_MASK_0

Offset: 0x214
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_OVERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_RX9_INT_SET_0

Offset: 0x218
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX9_INT_CLEAR_0

Offset: 0x21c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX9_CIF_CTRL_0

Offset: 0x220
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PACK8_ENABLE**
30	RW	0x0	**PACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_RX9_FIFO_CTRL_0

Offset: 0x228
Read/Write: RW
Parity Protection: N
Reset: 0x0000021c (0b0xxx,xxxx,xxxx,xxxx,xx00,0010,xx01,1100)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x1c	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_RX9_FIFO_READ_0

Offset: 0x22c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

ADMAIF_AXBAR_RX10_ENABLE_0

Offset: 0x240
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX10_SOFT_RESET_0

Offset: 0x244
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX10_STATUS_0

Offset: 0x24c
Read/Write: RO
Parity Protection: N
Reset: 0x0XXX00XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
25:16	X	**AVAILABLE_CREDITS**
7	X	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	X	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	X	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
4	X	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	X	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX10_INT_STATUS_0

Offset: 0x250
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX10_INT_MASK_0

Offset: 0x254
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_OVERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_RX10_INT_SET_0

Offset: 0x258
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX10_INT_CLEAR_0

Offset: 0x25c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX10_CIF_CTRL_0

Offset: 0x260
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PACK8_ENABLE**

Bit	R/W	Reset	Description
30	RW	0x0	**PACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32

Bit	R/W	Reset	Description
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_RX10_FIFO_CTRL_0

Offset: 0x268
Read/Write: RW
Parity Protection: N
Reset: 0x0000021f (0b0xxx,xxxx,xxxx,xxxx,xx00,0010,xx01,1111)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x1f	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_RX10_FIFO_READ_0

Offset: 0x26c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

ADMAIF_AXBAR_RX11_ENABLE_0

Offset: 0x280
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX11_SOFT_RESET_0

Offset: 0x284
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX11_STATUS_0

Offset: 0x28c
Read/Write: RO
Parity Protection: N
Reset: 0x0XXX00XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
25:16	X	**AVAILABLE_CREDITS**
7	X	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	X	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	X	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
4	X	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	X	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX11_INT_STATUS_0

Offset: 0x290
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX11_INT_MASK_0

Offset: 0x294
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_OVERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_RX11_INT_SET_0

Offset: 0x298
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX11_INT_CLEAR_0

Offset: 0x29c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX11_CIF_CTRL_0

Offset: 0x2a0
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PACK8_ENABLE**
30	RW	0x0	**PACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_RX11_FIFO_CTRL_0

Offset: 0x2a8
Read/Write: RW
Parity Protection: N
Reset: 0x00000222 (0b0xxx,xxxx,xxxx,xxxx,xx00,0010,xx10,0010)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x22	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_RX11_FIFO_READ_0

Offset: 0x2ac
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

ADMAIF_AXBAR_RX12_ENABLE_0

Offset: 0x2c0
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX12_SOFT_RESET_0

Offset: 0x2c4
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX12_STATUS_0

Offset: 0x2cc
Read/Write: RO
Parity Protection: N
Reset: 0x0XXX00XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
25:16	X	**AVAILABLE_CREDITS**
7	X	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	X	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	X	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
4	X	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	X	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX12_INT_STATUS_0

Offset: 0x2d0
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX12_INT_MASK_0

Offset: 0x2d4
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_OVERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_RX12_INT_SET_0

Offset: 0x2d8
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX12_INT_CLEAR_0

Offset: 0x2dc
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX12_CIF_CTRL_0

Offset: 0x2e0
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PACK8_ENABLE**

Bit	R/W	Reset	Description
30	RW	0x0	**PACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32

Bit	R/W	Reset	Description
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_RX12_FIFO_CTRL_0

Offset: 0x2e8
Read/Write: RW
Parity Protection: N
Reset: 0x00000225 (0b0xxx,xxxx,xxxx,xxxx,xx00,0010,xx10,0101)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x25	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_RX12_FIFO_READ_0

Offset: 0x2ec
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

ADMAIF_AXBAR_RX13_ENABLE_0

Offset: 0x300
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX13_SOFT_RESET_0

Offset: 0x304
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX13_STATUS_0

Offset: 0x30c
Read/Write: RO
Parity Protection: N
Reset: 0x0XXX00XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
25:16	X	**AVAILABLE_CREDITS**
7	X	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	X	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	X	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
4	X	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	X	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX13_INT_STATUS_0

Offset: 0x310
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX13_INT_MASK_0

Offset: 0x314
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_OVERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_RX13_INT_SET_0

Offset: 0x318
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX13_INT_CLEAR_0

Offset: 0x31c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX13_CIF_CTRL_0

Offset: 0x320
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PACK8_ENABLE**
30	RW	0x0	**PACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_RX13_FIFO_CTRL_0

Offset: 0x328
Read/Write: RW
Parity Protection: N
Reset: 0x00000228 (0b0xxx,xxxx,xxxx,xxxx,xx00,0010,xx10,1000)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x28	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_RX13_FIFO_READ_0

Offset: 0x32c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

ADMAIF_AXBAR_RX14_ENABLE_0

Offset: 0x340
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX14_SOFT_RESET_0

Offset: 0x344
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX14_STATUS_0

Offset: 0x34c
Read/Write: RO
Parity Protection: N
Reset: 0x0XXX00XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
25:16	X	**AVAILABLE_CREDITS**
7	X	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	X	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	X	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
4	X	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	X	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX14_INT_STATUS_0

Offset: 0x350
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX14_INT_MASK_0

Offset: 0x354
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_OVERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_RX14_INT_SET_0

Offset: 0x358
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX14_INT_CLEAR_0

Offset: 0x35c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX14_CIF_CTRL_0

Offset: 0x360
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PACK8_ENABLE**

Bit	R/W	Reset	Description
30	RW	0x0	**PACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32

Bit	R/W	Reset	Description
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_RX14_FIFO_CTRL_0

Offset: 0x368
Read/Write: RW
Parity Protection: N
Reset: 0x0000022b (0b0xxx,xxxx,xxxx,xxxx,xx00,0010,xx10,1011)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x2b	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_RX14_FIFO_READ_0

Offset: 0x36c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

ADMAIF_AXBAR_RX15_ENABLE_0

Offset: 0x380
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX15_SOFT_RESET_0

Offset: 0x384
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX15_STATUS_0

Offset: 0x38c
Read/Write: RO
Parity Protection: N
Reset: 0x0XXX00XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
25:16	X	**AVAILABLE_CREDITS**
7	X	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	X	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	X	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
4	X	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	X	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX15_INT_STATUS_0

Offset: 0x390
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX15_INT_MASK_0

Offset: 0x394
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_OVERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_RX15_INT_SET_0

Offset: 0x398
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX15_INT_CLEAR_0

Offset: 0x39c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX15_CIF_CTRL_0

Offset: 0x3a0
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PACK8_ENABLE**
30	RW	0x0	**PACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_RX15_FIFO_CTRL_0

Offset: 0x3a8
Read/Write: RW
Parity Protection: N
Reset: 0x0000022e (0b0xxx,xxxx,xxxx,xxxx,xx00,0010,xx10,1110)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x2e	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_RX15_FIFO_READ_0

Offset: 0x3ac
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

ADMAIF_AXBAR_RX16_ENABLE_0

Offset: 0x3c0
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX16_SOFT_RESET_0

Offset: 0x3c4
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX16_STATUS_0

Offset: 0x3cc
Read/Write: RO
Parity Protection: N
Reset: 0x0XXX00XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
25:16	X	**AVAILABLE_CREDITS**
7	X	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	X	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	X	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
4	X	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	X	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX16_INT_STATUS_0

Offset: 0x3d0
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX16_INT_MASK_0

Offset: 0x3d4
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_OVERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_RX16_INT_SET_0

Offset: 0x3d8
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX16_INT_CLEAR_0

Offset: 0x3dc
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX16_CIF_CTRL_0

Offset: 0x3e0
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PACK8_ENABLE**

Bit	R/W	Reset	Description
30	RW	0x0	**PACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32

Bit	R/W	Reset	Description
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_RX16_FIFO_CTRL_0

Offset: 0x3e8
Read/Write: RW
Parity Protection: N
Reset: 0x00000231 (0b0xxx,xxxx,xxxx,xxxx,xx00,0010,xx11,0001)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x31	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_RX16_FIFO_READ_0

Offset: 0x3ec
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

ADMAIF_AXBAR_RX17_ENABLE_0

Offset: 0x400
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX17_SOFT_RESET_0

Offset: 0x404
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX17_STATUS_0

Offset: 0x40c
Read/Write: RO
Parity Protection: N
Reset: 0x0XXX00XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
25:16	X	**AVAILABLE_CREDITS**
7	X	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	X	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	X	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
4	X	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	X	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX17_INT_STATUS_0

Offset: 0x410
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX17_INT_MASK_0

Offset: 0x414
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_OVERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_RX17_INT_SET_0

Offset: 0x418
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX17_INT_CLEAR_0

Offset: 0x41c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX17_CIF_CTRL_0

Offset: 0x420
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PACK8_ENABLE**
30	RW	0x0	**PACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_RX17_FIFO_CTRL_0

Offset: 0x428
Read/Write: RW
Parity Protection: N
Reset: 0x00000234 (0b0xxx,xxxx,xxxx,xxxx,xx00,0010,xx11,0100)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x34	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_RX17_FIFO_READ_0

Offset: 0x42c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

ADMAIF_AXBAR_RX18_ENABLE_0

Offset: 0x440
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX18_SOFT_RESET_0

Offset: 0x444
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX18_STATUS_0

Offset: 0x44c
Read/Write: RO
Parity Protection: N
Reset: 0x0XXX00XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
25:16	X	**AVAILABLE_CREDITS**
7	X	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	X	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	X	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
4	X	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	X	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX18_INT_STATUS_0

Offset: 0x450
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX18_INT_MASK_0

Offset: 0x454
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_OVERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_RX18_INT_SET_0

Offset: 0x458
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX18_INT_CLEAR_0

Offset: 0x45c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX18_CIF_CTRL_0

Offset: 0x460
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PACK8_ENABLE**

Bit	R/W	Reset	Description
30	RW	0x0	**PACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32

Bit	R/W	Reset	Description
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_RX18_FIFO_CTRL_0

Offset: 0x468
Read/Write: RW
Parity Protection: N
Reset: 0x00000237 (0b0xxx,xxxx,xxxx,xxxx,xx00,0010,xx11,0111)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x37	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_RX18_FIFO_READ_0

Offset: 0x46c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

ADMAIF_AXBAR_RX19_ENABLE_0

Offset: 0x480
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX19_SOFT_RESET_0

Offset: 0x484
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX19_STATUS_0

Offset: 0x48c
Read/Write: RO
Parity Protection: N
Reset: 0x0XXX00XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
25:16	X	**AVAILABLE_CREDITS**
7	X	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	X	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	X	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
4	X	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	X	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX19_INT_STATUS_0

Offset: 0x490
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX19_INT_MASK_0

Offset: 0x494
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_OVERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_RX19_INT_SET_0

Offset: 0x498
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX19_INT_CLEAR_0

Offset: 0x49c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX19_CIF_CTRL_0

Offset: 0x4a0
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PACK8_ENABLE**
30	RW	0x0	**PACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_RX19_FIFO_CTRL_0

Offset: 0x4a8
Read/Write: RW
Parity Protection: N
Reset: 0x0000023a (0b0xxx,xxxx,xxxx,xxxx,xx00,0010,xx11,1010)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x3a	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_RX19_FIFO_READ_0

Offset: 0x4ac
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

ADMAIF_AXBAR_RX20_ENABLE_0

Offset: 0x4c0
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX20_SOFT_RESET_0

Offset: 0x4c4
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX20_STATUS_0

Offset: 0x4cc
Read/Write: RO
Parity Protection: N
Reset: 0x0XXX00XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
25:16	X	**AVAILABLE_CREDITS**
7	X	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	X	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	X	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
4	X	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	X	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX20_INT_STATUS_0

Offset: 0x4d0
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX20_INT_MASK_0

Offset: 0x4d4
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_OVERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_RX20_INT_SET_0

Offset: 0x4d8
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX20_INT_CLEAR_0

Offset: 0x4dc
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_RX20_CIF_CTRL_0

Offset: 0x4e0
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PACK8_ENABLE**

Bit	R/W	Reset	Description
30	RW	0x0	**PACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32

Bit	R/W	Reset	Description
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_RX20_FIFO_CTRL_0

Offset: 0x4e8
Read/Write: RW
Parity Protection: N
Reset: 0x0000023d (0b0xxx,xxxx,xxxx,xxxx,xx00,0010,xx11,1101)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x3d	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_RX20_FIFO_READ_0

Offset: 0x4ec
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

ADMAIF_AXBAR_TX1_ENABLE_0

Offset: 0x500
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX1_SOFT_RESET_0

Offset: 0x504
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX1_STATUS_0

Offset: 0x50c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xx00,0000,0000,xxxx,xxxx,0000,xxx0)

Bit	Reset	Description
25:16	0x0	**AVAILABLE_CREDITS**
7	0x0	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	0x0	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	0x0	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
4	0x0	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX1_INT_STATUS_0

Offset: 0x510
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX1_INT_MASK_0

Offset: 0x514
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_UNDERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_TX1_INT_SET_0

Offset: 0x518
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX1_INT_CLEAR_0

Offset: 0x51c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX1_CIF_CTRL_0

Offset: 0x520
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**UNPACK8_ENABLE**
30	RW	0x0	**UNPACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_TX1_FIFO_CTRL_0

Offset: 0x528
Read/Write: RW
Parity Protection: N
Reset: 0x02000300 (0b0000,0010,0000,xxxx,xx00,0011,xx00,0000)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
30:20	0x20	**DMA_FIFO_THRESHOLD**
13:8	0x3	**DMA_FIFO_SIZE**
5:0	0x0	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_TX1_FIFO_WRITE_0

Offset: 0x52c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

ADMAIF_AXBAR_TX2_ENABLE_0

Offset: 0x540
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX2_SOFT_RESET_0

Offset: 0x544
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX2_STATUS_0

Offset: 0x54c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xx00,0000,0000,xxxx,xxxx,0000,xxx0)

Bit	Reset	Description
25:16	0x0	**AVAILABLE_CREDITS**
7	0x0	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	0x0	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	0x0	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
4	0x0	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX2_INT_STATUS_0

Offset: 0x550
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX2_INT_MASK_0

Offset: 0x554
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_UNDERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_TX2_INT_SET_0

Offset: 0x558
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX2_INT_CLEAR_0

Offset: 0x55c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX2_CIF_CTRL_0

Offset: 0x560
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**UNPACK8_ENABLE**

Bit	R/W	Reset	Description
30	RW	0x0	**UNPACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32

Bit	R/W	Reset	Description
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_TX2_FIFO_CTRL_0

Offset: 0x568
Read/Write: RW
Parity Protection: N
Reset: 0x02000304 (0b0000,0010,0000,xxxx,xx00,0011,xx00,0100)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
30:20	0x20	**DMA_FIFO_THRESHOLD**
13:8	0x3	**DMA_FIFO_SIZE**
5:0	0x4	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_TX2_FIFO_WRITE_0

Offset: 0x56c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

ADMAIF_AXBAR_TX3_ENABLE_0

Offset: 0x580
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX3_SOFT_RESET_0

Offset: 0x584
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX3_STATUS_0

Offset: 0x58c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xx00,0000,0000,xxxx,xxxx,0000,xxx0)

Bit	Reset	Description
25:16	0x0	**AVAILABLE_CREDITS**
7	0x0	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	0x0	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	0x0	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
4	0x0	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX3_INT_STATUS_0

Offset: 0x590
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX3_INT_MASK_0

Offset: 0x594
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_UNDERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_TX3_INT_SET_0

Offset: 0x598
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX3_INT_CLEAR_0

Offset: 0x59c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX3_CIF_CTRL_0

Offset: 0x5a0
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**UNPACK8_ENABLE**
30	RW	0x0	**UNPACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_TX3_FIFO_CTRL_0

Offset: 0x5a8
Read/Write: RW
Parity Protection: N
Reset: 0x02000308 (0b0000,0010,0000,xxxx,xx00,0011,xx00,1000)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
30:20	0x20	**DMA_FIFO_THRESHOLD**
13:8	0x3	**DMA_FIFO_SIZE**
5:0	0x8	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_TX3_FIFO_WRITE_0

Offset: 0x5ac
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

ADMAIF_AXBAR_TX4_ENABLE_0

Offset: 0x5c0
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX4_SOFT_RESET_0

Offset: 0x5c4
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX4_STATUS_0

Offset: 0x5cc
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xx00,0000,0000,xxxx,xxxx,0000,xxx0)

Bit	Reset	Description
25:16	0x0	**AVAILABLE_CREDITS**
7	0x0	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	0x0	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	0x0	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
4	0x0	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX4_INT_STATUS_0

Offset: 0x5d0
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX4_INT_MASK_0

Offset: 0x5d4
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_UNDERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_TX4_INT_SET_0

Offset: 0x5d8
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX4_INT_CLEAR_0

Offset: 0x5dc
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX4_CIF_CTRL_0

Offset: 0x5e0
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**UNPACK8_ENABLE**

Bit	R/W	Reset	Description
30	RW	0x0	**UNPACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32

Bit	R/W	Reset	Description
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_TX4_FIFO_CTRL_0

Offset: 0x5e8
Read/Write: RW
Parity Protection: N
Reset: 0x0200030c (0b0000,0010,0000,xxxx,xx00,0011,xx00,1100)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
30:20	0x20	**DMA_FIFO_THRESHOLD**
13:8	0x3	**DMA_FIFO_SIZE**
5:0	0xc	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_TX4_FIFO_WRITE_0

Offset: 0x5ec
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

ADMAIF_AXBAR_TX5_ENABLE_0

Offset: 0x600
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX5_SOFT_RESET_0

Offset: 0x604
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX5_STATUS_0

Offset: 0x60c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xx00,0000,0000,xxxx,xxxx,0000,xxx0)

Bit	Reset	Description
25:16	0x0	**AVAILABLE_CREDITS**
7	0x0	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	0x0	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	0x0	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
4	0x0	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX5_INT_STATUS_0

Offset: 0x610
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX5_INT_MASK_0

Offset: 0x614
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_UNDERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_TX5_INT_SET_0

Offset: 0x618
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX5_INT_CLEAR_0

Offset: 0x61c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX5_CIF_CTRL_0

Offset: 0x620
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**UNPACK8_ENABLE**
30	RW	0x0	**UNPACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_TX5_FIFO_CTRL_0

Offset: 0x628
Read/Write: RW
Parity Protection: N
Reset: 0x01800210 (0b0000,0001,1000,xxxx,xx00,0010,xx01,0000)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
30:20	0x18	**DMA_FIFO_THRESHOLD**
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x10	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_TX5_FIFO_WRITE_0

Offset: 0x62c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

ADMAIF_AXBAR_TX6_ENABLE_0

Offset: 0x640
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX6_SOFT_RESET_0

Offset: 0x644
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX6_STATUS_0

Offset: 0x64c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xx00,0000,0000,xxxx,xxxx,0000,xxx0)

Bit	Reset	Description
25:16	0x0	**AVAILABLE_CREDITS**
7	0x0	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	0x0	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	0x0	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
4	0x0	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX6_INT_STATUS_0

Offset: 0x650
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX6_INT_MASK_0

Offset: 0x654
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_UNDERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_TX6_INT_SET_0

Offset: 0x658
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX6_INT_CLEAR_0

Offset: 0x65c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX6_CIF_CTRL_0

Offset: 0x660
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**UNPACK8_ENABLE**

Bit	R/W	Reset	Description
30	RW	0x0	**UNPACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32

Bit	R/W	Reset	Description
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_TX6_FIFO_CTRL_0

Offset: 0x668
Read/Write: RW
Parity Protection: N
Reset: 0x01800213 (0b0000,0001,1000,xxxx,xx00,0010,xx01,0011)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
30:20	0x18	**DMA_FIFO_THRESHOLD**
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x13	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_TX6_FIFO_WRITE_0

Offset: 0x66c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

ADMAIF_AXBAR_TX7_ENABLE_0

Offset: 0x680
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX7_SOFT_RESET_0

Offset: 0x684
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX7_STATUS_0

Offset: 0x68c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xx00,0000,0000,xxxx,xxxx,0000,xxx0)

Bit	Reset	Description
25:16	0x0	**AVAILABLE_CREDITS**
7	0x0	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	0x0	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	0x0	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
4	0x0	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX7_INT_STATUS_0

Offset: 0x690
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX7_INT_MASK_0

Offset: 0x694
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_UNDERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_TX7_INT_SET_0

Offset: 0x698
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX7_INT_CLEAR_0

Offset: 0x69c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX7_CIF_CTRL_0

Offset: 0x6a0
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**UNPACK8_ENABLE**
30	RW	0x0	**UNPACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_TX7_FIFO_CTRL_0

Offset: 0x6a8
Read/Write: RW
Parity Protection: N
Reset: 0x01800216 (0b0000,0001,1000,xxxx,xx00,0010,xx01,0110)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
30:20	0x18	**DMA_FIFO_THRESHOLD**
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x16	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_TX7_FIFO_WRITE_0

Offset: 0x6ac
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

ADMAIF_AXBAR_TX8_ENABLE_0

Offset: 0x6c0
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX8_SOFT_RESET_0

Offset: 0x6c4
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX8_STATUS_0

Offset: 0x6cc
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xx00,0000,0000,xxxx,xxxx,0000,xxx0)

Bit	Reset	Description
25:16	0x0	**AVAILABLE_CREDITS**
7	0x0	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	0x0	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	0x0	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
4	0x0	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX8_INT_STATUS_0

Offset: 0x6d0
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX8_INT_MASK_0

Offset: 0x6d4
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_UNDERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_TX8_INT_SET_0

Offset: 0x6d8
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX8_INT_CLEAR_0

Offset: 0x6dc
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX8_CIF_CTRL_0

Offset: 0x6e0
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**UNPACK8_ENABLE**

Bit	R/W	Reset	Description
30	RW	0x0	**UNPACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32

Bit	R/W	Reset	Description
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_TX8_FIFO_CTRL_0

Offset: 0x6e8
Read/Write: RW
Parity Protection: N
Reset: 0x01800219 (0b0000,0001,1000,xxxx,xx00,0010,xx01,1001)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
30:20	0x18	**DMA_FIFO_THRESHOLD**
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x19	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_TX8_FIFO_WRITE_0

Offset: 0x6ec
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

ADMAIF_AXBAR_TX9_ENABLE_0

Offset: 0x700
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX9_SOFT_RESET_0

Offset: 0x704
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX9_STATUS_0

Offset: 0x70c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xx00,0000,0000,xxxx,xxxx,0000,xxx0)

Bit	Reset	Description
25:16	0x0	**AVAILABLE_CREDITS**
7	0x0	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	0x0	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	0x0	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
4	0x0	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX9_INT_STATUS_0

Offset: 0x710
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX9_INT_MASK_0

Offset: 0x714
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_UNDERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_TX9_INT_SET_0

Offset: 0x718
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX9_INT_CLEAR_0

Offset: 0x71c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX9_CIF_CTRL_0

Offset: 0x720
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**UNPACK8_ENABLE**
30	RW	0x0	**UNPACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_TX9_FIFO_CTRL_0

Offset: 0x728
Read/Write: RW
Parity Protection: N
Reset: 0x0180021c (0b0000,0001,1000,xxxx,xx00,0010,xx01,1100)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
30:20	0x18	**DMA_FIFO_THRESHOLD**
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x1c	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_TX9_FIFO_WRITE_0

Offset: 0x72c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

ADMAIF_AXBAR_TX10_ENABLE_0

Offset: 0x740
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX10_SOFT_RESET_0

Offset: 0x744
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX10_STATUS_0

Offset: 0x74c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xx00,0000,0000,xxxx,xxxx,0000,xxx0)

Bit	Reset	Description
25:16	0x0	**AVAILABLE_CREDITS**
7	0x0	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	0x0	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	0x0	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
4	0x0	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX10_INT_STATUS_0

Offset: 0x750
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX10_INT_MASK_0

Offset: 0x754
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_UNDERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_TX10_INT_SET_0

Offset: 0x758
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX10_INT_CLEAR_0

Offset: 0x75c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX10_CIF_CTRL_0

Offset: 0x760
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**UNPACK8_ENABLE**

Bit	R/W	Reset	Description
30	RW	0x0	**UNPACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32

Bit	R/W	Reset	Description
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_TX10_FIFO_CTRL_0

Offset: 0x768
Read/Write: RW
Parity Protection: N
Reset: 0x0180021f (0b0000,0001,1000,xxxx,xx00,0010,xx01,1111)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
30:20	0x18	**DMA_FIFO_THRESHOLD**
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x1f	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_TX10_FIFO_WRITE_0

Offset: 0x76c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

ADMAIF_AXBAR_TX11_ENABLE_0

Offset: 0x780
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX11_SOFT_RESET_0

Offset: 0x784
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX11_STATUS_0

Offset: 0x78c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xx00,0000,0000,xxxx,xxxx,0000,xxx0)

Bit	Reset	Description
25:16	0x0	**AVAILABLE_CREDITS**
7	0x0	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	0x0	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	0x0	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
4	0x0	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX11_INT_STATUS_0

Offset: 0x790
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX11_INT_MASK_0

Offset: 0x794
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_UNDERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_TX11_INT_SET_0

Offset: 0x798
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX11_INT_CLEAR_0

Offset: 0x79c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX11_CIF_CTRL_0

Offset: 0x7a0
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**UNPACK8_ENABLE**
30	RW	0x0	**UNPACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_TX11_FIFO_CTRL_0

Offset: 0x7a8
Read/Write: RW
Parity Protection: N
Reset: 0x01800222 (0b0000,0001,1000,xxxx,xx00,0010,xx10,0010)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
30:20	0x18	**DMA_FIFO_THRESHOLD**
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x22	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_TX11_FIFO_WRITE_0

Offset: 0x7ac
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

ADMAIF_AXBAR_TX12_ENABLE_0

Offset: 0x7c0
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX12_SOFT_RESET_0

Offset: 0x7c4
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX12_STATUS_0

Offset: 0x7cc
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xx00,0000,0000,xxxx,xxxx,0000,xxx0)

Bit	Reset	Description
25:16	0x0	**AVAILABLE_CREDITS**
7	0x0	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	0x0	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	0x0	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
4	0x0	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX12_INT_STATUS_0

Offset: 0x7d0
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX12_INT_MASK_0

Offset: 0x7d4
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_UNDERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_TX12_INT_SET_0

Offset: 0x7d8
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX12_INT_CLEAR_0

Offset: 0x7dc
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX12_CIF_CTRL_0

Offset: 0x7e0
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**UNPACK8_ENABLE**

Bit	R/W	Reset	Description
30	RW	0x0	**UNPACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32

Bit	R/W	Reset	Description
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_TX12_FIFO_CTRL_0

Offset: 0x7e8
Read/Write: RW
Parity Protection: N
Reset: 0x01800225 (0b0000,0001,1000,xxxx,xx00,0010,xx10,0101)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
30:20	0x18	**DMA_FIFO_THRESHOLD**
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x25	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_TX12_FIFO_WRITE_0

Offset: 0x7ec
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

ADMAIF_AXBAR_TX13_ENABLE_0

Offset: 0x800
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX13_SOFT_RESET_0

Offset: 0x804
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX13_STATUS_0

Offset: 0x80c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xx00,0000,0000,xxxx,xxxx,0000,xxx0)

Bit	Reset	Description
25:16	0x0	**AVAILABLE_CREDITS**
7	0x0	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	0x0	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	0x0	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
4	0x0	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX13_INT_STATUS_0

Offset: 0x810
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX13_INT_MASK_0

Offset: 0x814
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_UNDERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_TX13_INT_SET_0

Offset: 0x818
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX13_INT_CLEAR_0

Offset: 0x81c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX13_CIF_CTRL_0

Offset: 0x820
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**UNPACK8_ENABLE**
30	RW	0x0	**UNPACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_TX13_FIFO_CTRL_0

Offset: 0x828
Read/Write: RW
Parity Protection: N
Reset: 0x01800228 (0b0000,0001,1000,xxxx,xx00,0010,xx10,1000)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
30:20	0x18	**DMA_FIFO_THRESHOLD**
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x28	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_TX13_FIFO_WRITE_0

Offset: 0x82c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

ADMAIF_AXBAR_TX14_ENABLE_0

Offset: 0x840
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX14_SOFT_RESET_0

Offset: 0x844
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX14_STATUS_0

Offset: 0x84c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xx00,0000,0000,xxxx,xxxx,0000,xxx0)

Bit	Reset	Description
25:16	0x0	**AVAILABLE_CREDITS**
7	0x0	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	0x0	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	0x0	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
4	0x0	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX14_INT_STATUS_0

Offset: 0x850
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX14_INT_MASK_0

Offset: 0x854
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_UNDERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_TX14_INT_SET_0

Offset: 0x858
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX14_INT_CLEAR_0

Offset: 0x85c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX14_CIF_CTRL_0

Offset: 0x860
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**UNPACK8_ENABLE**

Bit	R/W	Reset	Description
30	RW	0x0	**UNPACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32

Bit	R/W	Reset	Description
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_TX14_FIFO_CTRL_0

Offset: 0x868
Read/Write: RW
Parity Protection: N
Reset: 0x0180022b (0b0000,0001,1000,xxxx,xx00,0010,xx10,1011)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
30:20	0x18	**DMA_FIFO_THRESHOLD**
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x2b	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_TX14_FIFO_WRITE_0

Offset: 0x86c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

ADMAIF_AXBAR_TX15_ENABLE_0

Offset: 0x880
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX15_SOFT_RESET_0

Offset: 0x884
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX15_STATUS_0

Offset: 0x88c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xx00,0000,0000,xxxx,xxxx,0000,xxx0)

Bit	Reset	Description
25:16	0x0	**AVAILABLE_CREDITS**
7	0x0	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	0x0	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	0x0	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
4	0x0	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX15_INT_STATUS_0

Offset: 0x890
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX15_INT_MASK_0

Offset: 0x894
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_UNDERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_TX15_INT_SET_0

Offset: 0x898
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX15_INT_CLEAR_0

Offset: 0x89c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX15_CIF_CTRL_0

Offset: 0x8a0
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**UNPACK8_ENABLE**
30	RW	0x0	**UNPACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_TX15_FIFO_CTRL_0

Offset: 0x8a8
Read/Write: RW
Parity Protection: N
Reset: 0x0180022e (0b0000,0001,1000,xxxx,xx00,0010,xx10,1110)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
30:20	0x18	**DMA_FIFO_THRESHOLD**
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x2e	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_TX15_FIFO_WRITE_0

Offset: 0x8ac
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

ADMAIF_AXBAR_TX16_ENABLE_0

Offset: 0x8c0
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX16_SOFT_RESET_0

Offset: 0x8c4
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX16_STATUS_0

Offset: 0x8cc
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xx00,0000,0000,xxxx,xxxx,0000,xxx0)

Bit	Reset	Description
25:16	0x0	**AVAILABLE_CREDITS**
7	0x0	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	0x0	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	0x0	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
4	0x0	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX16_INT_STATUS_0

Offset: 0x8d0
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX16_INT_MASK_0

Offset: 0x8d4
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_UNDERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_TX16_INT_SET_0

Offset: 0x8d8
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX16_INT_CLEAR_0

Offset: 0x8dc
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX16_CIF_CTRL_0

Offset: 0x8e0
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**UNPACK8_ENABLE**

Bit	R/W	Reset	Description
30	RW	0x0	**UNPACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32

Bit	R/W	Reset	Description
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_TX16_FIFO_CTRL_0

Offset: 0x8e8
Read/Write: RW
Parity Protection: N
Reset: 0x01800231 (0b0000,0001,1000,xxxx,xx00,0010,xx11,0001)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
30:20	0x18	**DMA_FIFO_THRESHOLD**
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x31	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_TX16_FIFO_WRITE_0

Offset: 0x8ec
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

ADMAIF_AXBAR_TX17_ENABLE_0

Offset: 0x900
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX17_SOFT_RESET_0

Offset: 0x904
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX17_STATUS_0

Offset: 0x90c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xx00,0000,0000,xxxx,xxxx,0000,xxx0)

Bit	Reset	Description
25:16	0x0	**AVAILABLE_CREDITS**
7	0x0	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	0x0	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	0x0	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
4	0x0	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX17_INT_STATUS_0

Offset: 0x910
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX17_INT_MASK_0

Offset: 0x914
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_UNDERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_TX17_INT_SET_0

Offset: 0x918
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX17_INT_CLEAR_0

Offset: 0x91c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX17_CIF_CTRL_0

Offset: 0x920
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**UNPACK8_ENABLE**
30	RW	0x0	**UNPACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_TX17_FIFO_CTRL_0

Offset: 0x928
Read/Write: RW
Parity Protection: N
Reset: 0x01800234 (0b0000,0001,1000,xxxx,xx00,0010,xx11,0100)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
30:20	0x18	**DMA_FIFO_THRESHOLD**
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x34	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_TX17_FIFO_WRITE_0

Offset: 0x92c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

ADMAIF_AXBAR_TX18_ENABLE_0

Offset: 0x940
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX18_SOFT_RESET_0

Offset: 0x944
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX18_STATUS_0

Offset: 0x94c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xx00,0000,0000,xxxx,xxxx,0000,xxx0)

Bit	Reset	Description
25:16	0x0	**AVAILABLE_CREDITS**
7	0x0	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	0x0	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	0x0	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
4	0x0	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX18_INT_STATUS_0

Offset: 0x950
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX18_INT_MASK_0

Offset: 0x954
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_UNDERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_TX18_INT_SET_0

Offset: 0x958
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX18_INT_CLEAR_0

Offset: 0x95c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX18_CIF_CTRL_0

Offset: 0x960
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**UNPACK8_ENABLE**

Bit	R/W	Reset	Description
30	RW	0x0	**UNPACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32

Bit	R/W	Reset	Description
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_TX18_FIFO_CTRL_0

Offset: 0x968
Read/Write: RW
Parity Protection: N
Reset: 0x01800237 (0b0000,0001,1000,xxxx,xx00,0010,xx11,0111)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
30:20	0x18	**DMA_FIFO_THRESHOLD**
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x37	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_TX18_FIFO_WRITE_0

Offset: 0x96c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

ADMAIF_AXBAR_TX19_ENABLE_0

Offset: 0x980
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX19_SOFT_RESET_0

Offset: 0x984
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX19_STATUS_0

Offset: 0x98c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xx00,0000,0000,xxxx,xxxx,0000,xxx0)

Bit	Reset	Description
25:16	0x0	**AVAILABLE_CREDITS**
7	0x0	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	0x0	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	0x0	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
4	0x0	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX19_INT_STATUS_0

Offset: 0x990
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX19_INT_MASK_0

Offset: 0x994
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_UNDERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_TX19_INT_SET_0

Offset: 0x998
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX19_INT_CLEAR_0

Offset: 0x99c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX19_CIF_CTRL_0

Offset: 0x9a0
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**UNPACK8_ENABLE**
30	RW	0x0	**UNPACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16

Bit	R/W	Reset	Description
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_TX19_FIFO_CTRL_0

Offset: 0x9a8
Read/Write: RW
Parity Protection: N
Reset: 0x0180023a (0b0000,0001,1000,xxxx,xx00,0010,xx11,1010)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
30:20	0x18	**DMA_FIFO_THRESHOLD**
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x3a	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_TX19_FIFO_WRITE_0

Offset: 0x9ac
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

ADMAIF_AXBAR_TX20_ENABLE_0

Offset: 0x9c0
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX20_SOFT_RESET_0

Offset: 0x9c4
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX20_STATUS_0

Offset: 0x9cc
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xx00,0000,0000,xxxx,xxxx,0000,xxx0)

Bit	Reset	Description
25:16	0x0	**AVAILABLE_CREDITS**
7	0x0	**DMA_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	0x0	**DMA_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
5	0x0	**ACIF_FIFO_FULL**: **0** = FALSE **1** = TRUE
4	0x0	**ACIF_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX20_INT_STATUS_0

Offset: 0x9d0
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX20_INT_MASK_0

Offset: 0x9d4
Read/Write: RW
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	MASK	**DMA_UNDERRUN**: **0** = UNMASK **1** = MASK

ADMAIF_AXBAR_TX20_INT_SET_0

Offset: 0x9d8
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX20_INT_CLEAR_0

Offset: 0x9dc
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_AXBAR_TX20_CIF_CTRL_0

Offset: 0x9e0
Read/Write: See table below
Parity Protection: N
Reset: 0x00007700 (0b0000,0000,0000,0000,x111,x111,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**UNPACK8_ENABLE**

Bit	R/W	Reset	Description
30	RW	0x0	**UNPACK16_ENABLE**
29:24	RO	0x0	**FIFO_THRESHOLD**
23:20	RW	0x0	**AXBAR_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
19:16	RW	0x0	**CLIENT_CHANNELS**: **0** = CH1 **1** = CH2 **2** = CH3 **3** = CH4 **4** = CH5 **5** = CH6 **6** = CH7 **7** = CH8 **8** = CH9 **9** = CH10 **10** = CH11 **11** = CH12 **12** = CH13 **13** = CH14 **14** = CH15 **15** = CH16
14:12	RW	0x7	**AXBAR_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32
10:8	RW	0x7	**CLIENT_BITS**: **0** = RVSD **1** = BIT8 **2** = BIT12 **3** = BIT16 **4** = BIT20 **5** = BIT24 **6** = BIT28 **7** = BIT32

Bit	R/W	Reset	Description
7:6	RW	0x0	**EXPAND**: **0** = ZERO **1** = ONE **2** = LFSR **3** = RSVD
5:4	RW	0x0	**STEREO_CONV**: **0** = CH0 **1** = CH1 **2** = AVG **3** = RSVD
3:2	RW	0x0	**FIFO_SIZE_DOWNSHIFT**
1	RW	0x0	**TRUNCATE**: **0** = ROUND **1** = CHOP
0	RW	0x0	**MONO_CONV**: **0** = ZERO **1** = COPY

ADMAIF_AXBAR_TX20_FIFO_CTRL_0

Offset: 0x9e8
Read/Write: RW
Parity Protection: N
Reset: 0x0180023d (0b0000,0001,1000,xxxx,xx00,0010,xx11,1101)

Bit	Reset	Description
31	0x0	**TRANSFER_MODE**: **0** = DMA **1** = PIO
30:20	0x18	**DMA_FIFO_THRESHOLD**
13:8	0x2	**DMA_FIFO_SIZE**
5:0	0x3d	**DMA_FIFO_START_ADDR**

ADMAIF_AXBAR_TX20_FIFO_WRITE_0

Offset: 0x9ec
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

ADMAIF_GLOBAL_ENABLE_0

Offset: 0xd00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_GLOBAL_SOFT_RESET_0

```
Self clearing Software initiated global reset.
It clears all channels enable bits, DMA/CIF FIFOs, state machines and any pipelines.
```

Offset: 0xd04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: **0** = FALSE **1** = TRUE

ADMAIF_GLOBAL_CG_0

Offset: 0xd08
Read/Write: RW
Parity Protection: N
Reset: 0x00000003 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11)

Bit	Reset	Description
1	0x1	**CHANNEL_SLCG_ENABLE**: Second level clock gating enable, for remaining channels specific logic portions. **0** = FALSE **1** = TRUE
0	0x1	**GLOBAL_SLCG_ENABLE**: Second level clock gating enable,for first 2 channels and global/common logic. **0** = FALSE **1** = TRUE

ADMAIF_GLOBAL_PIO_MODE_ENABLE_0

```
TZ protected
when '1' PIO mode is enabled; when '0' PIO mode is disabled
```

Offset: 0xd0c
Read/Write: RW
Parity Protection: N
Secure: Trust Zone Protected
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMAIF_GLOBAL_STATUS_0

Offset: 0xd10
Read/Write: RO
Parity Protection: N
Reset: 0x00000X0X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
9	X	**CHANNEL_CLK_ENABLE**: Clock Enable **0** = FALSE **1** = TRUE
8	X	**GLOBAL_CLK_ENABLE**: Clock Enable **0** = FALSE **1** = TRUE
0	X	**TRANSFER_ENABLED**: Asserts when at least one of the channels is enabled by the Software **0** = FALSE **1** = TRUE

ADMAIF_GLOBAL_RX_ENABLE_STATUS_0

Offset: 0xd20
Read/Write: RO
Parity Protection: N
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
19	X	**CH20_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
18	X	**CH19_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
17	X	**CH18_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
16	X	**CH17_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
15	X	**CH16_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
14	X	**CH15_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
13	X	**CH14_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
12	X	**CH13_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
11	X	**CH12_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
10	X	**CH11_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
9	X	**CH10_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
8	X	**CH9_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
7	X	**CH8_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
6	X	**CH7_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
5	X	**CH6_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
4	X	**CH5_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
3	X	**CH4_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
2	X	**CH3_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
1	X	**CH2_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
0	X	**CH1_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_GLOBAL_TX_ENABLE_STATUS_0

Offset: 0xd24
Read/Write: RO
Parity Protection: N
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
19	X	**CH20_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
18	X	**CH19_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
17	X	**CH18_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
16	X	**CH17_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
15	X	**CH16_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
14	X	**CH15_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
13	X	**CH14_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
12	X	**CH13_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
11	X	**CH12_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
10	X	**CH11_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
9	X	**CH10_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
8	X	**CH9_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
7	X	**CH8_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
6	X	**CH7_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
5	X	**CH6_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
4	X	**CH5_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
3	X	**CH4_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
2	X	**CH3_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
1	X	**CH2_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE
0	X	**CH1_TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMAIF_GLOBAL_RX_DMA_FIFO_EMPTY_STATUS_0

Offset: 0xd28
Read/Write: RO

Parity Protection: N
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
19	X	**CH20_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
18	X	**CH19_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
17	X	**CH18_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
16	X	**CH17_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
15	X	**CH16_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
14	X	**CH15_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
13	X	**CH14_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
12	X	**CH13_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
11	X	**CH12_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
10	X	**CH11_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
9	X	**CH10_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
8	X	**CH9_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
7	X	**CH8_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
6	X	**CH7_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
5	X	**CH6_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
4	X	**CH5_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
3	X	**CH4_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
2	X	**CH3_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
1	X	**CH2_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	X	**CH1_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

ADMAIF_GLOBAL_RX_DMA_FIFO_FULL_STATUS_0

Offset: 0xd2c
Read/Write: RO
Parity Protection: N
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
19	X	**CH20_FIFO_FULL**: **0** = FALSE **1** = TRUE
18	X	**CH19_FIFO_FULL**: **0** = FALSE **1** = TRUE
17	X	**CH18_FIFO_FULL**: **0** = FALSE **1** = TRUE
16	X	**CH17_FIFO_FULL**: **0** = FALSE **1** = TRUE
15	X	**CH16_FIFO_FULL**: **0** = FALSE **1** = TRUE
14	X	**CH15_FIFO_FULL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
13	X	**CH14_FIFO_FULL**: **0** = FALSE **1** = TRUE
12	X	**CH13_FIFO_FULL**: **0** = FALSE **1** = TRUE
11	X	**CH12_FIFO_FULL**: **0** = FALSE **1** = TRUE
10	X	**CH11_FIFO_FULL**: **0** = FALSE **1** = TRUE
9	X	**CH10_FIFO_FULL**: **0** = FALSE **1** = TRUE
8	X	**CH9_FIFO_FULL**: **0** = FALSE **1** = TRUE
7	X	**CH8_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	X	**CH7_FIFO_FULL**: **0** = FALSE **1** = TRUE
5	X	**CH6_FIFO_FULL**: **0** = FALSE **1** = TRUE
4	X	**CH5_FIFO_FULL**: **0** = FALSE **1** = TRUE
3	X	**CH4_FIFO_FULL**: **0** = FALSE **1** = TRUE
2	X	**CH3_FIFO_FULL**: **0** = FALSE **1** = TRUE
1	X	**CH2_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	X	**CH1_FIFO_FULL**: **0** = FALSE **1** = TRUE

ADMAIF_GLOBAL_TX_DMA_FIFO_EMPTY_STATUS_0

Offset: 0xd30
Read/Write: RO
Parity Protection: N
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
19	X	**CH20_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
18	X	**CH19_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
17	X	**CH18_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
16	X	**CH17_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
15	X	**CH16_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
14	X	**CH15_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
13	X	**CH14_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
12	X	**CH13_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
11	X	**CH12_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
10	X	**CH11_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
9	X	**CH10_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
8	X	**CH9_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
7	X	**CH8_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
6	X	**CH7_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
5	X	**CH6_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
4	X	**CH5_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
3	X	**CH4_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
2	X	**CH3_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
1	X	**CH2_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	X	**CH1_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

ADMAIF_GLOBAL_TX_DMA_FIFO_FULL_STATUS_0

Offset: 0xd34
Read/Write: RO
Parity Protection: N
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
19	X	**CH20_FIFO_FULL**: **0** = FALSE **1** = TRUE
18	X	**CH19_FIFO_FULL**: **0** = FALSE **1** = TRUE
17	X	**CH18_FIFO_FULL**: **0** = FALSE **1** = TRUE
16	X	**CH17_FIFO_FULL**: **0** = FALSE **1** = TRUE
15	X	**CH16_FIFO_FULL**: **0** = FALSE **1** = TRUE
14	X	**CH15_FIFO_FULL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
13	X	**CH14_FIFO_FULL**: **0** = FALSE **1** = TRUE
12	X	**CH13_FIFO_FULL**: **0** = FALSE **1** = TRUE
11	X	**CH12_FIFO_FULL**: **0** = FALSE **1** = TRUE
10	X	**CH11_FIFO_FULL**: **0** = FALSE **1** = TRUE
9	X	**CH10_FIFO_FULL**: **0** = FALSE **1** = TRUE
8	X	**CH9_FIFO_FULL**: **0** = FALSE **1** = TRUE
7	X	**CH8_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	X	**CH7_FIFO_FULL**: **0** = FALSE **1** = TRUE
5	X	**CH6_FIFO_FULL**: **0** = FALSE **1** = TRUE
4	X	**CH5_FIFO_FULL**: **0** = FALSE **1** = TRUE
3	X	**CH4_FIFO_FULL**: **0** = FALSE **1** = TRUE
2	X	**CH3_FIFO_FULL**: **0** = FALSE **1** = TRUE
1	X	**CH2_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	X	**CH1_FIFO_FULL**: **0** = FALSE **1** = TRUE

ADMAIF_GLOBAL_RX_ACIF_FIFO_EMPTY_STATUS_0

Offset: 0xd38
Read/Write: RO
Parity Protection: N
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
19	X	**CH20_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
18	X	**CH19_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
17	X	**CH18_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
16	X	**CH17_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
15	X	**CH16_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
14	X	**CH15_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
13	X	**CH14_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
12	X	**CH13_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
11	X	**CH12_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
10	X	**CH11_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
9	X	**CH10_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
8	X	**CH9_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
7	X	**CH8_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
6	X	**CH7_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
5	X	**CH6_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
4	X	**CH5_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
3	X	**CH4_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
2	X	**CH3_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
1	X	**CH2_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	X	**CH1_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

ADMAIF_GLOBAL_RX_ACIF_FIFO_FULL_STATUS_0

Offset: 0xd3c
Read/Write: RO
Parity Protection: N
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
19	X	**CH20_FIFO_FULL**: **0** = FALSE **1** = TRUE
18	X	**CH19_FIFO_FULL**: **0** = FALSE **1** = TRUE
17	X	**CH18_FIFO_FULL**: **0** = FALSE **1** = TRUE
16	X	**CH17_FIFO_FULL**: **0** = FALSE **1** = TRUE
15	X	**CH16_FIFO_FULL**: **0** = FALSE **1** = TRUE
14	X	**CH15_FIFO_FULL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
13	X	**CH14_FIFO_FULL**: **0** = FALSE **1** = TRUE
12	X	**CH13_FIFO_FULL**: **0** = FALSE **1** = TRUE
11	X	**CH12_FIFO_FULL**: **0** = FALSE **1** = TRUE
10	X	**CH11_FIFO_FULL**: **0** = FALSE **1** = TRUE
9	X	**CH10_FIFO_FULL**: **0** = FALSE **1** = TRUE
8	X	**CH9_FIFO_FULL**: **0** = FALSE **1** = TRUE
7	X	**CH8_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	X	**CH7_FIFO_FULL**: **0** = FALSE **1** = TRUE
5	X	**CH6_FIFO_FULL**: **0** = FALSE **1** = TRUE
4	X	**CH5_FIFO_FULL**: **0** = FALSE **1** = TRUE
3	X	**CH4_FIFO_FULL**: **0** = FALSE **1** = TRUE
2	X	**CH3_FIFO_FULL**: **0** = FALSE **1** = TRUE
1	X	**CH2_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	X	**CH1_FIFO_FULL**: **0** = FALSE **1** = TRUE

ADMAIF_GLOBAL_TX_ACIF_FIFO_EMPTY_STATUS_0

Offset: 0xd40
Read/Write: RO
Parity Protection: N
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
19	X	**CH20_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
18	X	**CH19_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
17	X	**CH18_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
16	X	**CH17_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
15	X	**CH16_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
14	X	**CH15_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
13	X	**CH14_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
12	X	**CH13_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
11	X	**CH12_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
10	X	**CH11_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
9	X	**CH10_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
8	X	**CH9_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
7	X	**CH8_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
6	X	**CH7_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
5	X	**CH6_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
4	X	**CH5_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
3	X	**CH4_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
2	X	**CH3_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
1	X	**CH2_FIFO_EMPTY**: **0** = FALSE **1** = TRUE
0	X	**CH1_FIFO_EMPTY**: **0** = FALSE **1** = TRUE

ADMAIF_GLOBAL_TX_ACIF_FIFO_FULL_STATUS_0

Offset: 0xd44
Read/Write: RO
Parity Protection: N
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
19	X	**CH20_FIFO_FULL**: **0** = FALSE **1** = TRUE
18	X	**CH19_FIFO_FULL**: **0** = FALSE **1** = TRUE
17	X	**CH18_FIFO_FULL**: **0** = FALSE **1** = TRUE
16	X	**CH17_FIFO_FULL**: **0** = FALSE **1** = TRUE
15	X	**CH16_FIFO_FULL**: **0** = FALSE **1** = TRUE
14	X	**CH15_FIFO_FULL**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
13	X	**CH14_FIFO_FULL**: **0** = FALSE **1** = TRUE
12	X	**CH13_FIFO_FULL**: **0** = FALSE **1** = TRUE
11	X	**CH12_FIFO_FULL**: **0** = FALSE **1** = TRUE
10	X	**CH11_FIFO_FULL**: **0** = FALSE **1** = TRUE
9	X	**CH10_FIFO_FULL**: **0** = FALSE **1** = TRUE
8	X	**CH9_FIFO_FULL**: **0** = FALSE **1** = TRUE
7	X	**CH8_FIFO_FULL**: **0** = FALSE **1** = TRUE
6	X	**CH7_FIFO_FULL**: **0** = FALSE **1** = TRUE
5	X	**CH6_FIFO_FULL**: **0** = FALSE **1** = TRUE
4	X	**CH5_FIFO_FULL**: **0** = FALSE **1** = TRUE
3	X	**CH4_FIFO_FULL**: **0** = FALSE **1** = TRUE
2	X	**CH3_FIFO_FULL**: **0** = FALSE **1** = TRUE
1	X	**CH2_FIFO_FULL**: **0** = FALSE **1** = TRUE
0	X	**CH1_FIFO_FULL**: **0** = FALSE **1** = TRUE

ADMAIF_GLOBAL_RX_DMA_OVERRUN_INT_STATUS_0

Offset: 0xd48
Read/Write: RO
Parity Protection: N
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
19	X	**CH20_DMA_OVERRUN**: **0** = FALSE **1** = TRUE
18	X	**CH19_DMA_OVERRUN**: **0** = FALSE **1** = TRUE
17	X	**CH18_DMA_OVERRUN**: **0** = FALSE **1** = TRUE
16	X	**CH17_DMA_OVERRUN**: **0** = FALSE **1** = TRUE
15	X	**CH16_DMA_OVERRUN**: **0** = FALSE **1** = TRUE
14	X	**CH15_DMA_OVERRUN**: **0** = FALSE **1** = TRUE
13	X	**CH14_DMA_OVERRUN**: **0** = FALSE **1** = TRUE
12	X	**CH13_DMA_OVERRUN**: **0** = FALSE **1** = TRUE
11	X	**CH12_DMA_OVERRUN**: **0** = FALSE **1** = TRUE
10	X	**CH11_DMA_OVERRUN**: **0** = FALSE **1** = TRUE
9	X	**CH10_DMA_OVERRUN**: **0** = FALSE **1** = TRUE
8	X	**CH9_DMA_OVERRUN**: **0** = FALSE **1** = TRUE
7	X	**CH8_DMA_OVERRUN**: **0** = FALSE **1** = TRUE
6	X	**CH7_DMA_OVERRUN**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
5	X	**CH6_DMA_OVERRUN**: **0** = FALSE **1** = TRUE
4	X	**CH5_DMA_OVERRUN**: **0** = FALSE **1** = TRUE
3	X	**CH4_DMA_OVERRUN**: **0** = FALSE **1** = TRUE
2	X	**CH3_DMA_OVERRUN**: **0** = FALSE **1** = TRUE
1	X	**CH2_DMA_OVERRUN**: **0** = FALSE **1** = TRUE
0	X	**CH1_DMA_OVERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_GLOBAL_TX_DMA_UNDERRUN_INT_STATUS_0

Offset: 0xd4c
Read/Write: RO
Parity Protection: N
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
19	X	**CH20_DMA_UNDERRUN**: **0** = FALSE **1** = TRUE
18	X	**CH19_DMA_UNDERRUN**: **0** = FALSE **1** = TRUE
17	X	**CH18_DMA_UNDERRUN**: **0** = FALSE **1** = TRUE
16	X	**CH17_DMA_UNDERRUN**: **0** = FALSE **1** = TRUE
15	X	**CH16_DMA_UNDERRUN**: **0** = FALSE **1** = TRUE
14	X	**CH15_DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
13	X	**CH14_DMA_UNDERRUN**: **0** = FALSE **1** = TRUE
12	X	**CH13_DMA_UNDERRUN**: **0** = FALSE **1** = TRUE
11	X	**CH12_DMA_UNDERRUN**: **0** = FALSE **1** = TRUE
10	X	**CH11_DMA_UNDERRUN**: **0** = FALSE **1** = TRUE
9	X	**CH10_DMA_UNDERRUN**: **0** = FALSE **1** = TRUE
8	X	**CH9_DMA_UNDERRUN**: **0** = FALSE **1** = TRUE
7	X	**CH8_DMA_UNDERRUN**: **0** = FALSE **1** = TRUE
6	X	**CH7_DMA_UNDERRUN**: **0** = FALSE **1** = TRUE
5	X	**CH6_DMA_UNDERRUN**: **0** = FALSE **1** = TRUE
4	X	**CH5_DMA_UNDERRUN**: **0** = FALSE **1** = TRUE
3	X	**CH4_DMA_UNDERRUN**: **0** = FALSE **1** = TRUE
2	X	**CH3_DMA_UNDERRUN**: **0** = FALSE **1** = TRUE
1	X	**CH2_DMA_UNDERRUN**: **0** = FALSE **1** = TRUE
0	X	**CH1_DMA_UNDERRUN**: **0** = FALSE **1** = TRUE

ADMAIF_GLOBAL_CYA_0

This is an array of two identical register entries; the register fields below apply to each entry.
Offset: 0xd50,..,0xd54
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**Reserved**

ADMAIF_GLOBAL_DBG_0

This is an array of two identical register entries; the register fields below apply to each entry.
Offset: 0xd58,..,0xd5c
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

7.7.4.18 Audio Direct Memory Access (ADMA) Control Registers

ADMA_GLOBAL_CMD_0

Offset: 0x0
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**GLOBAL_ENABLE**: This bit must be set to "1" for enabling the DMA transfers from channels. **0** = FALSE **1** = TRUE

ADMA_GLOBAL_SOFT_RESET_0

Offset: 0x4
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: Self clearing soft reset. **0** = FALSE **1** = TRUE

ADMA_GLOBAL_CG_0

Offset: 0x8
Read/Write: RW
Parity Protection: N
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**CFG_SLCG_ENABLE**: Second level clock gating enable for the clock driving the registers. **0** = FALSE **1** = TRUE
1	0x1	**CHANNEL_SLCG_ENABLE**: Second level clock gating enable for the clock driving the channel controllers except of first 4 channels. **0** = FALSE **1** = TRUE
0	0x1	**GLOBAL_SLCG_ENABLE**: Second level clock gating enable for the clock driving first 4 channel controllers and remaining modules. **0** = FALSE **1** = TRUE

ADMA_GLOBAL_STATUS_0

Offset: 0xc
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**TRANSFER_PAUSED**: Transfers from ALL channels can be paused by setting this bit and can be resumed by clearing it. **0** = FALSE **1** = TRUE
3	0x0	**CFG_CLK_ENABLED**: Indicates whether Registers clock is enabled or disable. **0** = FALSE **1** = TRUE

Bit	Reset	Description
2	0x0	**CHANNEL_CLK_ENABLED**: Indicates whether the clock driving channel controllers is enabled or disabled. **0** = FALSE **1** = TRUE
1	0x0	**GLOBAL_CLK_ENABLED**: Indicates whether the clock driving common modules is enabled or disabled. **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: Asserts when any of the channels is enabled for DMA transfer. **0** = FALSE **1** = TRUE

ADMA_GLOBAL_CTRL_0

Offset: 0x20
Read/Write: RW
Parity Protection: N
Reset: 0x00080800 (0bxxxx,xxxx,xxxx,1000,xxxx,1000,xxxx,xxx0)

Bit	Reset	Description
19:16	0x8	**OUTSTANDING_MEM_WRITES**: Supports a max of 8 oustanding memory reads to ARAM/DRAM.
11:8	0x8	**OUTSTANDING_MEM_READS**: Supports a max of 8 outstanding memory writes to ARAM/DRAM.
0	0x0	**TRANSFER_PAUSE**: S/W may choose to PASUE Transfers from all channels by setting this bit to "1". Clearing it resume all transfers. **0** = DISABLE **1** = ENABLE

ADMA_GLOBAL_REGION_ID_LOCK_0

Offset: 0x2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	0x0	**LOCK**

ADMA_GLOBAL_PAGE1_CHGRP_0

Offset: 0x30
Read/Write: RW
Parity Protection: N
Reset: 0x000000ff (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1111,1111)

Bit	Reset	Description
7:0	0xff	**ENABLE**

ADMA_GLOBAL_PAGE2_CHGRP_0

Offset: 0x34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**ENABLE**

ADMA_GLOBAL_PAGE3_CHGRP_0

Offset: 0x38
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**ENABLE**

ADMA_GLOBAL_PAGE4_CHGRP_0

Offset: 0x3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**ENABLE**

ADMA_GLOBAL_PAGE1_ARAM_CFG_0

Offset: 0x50
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ARAM_DISABLE**: **0** = FALSE **1** = TRUE

ADMA_GLOBAL_PAGE2_ARAM_CFG_0

Offset: 0x54
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ARAM_DISABLE**: **0** = FALSE **1** = TRUE

ADMA_GLOBAL_PAGE3_ARAM_CFG_0

Offset: 0x58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ARAM_DISABLE**: **0** = FALSE **1** = TRUE

ADMA_GLOBAL_PAGE4_ARAM_CFG_0

Offset: 0x5c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ARAM_DISABLE**: **0** = FALSE **1** = TRUE

ADMA_GLOBAL_PAGE1_RX_REQUESTORS_0

Offset: 0x70
Read/Write: RW
Parity Protection: N
Reset: 0x01ffffff (0bxxxx,xxx1,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
24:0	0x1ffffff	**ENABLE**

ADMA_GLOBAL_PAGE2_RX_REQUESTORS_0

Offset: 0x74
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
24:0	0x0	**ENABLE**

ADMA_GLOBAL_PAGE3_RX_REQUESTORS_0

Offset: 0x78
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
24:0	0x0	**ENABLE**

ADMA_GLOBAL_PAGE4_RX_REQUESTORS_0

Offset: 0x7c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
24:0	0x0	**ENABLE**

ADMA_GLOBAL_PAGE1_TX_REQUESTORS_0

Offset: 0x84
Read/Write: RW
Parity Protection: N
Reset: 0x00ffffff (0bxxxx,xxxx,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
23:0	0xffffff	**ENABLE**

ADMA_GLOBAL_PAGE2_TX_REQUESTORS_0

Offset: 0x88
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:0	0x0	**ENABLE**

ADMA_GLOBAL_PAGE3_TX_REQUESTORS_0

Offset: 0x8c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:0	0x0	**ENABLE**

ADMA_GLOBAL_PAGE4_TX_REQUESTORS_0

Offset: 0x90
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:0	0x0	**ENABLE**

ADMA_GLOBAL_CH1_REGION_ID_0

Offset: 0x98
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH2_REGION_ID_0

Offset: 0x9c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH3_REGION_ID_0

Offset: 0xa0
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH4_REGION_ID_0

Offset: 0xa4
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH5_REGION_ID_0

Offset: 0xa8
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH6_REGION_ID_0

Offset: 0xac
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH7_REGION_ID_0

Offset: 0xb0
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH8_REGION_ID_0

Offset: 0xb4
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH9_REGION_ID_0

Offset: 0xb8
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH10_REGION_ID_0

Offset: 0xbc
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH11_REGION_ID_0

Offset: 0xc0
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH12_REGION_ID_0

Offset: 0xc4
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH13_REGION_ID_0

Offset: 0xc8
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH14_REGION_ID_0

Offset: 0xcc
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH15_REGION_ID_0

Offset: 0xd0
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH16_REGION_ID_0

Offset: 0xd4
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH17_REGION_ID_0

Offset: 0xd8
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH18_REGION_ID_0

Offset: 0xdc
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH19_REGION_ID_0

Offset: 0xe0
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH20_REGION_ID_0

Offset: 0xe4
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH21_REGION_ID_0

Offset: 0xe8
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH22_REGION_ID_0

Offset: 0xec
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH23_REGION_ID_0

Offset: 0xf0
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH24_REGION_ID_0

Offset: 0xf4
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH25_REGION_ID_0

Offset: 0xf8
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH26_REGION_ID_0

Offset: 0xfc
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH27_REGION_ID_0

Offset: 0x100
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH28_REGION_ID_0

Offset: 0x104
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH29_REGION_ID_0

Offset: 0x108
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH30_REGION_ID_0

Offset: 0x10c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH31_REGION_ID_0

Offset: 0x110
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_GLOBAL_CH32_REGION_ID_0

Offset: 0x114
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**REGION_ID**

ADMA_PAGE1_CH1_CMD_0

Offset: 0x10000
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH1_SOFT_RESET_0

Offset: 0x10004
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH1_STATUS_0

Offset: 0x1000c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH1_INT_STATUS_0

Offset: 0x10010
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH1_INT_SET_0

Offset: 0x10018
Read/Write: RW

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH1_INT_CLEAR_0

Offset: 0x1001c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH1_CTRL_0

Offset: 0x10024
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH1_CONFIG_0

Offset: 0x10028
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH1_AHUB_FIFO_CTRL_0

Offset: 0x1002c
Read/Write: RW
Parity Protection: N
Reset: 0x01010303 (0b0x00,0001,xx00,0001,xx00,0011,xx00,0011)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x1	**OVERFLOW_THRESHOLD**
21:16	0x1	**STARVATION_THRESHOLD**
13:8	0x3	**TX_FIFO_SIZE**
5:0	0x3	**RX_FIFO_SIZE**

ADMA_PAGE1_CH1_TC_STATUS_0

Offset: 0x10030
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH1_LOWER_SOURCE_ADDR_0

Offset: 0x10034
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH1_LOWER_TARGET_ADDR_0

Offset: 0x1003c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH1_TC_0

Offset: 0x10044
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH1_LOWER_DESC_ADDR_0

Offset: 0x10048
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH1_TRANSFER_STATUS_0

Offset: 0x10054
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH1_TIMESTAMP_ADDR_0

Offset: 0x10058
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH1_TIMESTAMP_STATUS_0

Offset: 0x1005c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH1_APR_0

Offset: 0x10060
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH2_CMD_0

Offset: 0x10100
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH2_SOFT_RESET_0

Offset: 0x10104
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH2_STATUS_0

Offset: 0x1010c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH2_INT_STATUS_0

Offset: 0x10110
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH2_INT_SET_0

Offset: 0x10118
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH2_INT_CLEAR_0

Offset: 0x1011c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH2_CTRL_0

Offset: 0x10124
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH2_CONFIG_0

Offset: 0x10128
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH2_AHUB_FIFO_CTRL_0

Offset: 0x1012c
Read/Write: RW
Parity Protection: N
Reset: 0x01010303 (0b0x00,0001,xx00,0001,xx00,0011,xx00,0011)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x1	**OVERFLOW_THRESHOLD**
21:16	0x1	**STARVATION_THRESHOLD**
13:8	0x3	**TX_FIFO_SIZE**
5:0	0x3	**RX_FIFO_SIZE**

ADMA_PAGE1_CH2_TC_STATUS_0

Offset: 0x10130
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH2_LOWER_SOURCE_ADDR_0

Offset: 0x10134
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH2_LOWER_TARGET_ADDR_0

Offset: 0x1013c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH2_TC_0

Offset: 0x10144
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH2_LOWER_DESC_ADDR_0

Offset: 0x10148
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH2_TRANSFER_STATUS_0

Offset: 0x10154
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH2_TIMESTAMP_ADDR_0

Offset: 0x10158
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH2_TIMESTAMP_STATUS_0

Offset: 0x1015c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH2_APR_0

Offset: 0x10160
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH3_CMD_0

Offset: 0x10200
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH3_SOFT_RESET_0

Offset: 0x10204
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH3_STATUS_0

Offset: 0x1020c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH3_INT_STATUS_0

Offset: 0x10210
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH3_INT_SET_0

Offset: 0x10218
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH3_INT_CLEAR_0

Offset: 0x1021c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH3_CTRL_0

Offset: 0x10224
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH3_CONFIG_0

Offset: 0x10228
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH3_AHUB_FIFO_CTRL_0

Offset: 0x1022c
Read/Write: RW
Parity Protection: N
Reset: 0x01010303 (0b0x00,0001,xx00,0001,xx00,0011,xx00,0011)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x1	**OVERFLOW_THRESHOLD**
21:16	0x1	**STARVATION_THRESHOLD**
13:8	0x3	**TX_FIFO_SIZE**
5:0	0x3	**RX_FIFO_SIZE**

ADMA_PAGE1_CH3_TC_STATUS_0

Offset: 0x10230
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH3_LOWER_SOURCE_ADDR_0

Offset: 0x10234
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH3_LOWER_TARGET_ADDR_0

Offset: 0x1023c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH3_TC_0

Offset: 0x10244
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH3_LOWER_DESC_ADDR_0

Offset: 0x10248
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH3_TRANSFER_STATUS_0

Offset: 0x10254
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH3_TIMESTAMP_ADDR_0

Offset: 0x10258
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH3_TIMESTAMP_STATUS_0

Offset: 0x1025c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH3_APR_0

Offset: 0x10260
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH4_CMD_0

Offset: 0x10300
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH4_SOFT_RESET_0

Offset: 0x10304
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH4_STATUS_0

Offset: 0x1030c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH4_INT_STATUS_0

Offset: 0x10310
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH4_INT_SET_0

Offset: 0x10318
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH4_INT_CLEAR_0

Offset: 0x1031c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH4_CTRL_0

Offset: 0x10324
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH4_CONFIG_0

Offset: 0x10328
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH4_AHUB_FIFO_CTRL_0

Offset: 0x1032c
Read/Write: RW
Parity Protection: N
Reset: 0x01010303 (0b0x00,0001,xx00,0001,xx00,0011,xx00,0011)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x1	**OVERFLOW_THRESHOLD**
21:16	0x1	**STARVATION_THRESHOLD**
13:8	0x3	**TX_FIFO_SIZE**
5:0	0x3	**RX_FIFO_SIZE**

ADMA_PAGE1_CH4_TC_STATUS_0

Offset: 0x10330
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH4_LOWER_SOURCE_ADDR_0

Offset: 0x10334
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH4_LOWER_TARGET_ADDR_0

Offset: 0x1033c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH4_TC_0

Offset: 0x10344
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH4_LOWER_DESC_ADDR_0

Offset: 0x10348
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH4_TRANSFER_STATUS_0

Offset: 0x10354
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH4_TIMESTAMP_ADDR_0

Offset: 0x10358
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH4_TIMESTAMP_STATUS_0

Offset: 0x1035c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH4_APR_0

Offset: 0x10360
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH5_CMD_0

Offset: 0x10400
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH5_SOFT_RESET_0

Offset: 0x10404
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH5_STATUS_0

Offset: 0x1040c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH5_INT_STATUS_0

Offset: 0x10410
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH5_INT_SET_0

Offset: 0x10418
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH5_INT_CLEAR_0

Offset: 0x1041c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH5_CTRL_0

Offset: 0x10424
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH5_CONFIG_0

Offset: 0x10428
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH5_AHUB_FIFO_CTRL_0

Offset: 0x1042c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE1_CH5_TC_STATUS_0

Offset: 0x10430
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH5_LOWER_SOURCE_ADDR_0

Offset: 0x10434
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH5_LOWER_TARGET_ADDR_0

Offset: 0x1043c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH5_TC_0

Offset: 0x10444
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH5_LOWER_DESC_ADDR_0

Offset: 0x10448
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH5_TRANSFER_STATUS_0

Offset: 0x10454
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH5_TIMESTAMP_ADDR_0

Offset: 0x10458
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH5_TIMESTAMP_STATUS_0

Offset: 0x1045c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH5_APR_0

Offset: 0x10460
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH6_CMD_0

Offset: 0x10500
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH6_SOFT_RESET_0

Offset: 0x10504
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH6_STATUS_0

Offset: 0x1050c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH6_INT_STATUS_0

Offset: 0x10510
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH6_INT_SET_0

Offset: 0x10518
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH6_INT_CLEAR_0

Offset: 0x1051c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH6_CTRL_0

Offset: 0x10524
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH6_CONFIG_0

Offset: 0x10528
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH6_AHUB_FIFO_CTRL_0

Offset: 0x1052c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE1_CH6_TC_STATUS_0

Offset: 0x10530
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH6_LOWER_SOURCE_ADDR_0

Offset: 0x10534
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH6_LOWER_TARGET_ADDR_0

Offset: 0x1053c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH6_TC_0

Offset: 0x10544
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH6_LOWER_DESC_ADDR_0

Offset: 0x10548
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH6_TRANSFER_STATUS_0

Offset: 0x10554
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH6_TIMESTAMP_ADDR_0

Offset: 0x10558
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH6_TIMESTAMP_STATUS_0

Offset: 0x1055c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH6_APR_0

Offset: 0x10560
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH7_CMD_0

Offset: 0x10600
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH7_SOFT_RESET_0

Offset: 0x10604
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH7_STATUS_0

Offset: 0x1060c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH7_INT_STATUS_0

Offset: 0x10610
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH7_INT_SET_0

Offset: 0x10618
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH7_INT_CLEAR_0

Offset: 0x1061c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH7_CTRL_0

Offset: 0x10624
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH7_CONFIG_0

Offset: 0x10628
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH7_AHUB_FIFO_CTRL_0

Offset: 0x1062c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE1_CH7_TC_STATUS_0

Offset: 0x10630
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH7_LOWER_SOURCE_ADDR_0

Offset: 0x10634
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH7_LOWER_TARGET_ADDR_0

Offset: 0x1063c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH7_TC_0

Offset: 0x10644
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH7_LOWER_DESC_ADDR_0

Offset: 0x10648
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH7_TRANSFER_STATUS_0

Offset: 0x10654
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH7_TIMESTAMP_ADDR_0

Offset: 0x10658
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH7_TIMESTAMP_STATUS_0

Offset: 0x1065c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH7_APR_0

Offset: 0x10660
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH8_CMD_0

Offset: 0x10700
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH8_SOFT_RESET_0

Offset: 0x10704
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH8_STATUS_0

Offset: 0x1070c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH8_INT_STATUS_0

Offset: 0x10710
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH8_INT_SET_0

Offset: 0x10718
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH8_INT_CLEAR_0

Offset: 0x1071c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH8_CTRL_0

Offset: 0x10724
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH8_CONFIG_0

Offset: 0x10728
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH8_AHUB_FIFO_CTRL_0

Offset: 0x1072c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE1_CH8_TC_STATUS_0

Offset: 0x10730
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH8_LOWER_SOURCE_ADDR_0

Offset: 0x10734
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH8_LOWER_TARGET_ADDR_0

Offset: 0x1073c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH8_TC_0

Offset: 0x10744
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH8_LOWER_DESC_ADDR_0

Offset: 0x10748
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH8_TRANSFER_STATUS_0

Offset: 0x10754
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH8_TIMESTAMP_ADDR_0

Offset: 0x10758
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH8_TIMESTAMP_STATUS_0

Offset: 0x1075c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH8_APR_0

Offset: 0x10760
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH9_CMD_0

Offset: 0x10800
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH9_SOFT_RESET_0

Offset: 0x10804
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH9_STATUS_0

Offset: 0x1080c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH9_INT_STATUS_0

Offset: 0x10810
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH9_INT_SET_0

Offset: 0x10818
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH9_INT_CLEAR_0

Offset: 0x1081c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH9_CTRL_0

Offset: 0x10824
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH9_CONFIG_0

Offset: 0x10828
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH9_AHUB_FIFO_CTRL_0

Offset: 0x1082c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE1_CH9_TC_STATUS_0

Offset: 0x10830
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH9_LOWER_SOURCE_ADDR_0

Offset: 0x10834
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH9_LOWER_TARGET_ADDR_0

Offset: 0x1083c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH9_TC_0

Offset: 0x10844
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH9_LOWER_DESC_ADDR_0

Offset: 0x10848
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH9_TRANSFER_STATUS_0

Offset: 0x10854
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH9_TIMESTAMP_ADDR_0

Offset: 0x10858
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH9_TIMESTAMP_STATUS_0

Offset: 0x1085c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH9_APR_0

Offset: 0x10860
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH10_CMD_0

Offset: 0x10900
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH10_SOFT_RESET_0

Offset: 0x10904
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH10_STATUS_0

Offset: 0x1090c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH10_INT_STATUS_0

Offset: 0x10910
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH10_INT_SET_0

Offset: 0x10918
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH10_INT_CLEAR_0

Offset: 0x1091c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH10_CTRL_0

Offset: 0x10924
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH10_CONFIG_0

Offset: 0x10928
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH10_AHUB_FIFO_CTRL_0

Offset: 0x1092c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE1_CH10_TC_STATUS_0

Offset: 0x10930
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH10_LOWER_SOURCE_ADDR_0

Offset: 0x10934
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH10_LOWER_TARGET_ADDR_0

Offset: 0x1093c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH10_TC_0

Offset: 0x10944
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH10_LOWER_DESC_ADDR_0

Offset: 0x10948
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH10_TRANSFER_STATUS_0

Offset: 0x10954
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH10_TIMESTAMP_ADDR_0

Offset: 0x10958
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH10_TIMESTAMP_STATUS_0

Offset: 0x1095c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH10_APR_0

Offset: 0x10960
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH11_CMD_0

Offset: 0x10a00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH11_SOFT_RESET_0

Offset: 0x10a04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH11_STATUS_0

Offset: 0x10a0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH11_INT_STATUS_0

Offset: 0x10a10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH11_INT_SET_0

Offset: 0x10a18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH11_INT_CLEAR_0

Offset: 0x10a1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH11_CTRL_0

Offset: 0x10a24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH11_CONFIG_0

Offset: 0x10a28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH11_AHUB_FIFO_CTRL_0

Offset: 0x10a2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE1_CH11_TC_STATUS_0

Offset: 0x10a30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH11_LOWER_SOURCE_ADDR_0

Offset: 0x10a34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH11_LOWER_TARGET_ADDR_0

Offset: 0x10a3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH11_TC_0

Offset: 0x10a44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH11_LOWER_DESC_ADDR_0

Offset: 0x10a48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH11_TRANSFER_STATUS_0

Offset: 0x10a54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH11_TIMESTAMP_ADDR_0

Offset: 0x10a58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH11_TIMESTAMP_STATUS_0

Offset: 0x10a5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH11_APR_0

Offset: 0x10a60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH12_CMD_0

Offset: 0x10b00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH12_SOFT_RESET_0

Offset: 0x10b04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH12_STATUS_0

Offset: 0x10b0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH12_INT_STATUS_0

Offset: 0x10b10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH12_INT_SET_0

Offset: 0x10b18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH12_INT_CLEAR_0

Offset: 0x10b1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH12_CTRL_0

Offset: 0x10b24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH12_CONFIG_0

Offset: 0x10b28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH12_AHUB_FIFO_CTRL_0

Offset: 0x10b2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE1_CH12_TC_STATUS_0

Offset: 0x10b30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH12_LOWER_SOURCE_ADDR_0

Offset: 0x10b34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH12_LOWER_TARGET_ADDR_0

Offset: 0x10b3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH12_TC_0

Offset: 0x10b44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH12_LOWER_DESC_ADDR_0

Offset: 0x10b48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH12_TRANSFER_STATUS_0

Offset: 0x10b54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH12_TIMESTAMP_ADDR_0

Offset: 0x10b58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH12_TIMESTAMP_STATUS_0

Offset: 0x10b5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH12_APR_0

Offset: 0x10b60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH13_CMD_0

Offset: 0x10c00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH13_SOFT_RESET_0

Offset: 0x10c04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH13_STATUS_0

Offset: 0x10c0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH13_INT_STATUS_0

Offset: 0x10c10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH13_INT_SET_0

Offset: 0x10c18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH13_INT_CLEAR_0

Offset: 0x10c1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH13_CTRL_0

Offset: 0x10c24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH13_CONFIG_0

Offset: 0x10c28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH13_AHUB_FIFO_CTRL_0

Offset: 0x10c2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE1_CH13_TC_STATUS_0

Offset: 0x10c30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH13_LOWER_SOURCE_ADDR_0

Offset: 0x10c34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH13_LOWER_TARGET_ADDR_0

Offset: 0x10c3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH13_TC_0

Offset: 0x10c44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH13_LOWER_DESC_ADDR_0

Offset: 0x10c48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH13_TRANSFER_STATUS_0

Offset: 0x10c54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH13_TIMESTAMP_ADDR_0

Offset: 0x10c58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH13_TIMESTAMP_STATUS_0

Offset: 0x10c5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH13_APR_0

Offset: 0x10c60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH14_CMD_0

Offset: 0x10d00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH14_SOFT_RESET_0

Offset: 0x10d04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH14_STATUS_0

Offset: 0x10d0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH14_INT_STATUS_0

Offset: 0x10d10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH14_INT_SET_0

Offset: 0x10d18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH14_INT_CLEAR_0

Offset: 0x10d1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH14_CTRL_0

Offset: 0x10d24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH14_CONFIG_0

Offset: 0x10d28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH14_AHUB_FIFO_CTRL_0

Offset: 0x10d2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE1_CH14_TC_STATUS_0

Offset: 0x10d30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH14_LOWER_SOURCE_ADDR_0

Offset: 0x10d34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH14_LOWER_TARGET_ADDR_0

Offset: 0x10d3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH14_TC_0

Offset: 0x10d44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH14_LOWER_DESC_ADDR_0

Offset: 0x10d48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH14_TRANSFER_STATUS_0

Offset: 0x10d54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH14_TIMESTAMP_ADDR_0

Offset: 0x10d58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH14_TIMESTAMP_STATUS_0

Offset: 0x10d5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH14_APR_0

Offset: 0x10d60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH15_CMD_0

Offset: 0x10e00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH15_SOFT_RESET_0

Offset: 0x10e04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH15_STATUS_0

Offset: 0x10e0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH15_INT_STATUS_0

Offset: 0x10e10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH15_INT_SET_0

Offset: 0x10e18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH15_INT_CLEAR_0

Offset: 0x10e1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH15_CTRL_0

Offset: 0x10e24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH15_CONFIG_0

Offset: 0x10e28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH15_AHUB_FIFO_CTRL_0

Offset: 0x10e2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE1_CH15_TC_STATUS_0

Offset: 0x10e30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH15_LOWER_SOURCE_ADDR_0

Offset: 0x10e34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH15_LOWER_TARGET_ADDR_0

Offset: 0x10e3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH15_TC_0

Offset: 0x10e44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH15_LOWER_DESC_ADDR_0

Offset: 0x10e48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH15_TRANSFER_STATUS_0

Offset: 0x10e54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH15_TIMESTAMP_ADDR_0

Offset: 0x10e58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH15_TIMESTAMP_STATUS_0

Offset: 0x10e5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH15_APR_0

Offset: 0x10e60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH16_CMD_0

Offset: 0x10f00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH16_SOFT_RESET_0

Offset: 0x10f04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH16_STATUS_0

Offset: 0x10f0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH16_INT_STATUS_0

Offset: 0x10f10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH16_INT_SET_0

Offset: 0x10f18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH16_INT_CLEAR_0

Offset: 0x10f1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH16_CTRL_0

Offset: 0x10f24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH16_CONFIG_0

Offset: 0x10f28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH16_AHUB_FIFO_CTRL_0

Offset: 0x10f2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE1_CH16_TC_STATUS_0

Offset: 0x10f30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH16_LOWER_SOURCE_ADDR_0

Offset: 0x10f34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH16_LOWER_TARGET_ADDR_0

Offset: 0x10f3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH16_TC_0

Offset: 0x10f44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH16_LOWER_DESC_ADDR_0

Offset: 0x10f48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH16_TRANSFER_STATUS_0

Offset: 0x10f54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH16_TIMESTAMP_ADDR_0

Offset: 0x10f58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH16_TIMESTAMP_STATUS_0

Offset: 0x10f5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH16_APR_0

Offset: 0x10f60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH17_CMD_0

Offset: 0x11000
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH17_SOFT_RESET_0

Offset: 0x11004
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH17_STATUS_0

Offset: 0x1100c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH17_INT_STATUS_0

Offset: 0x11010
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH17_INT_SET_0

Offset: 0x11018
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH17_INT_CLEAR_0

Offset: 0x1101c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH17_CTRL_0

Offset: 0x11024
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH17_CONFIG_0

Offset: 0x11028
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH17_AHUB_FIFO_CTRL_0

Offset: 0x1102c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE1_CH17_TC_STATUS_0

Offset: 0x11030
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH17_LOWER_SOURCE_ADDR_0

Offset: 0x11034
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH17_LOWER_TARGET_ADDR_0

Offset: 0x1103c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH17_TC_0

Offset: 0x11044
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH17_LOWER_DESC_ADDR_0

Offset: 0x11048
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH17_TRANSFER_STATUS_0

Offset: 0x11054
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH17_TIMESTAMP_ADDR_0

Offset: 0x11058
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH17_TIMESTAMP_STATUS_0

Offset: 0x1105c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH17_APR_0

Offset: 0x11060
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH18_CMD_0

Offset: 0x11100
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH18_SOFT_RESET_0

Offset: 0x11104
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH18_STATUS_0

Offset: 0x1110c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH18_INT_STATUS_0

Offset: 0x11110
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH18_INT_SET_0

Offset: 0x11118
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH18_INT_CLEAR_0

Offset: 0x1111c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH18_CTRL_0

Offset: 0x11124
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH18_CONFIG_0

Offset: 0x11128
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH18_AHUB_FIFO_CTRL_0

Offset: 0x1112c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE1_CH18_TC_STATUS_0

Offset: 0x11130
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH18_LOWER_SOURCE_ADDR_0

Offset: 0x11134
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH18_LOWER_TARGET_ADDR_0

Offset: 0x1113c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH18_TC_0

Offset: 0x11144
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH18_LOWER_DESC_ADDR_0

Offset: 0x11148
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH18_TRANSFER_STATUS_0

Offset: 0x11154
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH18_TIMESTAMP_ADDR_0

Offset: 0x11158
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH18_TIMESTAMP_STATUS_0

Offset: 0x1115c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH18_APR_0

Offset: 0x11160
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH19_CMD_0

Offset: 0x11200
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH19_SOFT_RESET_0

Offset: 0x11204
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH19_STATUS_0

Offset: 0x1120c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH19_INT_STATUS_0

Offset: 0x11210
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH19_INT_SET_0

Offset: 0x11218
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH19_INT_CLEAR_0

Offset: 0x1121c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH19_CTRL_0

Offset: 0x11224
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH19_CONFIG_0

Offset: 0x11228
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH19_AHUB_FIFO_CTRL_0

Offset: 0x1122c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE1_CH19_TC_STATUS_0

Offset: 0x11230
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH19_LOWER_SOURCE_ADDR_0

Offset: 0x11234
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH19_LOWER_TARGET_ADDR_0

Offset: 0x1123c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH19_TC_0

Offset: 0x11244
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH19_LOWER_DESC_ADDR_0

Offset: 0x11248
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH19_TRANSFER_STATUS_0

Offset: 0x11254
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH19_TIMESTAMP_ADDR_0

Offset: 0x11258
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH19_TIMESTAMP_STATUS_0

Offset: 0x1125c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH19_APR_0

Offset: 0x11260
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH20_CMD_0

Offset: 0x11300
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH20_SOFT_RESET_0

Offset: 0x11304
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH20_STATUS_0

Offset: 0x1130c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH20_INT_STATUS_0

Offset: 0x11310
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH20_INT_SET_0

Offset: 0x11318
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH20_INT_CLEAR_0

Offset: 0x1131c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH20_CTRL_0

Offset: 0x11324
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH20_CONFIG_0

Offset: 0x11328
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH20_AHUB_FIFO_CTRL_0

Offset: 0x1132c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE1_CH20_TC_STATUS_0

Offset: 0x11330
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH20_LOWER_SOURCE_ADDR_0

Offset: 0x11334
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH20_LOWER_TARGET_ADDR_0

Offset: 0x1133c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH20_TC_0

Offset: 0x11344
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH20_LOWER_DESC_ADDR_0

Offset: 0x11348
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH20_TRANSFER_STATUS_0

Offset: 0x11354
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH20_TIMESTAMP_ADDR_0

Offset: 0x11358
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH20_TIMESTAMP_STATUS_0

Offset: 0x1135c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH20_APR_0

Offset: 0x11360
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH21_CMD_0

Offset: 0x11400
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH21_SOFT_RESET_0

Offset: 0x11404
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH21_STATUS_0

Offset: 0x1140c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH21_INT_STATUS_0

Offset: 0x11410
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH21_INT_SET_0

Offset: 0x11418
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH21_INT_CLEAR_0

Offset: 0x1141c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH21_CTRL_0

Offset: 0x11424
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH21_CONFIG_0

Offset: 0x11428
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH21_AHUB_FIFO_CTRL_0

Offset: 0x1142c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE1_CH21_TC_STATUS_0

Offset: 0x11430
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH21_LOWER_SOURCE_ADDR_0

Offset: 0x11434
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH21_LOWER_TARGET_ADDR_0

Offset: 0x1143c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH21_TC_0

Offset: 0x11444
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH21_LOWER_DESC_ADDR_0

Offset: 0x11448
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH21_TRANSFER_STATUS_0

Offset: 0x11454
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH21_TIMESTAMP_ADDR_0

Offset: 0x11458
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH21_TIMESTAMP_STATUS_0

Offset: 0x1145c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH21_APR_0

Offset: 0x11460
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH22_CMD_0

Offset: 0x11500
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH22_SOFT_RESET_0

Offset: 0x11504
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH22_STATUS_0

Offset: 0x1150c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH22_INT_STATUS_0

Offset: 0x11510
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH22_INT_SET_0

Offset: 0x11518
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH22_INT_CLEAR_0

Offset: 0x1151c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH22_CTRL_0

Offset: 0x11524
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH22_CONFIG_0

Offset: 0x11528
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH22_AHUB_FIFO_CTRL_0

Offset: 0x1152c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE1_CH22_TC_STATUS_0

Offset: 0x11530
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH22_LOWER_SOURCE_ADDR_0

Offset: 0x11534
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH22_LOWER_TARGET_ADDR_0

Offset: 0x1153c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH22_TC_0

Offset: 0x11544
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH22_LOWER_DESC_ADDR_0

Offset: 0x11548
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH22_TRANSFER_STATUS_0

Offset: 0x11554
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH22_TIMESTAMP_ADDR_0

Offset: 0x11558
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH22_TIMESTAMP_STATUS_0

Offset: 0x1155c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH22_APR_0

Offset: 0x11560
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH23_CMD_0

Offset: 0x11600
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH23_SOFT_RESET_0

Offset: 0x11604
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH23_STATUS_0

Offset: 0x1160c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH23_INT_STATUS_0

Offset: 0x11610
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH23_INT_SET_0

Offset: 0x11618
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH23_INT_CLEAR_0

Offset: 0x1161c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH23_CTRL_0

Offset: 0x11624
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH23_CONFIG_0

Offset: 0x11628
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH23_AHUB_FIFO_CTRL_0

Offset: 0x1162c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE1_CH23_TC_STATUS_0

Offset: 0x11630
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH23_LOWER_SOURCE_ADDR_0

Offset: 0x11634
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH23_LOWER_TARGET_ADDR_0

Offset: 0x1163c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH23_TC_0

Offset: 0x11644
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH23_LOWER_DESC_ADDR_0

Offset: 0x11648
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH23_TRANSFER_STATUS_0

Offset: 0x11654
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH23_TIMESTAMP_ADDR_0

Offset: 0x11658
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH23_TIMESTAMP_STATUS_0

Offset: 0x1165c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH23_APR_0

Offset: 0x11660
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH24_CMD_0

Offset: 0x11700
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH24_SOFT_RESET_0

Offset: 0x11704
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH24_STATUS_0

Offset: 0x1170c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH24_INT_STATUS_0

Offset: 0x11710
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH24_INT_SET_0

Offset: 0x11718
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH24_INT_CLEAR_0

Offset: 0x1171c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH24_CTRL_0

Offset: 0x11724
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH24_CONFIG_0

Offset: 0x11728
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH24_AHUB_FIFO_CTRL_0

Offset: 0x1172c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE1_CH24_TC_STATUS_0

Offset: 0x11730
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH24_LOWER_SOURCE_ADDR_0

Offset: 0x11734
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH24_LOWER_TARGET_ADDR_0

Offset: 0x1173c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH24_TC_0

Offset: 0x11744
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH24_LOWER_DESC_ADDR_0

Offset: 0x11748
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH24_TRANSFER_STATUS_0

Offset: 0x11754
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH24_TIMESTAMP_ADDR_0

Offset: 0x11758
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH24_TIMESTAMP_STATUS_0

Offset: 0x1175c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH24_APR_0

Offset: 0x11760
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH25_CMD_0

Offset: 0x11800
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH25_SOFT_RESET_0

Offset: 0x11804
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH25_STATUS_0

Offset: 0x1180c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH25_INT_STATUS_0

Offset: 0x11810
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH25_INT_SET_0

Offset: 0x11818
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH25_INT_CLEAR_0

Offset: 0x1181c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH25_CTRL_0

Offset: 0x11824
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH25_CONFIG_0

Offset: 0x11828
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH25_AHUB_FIFO_CTRL_0

Offset: 0x1182c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE1_CH25_TC_STATUS_0

Offset: 0x11830
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH25_LOWER_SOURCE_ADDR_0

Offset: 0x11834
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH25_LOWER_TARGET_ADDR_0

Offset: 0x1183c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH25_TC_0

Offset: 0x11844
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH25_LOWER_DESC_ADDR_0

Offset: 0x11848
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH25_TRANSFER_STATUS_0

Offset: 0x11854
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH25_TIMESTAMP_ADDR_0

Offset: 0x11858
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH25_TIMESTAMP_STATUS_0

Offset: 0x1185c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH25_APR_0

Offset: 0x11860
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH26_CMD_0

Offset: 0x11900
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH26_SOFT_RESET_0

Offset: 0x11904
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH26_STATUS_0

Offset: 0x1190c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH26_INT_STATUS_0

Offset: 0x11910
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH26_INT_SET_0

Offset: 0x11918
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH26_INT_CLEAR_0

Offset: 0x1191c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH26_CTRL_0

Offset: 0x11924
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH26_CONFIG_0

Offset: 0x11928
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH26_AHUB_FIFO_CTRL_0

Offset: 0x1192c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE1_CH26_TC_STATUS_0

Offset: 0x11930
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH26_LOWER_SOURCE_ADDR_0

Offset: 0x11934
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH26_LOWER_TARGET_ADDR_0

Offset: 0x1193c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH26_TC_0

Offset: 0x11944
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH26_LOWER_DESC_ADDR_0

Offset: 0x11948
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH26_TRANSFER_STATUS_0

Offset: 0x11954
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH26_TIMESTAMP_ADDR_0

Offset: 0x11958
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH26_TIMESTAMP_STATUS_0

Offset: 0x1195c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH26_APR_0

Offset: 0x11960
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH27_CMD_0

Offset: 0x11a00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH27_SOFT_RESET_0

Offset: 0x11a04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH27_STATUS_0

Offset: 0x11a0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH27_INT_STATUS_0

Offset: 0x11a10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH27_INT_SET_0

Offset: 0x11a18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH27_INT_CLEAR_0

Offset: 0x11a1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH27_CTRL_0

Offset: 0x11a24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH27_CONFIG_0

Offset: 0x11a28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH27_AHUB_FIFO_CTRL_0

Offset: 0x11a2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE1_CH27_TC_STATUS_0

Offset: 0x11a30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH27_LOWER_SOURCE_ADDR_0

Offset: 0x11a34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH27_LOWER_TARGET_ADDR_0

Offset: 0x11a3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH27_TC_0

Offset: 0x11a44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH27_LOWER_DESC_ADDR_0

Offset: 0x11a48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH27_TRANSFER_STATUS_0

Offset: 0x11a54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH27_TIMESTAMP_ADDR_0

Offset: 0x11a58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH27_TIMESTAMP_STATUS_0

Offset: 0x11a5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH27_APR_0

Offset: 0x11a60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH28_CMD_0

Offset: 0x11b00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH28_SOFT_RESET_0

Offset: 0x11b04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH28_STATUS_0

Offset: 0x11b0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH28_INT_STATUS_0

Offset: 0x11b10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH28_INT_SET_0

Offset: 0x11b18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH28_INT_CLEAR_0

Offset: 0x11b1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH28_CTRL_0

Offset: 0x11b24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH28_CONFIG_0

Offset: 0x11b28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH28_AHUB_FIFO_CTRL_0

Offset: 0x11b2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE1_CH28_TC_STATUS_0

Offset: 0x11b30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH28_LOWER_SOURCE_ADDR_0

Offset: 0x11b34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH28_LOWER_TARGET_ADDR_0

Offset: 0x11b3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH28_TC_0

Offset: 0x11b44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH28_LOWER_DESC_ADDR_0

Offset: 0x11b48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH28_TRANSFER_STATUS_0

Offset: 0x11b54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH28_TIMESTAMP_ADDR_0

Offset: 0x11b58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH28_TIMESTAMP_STATUS_0

Offset: 0x11b5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH28_APR_0

Offset: 0x11b60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH29_CMD_0

Offset: 0x11c00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH29_SOFT_RESET_0

Offset: 0x11c04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH29_STATUS_0

Offset: 0x11c0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH29_INT_STATUS_0

Offset: 0x11c10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH29_INT_SET_0

Offset: 0x11c18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH29_INT_CLEAR_0

Offset: 0x11c1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH29_CTRL_0

Offset: 0x11c24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH29_CONFIG_0

Offset: 0x11c28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH29_AHUB_FIFO_CTRL_0

Offset: 0x11c2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE1_CH29_TC_STATUS_0

Offset: 0x11c30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH29_LOWER_SOURCE_ADDR_0

Offset: 0x11c34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH29_LOWER_TARGET_ADDR_0

Offset: 0x11c3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH29_TC_0

Offset: 0x11c44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH29_LOWER_DESC_ADDR_0

Offset: 0x11c48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH29_TRANSFER_STATUS_0

Offset: 0x11c54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH29_TIMESTAMP_ADDR_0

Offset: 0x11c58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH29_TIMESTAMP_STATUS_0

Offset: 0x11c5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH29_APR_0

Offset: 0x11c60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH30_CMD_0

Offset: 0x11d00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH30_SOFT_RESET_0

Offset: 0x11d04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH30_STATUS_0

Offset: 0x11d0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH30_INT_STATUS_0

Offset: 0x11d10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH30_INT_SET_0

Offset: 0x11d18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH30_INT_CLEAR_0

Offset: 0x11d1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH30_CTRL_0

Offset: 0x11d24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH30_CONFIG_0

Offset: 0x11d28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH30_AHUB_FIFO_CTRL_0

Offset: 0x11d2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE1_CH30_TC_STATUS_0

Offset: 0x11d30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH30_LOWER_SOURCE_ADDR_0

Offset: 0x11d34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH30_LOWER_TARGET_ADDR_0

Offset: 0x11d3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH30_TC_0

Offset: 0x11d44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH30_LOWER_DESC_ADDR_0

Offset: 0x11d48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH30_TRANSFER_STATUS_0

Offset: 0x11d54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH30_TIMESTAMP_ADDR_0

Offset: 0x11d58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH30_TIMESTAMP_STATUS_0

Offset: 0x11d5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH30_APR_0

Offset: 0x11d60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH31_CMD_0

Offset: 0x11e00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH31_SOFT_RESET_0

Offset: 0x11e04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH31_STATUS_0

Offset: 0x11e0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH31_INT_STATUS_0

Offset: 0x11e10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH31_INT_SET_0

Offset: 0x11e18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH31_INT_CLEAR_0

Offset: 0x11e1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH31_CTRL_0

Offset: 0x11e24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH31_CONFIG_0

Offset: 0x11e28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH31_AHUB_FIFO_CTRL_0

Offset: 0x11e2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE1_CH31_TC_STATUS_0

Offset: 0x11e30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH31_LOWER_SOURCE_ADDR_0

Offset: 0x11e34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH31_LOWER_TARGET_ADDR_0

Offset: 0x11e3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH31_TC_0

Offset: 0x11e44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH31_LOWER_DESC_ADDR_0

Offset: 0x11e48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH31_TRANSFER_STATUS_0

Offset: 0x11e54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH31_TIMESTAMP_ADDR_0

Offset: 0x11e58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH31_TIMESTAMP_STATUS_0

Offset: 0x11e5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH31_APR_0

Offset: 0x11e60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH32_CMD_0

Offset: 0x11f00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH32_SOFT_RESET_0

Offset: 0x11f04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH32_STATUS_0

Offset: 0x11f0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH32_INT_STATUS_0

Offset: 0x11f10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH32_INT_SET_0

Offset: 0x11f18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH32_INT_CLEAR_0

Offset: 0x11f1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH32_CTRL_0

Offset: 0x11f24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE1_CH32_CONFIG_0

Offset: 0x11f28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE1_CH32_AHUB_FIFO_CTRL_0

Offset: 0x11f2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE1_CH32_TC_STATUS_0

Offset: 0x11f30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE1_CH32_LOWER_SOURCE_ADDR_0

Offset: 0x11f34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH32_LOWER_TARGET_ADDR_0

Offset: 0x11f3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH32_TC_0

Offset: 0x11f44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE1_CH32_LOWER_DESC_ADDR_0

Offset: 0x11f48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE1_CH32_TRANSFER_STATUS_0

Offset: 0x11f54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE1_CH32_TIMESTAMP_ADDR_0

Offset: 0x11f58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE1_CH32_TIMESTAMP_STATUS_0

Offset: 0x11f5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE1_CH32_APR_0

Offset: 0x11f60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH1_CMD_0

Offset: 0x20000
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH1_SOFT_RESET_0

Offset: 0x20004
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH1_STATUS_0

Offset: 0x2000c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH1_INT_STATUS_0

Offset: 0x20010
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH1_INT_SET_0

Offset: 0x20018
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH1_INT_CLEAR_0

Offset: 0x2001c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH1_CTRL_0

Offset: 0x20024
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH1_CONFIG_0

Offset: 0x20028
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH1_AHUB_FIFO_CTRL_0

Offset: 0x2002c
Read/Write: RW
Parity Protection: N
Reset: 0x01010303 (0b0x00,0001,xx00,0001,xx00,0011,xx00,0011)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x1	**OVERFLOW_THRESHOLD**
21:16	0x1	**STARVATION_THRESHOLD**
13:8	0x3	**TX_FIFO_SIZE**
5:0	0x3	**RX_FIFO_SIZE**

ADMA_PAGE2_CH1_TC_STATUS_0

Offset: 0x20030
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH1_LOWER_SOURCE_ADDR_0

Offset: 0x20034
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH1_LOWER_TARGET_ADDR_0

Offset: 0x2003c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH1_TC_0

Offset: 0x20044
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH1_LOWER_DESC_ADDR_0

Offset: 0x20048
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH1_TRANSFER_STATUS_0

Offset: 0x20054
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH1_TIMESTAMP_ADDR_0

Offset: 0x20058
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH1_TIMESTAMP_STATUS_0

Offset: 0x2005c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH1_APR_0

Offset: 0x20060
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH2_CMD_0

Offset: 0x20100
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH2_SOFT_RESET_0

Offset: 0x20104
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH2_STATUS_0

Offset: 0x2010c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH2_INT_STATUS_0

Offset: 0x20110
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH2_INT_SET_0

Offset: 0x20118
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH2_INT_CLEAR_0

Offset: 0x2011c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH2_CTRL_0

Offset: 0x20124
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH2_CONFIG_0

Offset: 0x20128
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH2_AHUB_FIFO_CTRL_0

Offset: 0x2012c
Read/Write: RW
Parity Protection: N
Reset: 0x01010303 (0b0x00,0001,xx00,0001,xx00,0011,xx00,0011)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x1	**OVERFLOW_THRESHOLD**
21:16	0x1	**STARVATION_THRESHOLD**
13:8	0x3	**TX_FIFO_SIZE**
5:0	0x3	**RX_FIFO_SIZE**

ADMA_PAGE2_CH2_TC_STATUS_0

Offset: 0x20130
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH2_LOWER_SOURCE_ADDR_0

Offset: 0x20134
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH2_LOWER_TARGET_ADDR_0

Offset: 0x2013c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH2_TC_0

Offset: 0x20144
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH2_LOWER_DESC_ADDR_0

Offset: 0x20148
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH2_TRANSFER_STATUS_0

Offset: 0x20154
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH2_TIMESTAMP_ADDR_0

Offset: 0x20158
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH2_TIMESTAMP_STATUS_0

Offset: 0x2015c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH2_APR_0

Offset: 0x20160
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH3_CMD_0

Offset: 0x20200
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH3_SOFT_RESET_0

Offset: 0x20204
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH3_STATUS_0

Offset: 0x2020c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH3_INT_STATUS_0

Offset: 0x20210
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH3_INT_SET_0

Offset: 0x20218
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH3_INT_CLEAR_0

Offset: 0x2021c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH3_CTRL_0

Offset: 0x20224
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH3_CONFIG_0

Offset: 0x20228
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH3_AHUB_FIFO_CTRL_0

Offset: 0x2022c
Read/Write: RW
Parity Protection: N
Reset: 0x01010303 (0b0x00,0001,xx00,0001,xx00,0011,xx00,0011)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x1	**OVERFLOW_THRESHOLD**
21:16	0x1	**STARVATION_THRESHOLD**
13:8	0x3	**TX_FIFO_SIZE**
5:0	0x3	**RX_FIFO_SIZE**

ADMA_PAGE2_CH3_TC_STATUS_0

Offset: 0x20230
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH3_LOWER_SOURCE_ADDR_0

Offset: 0x20234
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH3_LOWER_TARGET_ADDR_0

Offset: 0x2023c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH3_TC_0

Offset: 0x20244
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH3_LOWER_DESC_ADDR_0

Offset: 0x20248
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH3_TRANSFER_STATUS_0

Offset: 0x20254
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH3_TIMESTAMP_ADDR_0

Offset: 0x20258
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH3_TIMESTAMP_STATUS_0

Offset: 0x2025c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH3_APR_0

Offset: 0x20260
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH4_CMD_0

Offset: 0x20300
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH4_SOFT_RESET_0

Offset: 0x20304
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH4_STATUS_0

Offset: 0x2030c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH4_INT_STATUS_0

Offset: 0x20310
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH4_INT_SET_0

Offset: 0x20318
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH4_INT_CLEAR_0

Offset: 0x2031c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH4_CTRL_0

Offset: 0x20324
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH4_CONFIG_0

Offset: 0x20328
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH4_AHUB_FIFO_CTRL_0

Offset: 0x2032c
Read/Write: RW
Parity Protection: N
Reset: 0x01010303 (0b0x00,0001,xx00,0001,xx00,0011,xx00,0011)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x1	**OVERFLOW_THRESHOLD**
21:16	0x1	**STARVATION_THRESHOLD**
13:8	0x3	**TX_FIFO_SIZE**
5:0	0x3	**RX_FIFO_SIZE**

ADMA_PAGE2_CH4_TC_STATUS_0

Offset: 0x20330
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH4_LOWER_SOURCE_ADDR_0

Offset: 0x20334
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH4_LOWER_TARGET_ADDR_0

Offset: 0x2033c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH4_TC_0

Offset: 0x20344
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH4_LOWER_DESC_ADDR_0

Offset: 0x20348
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH4_TRANSFER_STATUS_0

Offset: 0x20354
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH4_TIMESTAMP_ADDR_0

Offset: 0x20358
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH4_TIMESTAMP_STATUS_0

Offset: 0x2035c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH4_APR_0

Offset: 0x20360
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH5_CMD_0

Offset: 0x20400
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH5_SOFT_RESET_0

Offset: 0x20404
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH5_STATUS_0

Offset: 0x2040c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH5_INT_STATUS_0

Offset: 0x20410
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH5_INT_SET_0

Offset: 0x20418
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH5_INT_CLEAR_0

Offset: 0x2041c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH5_CTRL_0

Offset: 0x20424
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH5_CONFIG_0

Offset: 0x20428
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH5_AHUB_FIFO_CTRL_0

Offset: 0x2042c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE2_CH5_TC_STATUS_0

Offset: 0x20430
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH5_LOWER_SOURCE_ADDR_0

Offset: 0x20434
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH5_LOWER_TARGET_ADDR_0

Offset: 0x2043c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH5_TC_0

Offset: 0x20444
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH5_LOWER_DESC_ADDR_0

Offset: 0x20448
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH5_TRANSFER_STATUS_0

Offset: 0x20454
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH5_TIMESTAMP_ADDR_0

Offset: 0x20458
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH5_TIMESTAMP_STATUS_0

Offset: 0x2045c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH5_APR_0

Offset: 0x20460
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH6_CMD_0

Offset: 0x20500
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH6_SOFT_RESET_0

Offset: 0x20504
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH6_STATUS_0

Offset: 0x2050c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH6_INT_STATUS_0

Offset: 0x20510
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH6_INT_SET_0

Offset: 0x20518
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH6_INT_CLEAR_0

Offset: 0x2051c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH6_CTRL_0

Offset: 0x20524
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH6_CONFIG_0

Offset: 0x20528
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH6_AHUB_FIFO_CTRL_0

Offset: 0x2052c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE2_CH6_TC_STATUS_0

Offset: 0x20530
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH6_LOWER_SOURCE_ADDR_0

Offset: 0x20534
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH6_LOWER_TARGET_ADDR_0

Offset: 0x2053c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH6_TC_0

Offset: 0x20544
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH6_LOWER_DESC_ADDR_0

Offset: 0x20548
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH6_TRANSFER_STATUS_0

Offset: 0x20554
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH6_TIMESTAMP_ADDR_0

Offset: 0x20558
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH6_TIMESTAMP_STATUS_0

Offset: 0x2055c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH6_APR_0

Offset: 0x20560
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH7_CMD_0

Offset: 0x20600
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH7_SOFT_RESET_0

Offset: 0x20604
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH7_STATUS_0

Offset: 0x2060c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH7_INT_STATUS_0

Offset: 0x20610
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH7_INT_SET_0

Offset: 0x20618
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH7_INT_CLEAR_0

Offset: 0x2061c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH7_CTRL_0

Offset: 0x20624
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH7_CONFIG_0

Offset: 0x20628
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH7_AHUB_FIFO_CTRL_0

Offset: 0x2062c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE2_CH7_TC_STATUS_0

Offset: 0x20630
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH7_LOWER_SOURCE_ADDR_0

Offset: 0x20634
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH7_LOWER_TARGET_ADDR_0

Offset: 0x2063c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH7_TC_0

Offset: 0x20644
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH7_LOWER_DESC_ADDR_0

Offset: 0x20648
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH7_TRANSFER_STATUS_0

Offset: 0x20654
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH7_TIMESTAMP_ADDR_0

Offset: 0x20658
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH7_TIMESTAMP_STATUS_0

Offset: 0x2065c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH7_APR_0

Offset: 0x20660
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH8_CMD_0

Offset: 0x20700
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH8_SOFT_RESET_0

Offset: 0x20704
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH8_STATUS_0

Offset: 0x2070c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH8_INT_STATUS_0

Offset: 0x20710
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH8_INT_SET_0

Offset: 0x20718
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH8_INT_CLEAR_0

Offset: 0x2071c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH8_CTRL_0

Offset: 0x20724
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH8_CONFIG_0

Offset: 0x20728
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH8_AHUB_FIFO_CTRL_0

Offset: 0x2072c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE2_CH8_TC_STATUS_0

Offset: 0x20730
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH8_LOWER_SOURCE_ADDR_0

Offset: 0x20734
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH8_LOWER_TARGET_ADDR_0

Offset: 0x2073c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH8_TC_0

Offset: 0x20744
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH8_LOWER_DESC_ADDR_0

Offset: 0x20748
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH8_TRANSFER_STATUS_0

Offset: 0x20754
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH8_TIMESTAMP_ADDR_0

Offset: 0x20758
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH8_TIMESTAMP_STATUS_0

Offset: 0x2075c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH8_APR_0

Offset: 0x20760
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH9_CMD_0

Offset: 0x20800
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH9_SOFT_RESET_0

Offset: 0x20804
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH9_STATUS_0

Offset: 0x2080c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH9_INT_STATUS_0

Offset: 0x20810
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH9_INT_SET_0

Offset: 0x20818
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH9_INT_CLEAR_0

Offset: 0x2081c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH9_CTRL_0

Offset: 0x20824
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH9_CONFIG_0

Offset: 0x20828
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH9_AHUB_FIFO_CTRL_0

Offset: 0x2082c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE2_CH9_TC_STATUS_0

Offset: 0x20830
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH9_LOWER_SOURCE_ADDR_0

Offset: 0x20834
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH9_LOWER_TARGET_ADDR_0

Offset: 0x2083c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH9_TC_0

Offset: 0x20844
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH9_LOWER_DESC_ADDR_0

Offset: 0x20848
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH9_TRANSFER_STATUS_0

Offset: 0x20854
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH9_TIMESTAMP_ADDR_0

Offset: 0x20858
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH9_TIMESTAMP_STATUS_0

Offset: 0x2085c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH9_APR_0

Offset: 0x20860
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH10_CMD_0

Offset: 0x20900
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH10_SOFT_RESET_0

Offset: 0x20904
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH10_STATUS_0

Offset: 0x2090c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH10_INT_STATUS_0

Offset: 0x20910
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH10_INT_SET_0

Offset: 0x20918
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH10_INT_CLEAR_0

Offset: 0x2091c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH10_CTRL_0

Offset: 0x20924
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH10_CONFIG_0

Offset: 0x20928
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH10_AHUB_FIFO_CTRL_0

Offset: 0x2092c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE2_CH10_TC_STATUS_0

Offset: 0x20930
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH10_LOWER_SOURCE_ADDR_0

Offset: 0x20934
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH10_LOWER_TARGET_ADDR_0

Offset: 0x2093c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH10_TC_0

Offset: 0x20944
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH10_LOWER_DESC_ADDR_0

Offset: 0x20948
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH10_TRANSFER_STATUS_0

Offset: 0x20954
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH10_TIMESTAMP_ADDR_0

Offset: 0x20958
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH10_TIMESTAMP_STATUS_0

Offset: 0x2095c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH10_APR_0

Offset: 0x20960
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH11_CMD_0

Offset: 0x20a00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH11_SOFT_RESET_0

Offset: 0x20a04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH11_STATUS_0

Offset: 0x20a0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH11_INT_STATUS_0

Offset: 0x20a10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH11_INT_SET_0

Offset: 0x20a18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH11_INT_CLEAR_0

Offset: 0x20a1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH11_CTRL_0

Offset: 0x20a24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH11_CONFIG_0

Offset: 0x20a28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH11_AHUB_FIFO_CTRL_0

Offset: 0x20a2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE2_CH11_TC_STATUS_0

Offset: 0x20a30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH11_LOWER_SOURCE_ADDR_0

Offset: 0x20a34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH11_LOWER_TARGET_ADDR_0

Offset: 0x20a3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH11_TC_0

Offset: 0x20a44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH11_LOWER_DESC_ADDR_0

Offset: 0x20a48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH11_TRANSFER_STATUS_0

Offset: 0x20a54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH11_TIMESTAMP_ADDR_0

Offset: 0x20a58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH11_TIMESTAMP_STATUS_0

Offset: 0x20a5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH11_APR_0

Offset: 0x20a60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH12_CMD_0

Offset: 0x20b00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH12_SOFT_RESET_0

Offset: 0x20b04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH12_STATUS_0

Offset: 0x20b0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH12_INT_STATUS_0

Offset: 0x20b10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH12_INT_SET_0

Offset: 0x20b18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH12_INT_CLEAR_0

Offset: 0x20b1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH12_CTRL_0

Offset: 0x20b24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH12_CONFIG_0

Offset: 0x20b28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH12_AHUB_FIFO_CTRL_0

Offset: 0x20b2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE2_CH12_TC_STATUS_0

Offset: 0x20b30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH12_LOWER_SOURCE_ADDR_0

Offset: 0x20b34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH12_LOWER_TARGET_ADDR_0

Offset: 0x20b3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH12_TC_0

Offset: 0x20b44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH12_LOWER_DESC_ADDR_0

Offset: 0x20b48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH12_TRANSFER_STATUS_0

Offset: 0x20b54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH12_TIMESTAMP_ADDR_0

Offset: 0x20b58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH12_TIMESTAMP_STATUS_0

Offset: 0x20b5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH12_APR_0

Offset: 0x20b60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH13_CMD_0

Offset: 0x20c00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH13_SOFT_RESET_0

Offset: 0x20c04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH13_STATUS_0

Offset: 0x20c0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH13_INT_STATUS_0

Offset: 0x20c10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH13_INT_SET_0

Offset: 0x20c18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH13_INT_CLEAR_0

Offset: 0x20c1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH13_CTRL_0

Offset: 0x20c24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH13_CONFIG_0

Offset: 0x20c28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH13_AHUB_FIFO_CTRL_0

Offset: 0x20c2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE2_CH13_TC_STATUS_0

Offset: 0x20c30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH13_LOWER_SOURCE_ADDR_0

Offset: 0x20c34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH13_LOWER_TARGET_ADDR_0

Offset: 0x20c3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH13_TC_0

Offset: 0x20c44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH13_LOWER_DESC_ADDR_0

Offset: 0x20c48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH13_TRANSFER_STATUS_0

Offset: 0x20c54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH13_TIMESTAMP_ADDR_0

Offset: 0x20c58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH13_TIMESTAMP_STATUS_0

Offset: 0x20c5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH13_APR_0

Offset: 0x20c60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH14_CMD_0

Offset: 0x20d00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH14_SOFT_RESET_0

Offset: 0x20d04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH14_STATUS_0

Offset: 0x20d0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH14_INT_STATUS_0

Offset: 0x20d10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH14_INT_SET_0

Offset: 0x20d18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH14_INT_CLEAR_0

Offset: 0x20d1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH14_CTRL_0

Offset: 0x20d24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH14_CONFIG_0

Offset: 0x20d28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH14_AHUB_FIFO_CTRL_0

Offset: 0x20d2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE2_CH14_TC_STATUS_0

Offset: 0x20d30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH14_LOWER_SOURCE_ADDR_0

Offset: 0x20d34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH14_LOWER_TARGET_ADDR_0

Offset: 0x20d3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH14_TC_0

Offset: 0x20d44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH14_LOWER_DESC_ADDR_0

Offset: 0x20d48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH14_TRANSFER_STATUS_0

Offset: 0x20d54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH14_TIMESTAMP_ADDR_0

Offset: 0x20d58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH14_TIMESTAMP_STATUS_0

Offset: 0x20d5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH14_APR_0

Offset: 0x20d60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH15_CMD_0

Offset: 0x20e00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH15_SOFT_RESET_0

Offset: 0x20e04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH15_STATUS_0

Offset: 0x20e0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH15_INT_STATUS_0

Offset: 0x20e10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH15_INT_SET_0

Offset: 0x20e18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH15_INT_CLEAR_0

Offset: 0x20e1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH15_CTRL_0

Offset: 0x20e24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH15_CONFIG_0

Offset: 0x20e28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH15_AHUB_FIFO_CTRL_0

Offset: 0x20e2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE2_CH15_TC_STATUS_0

Offset: 0x20e30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH15_LOWER_SOURCE_ADDR_0

Offset: 0x20e34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH15_LOWER_TARGET_ADDR_0

Offset: 0x20e3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH15_TC_0

Offset: 0x20e44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH15_LOWER_DESC_ADDR_0

Offset: 0x20e48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH15_TRANSFER_STATUS_0

Offset: 0x20e54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH15_TIMESTAMP_ADDR_0

Offset: 0x20e58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH15_TIMESTAMP_STATUS_0

Offset: 0x20e5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH15_APR_0

Offset: 0x20e60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH16_CMD_0

Offset: 0x20f00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH16_SOFT_RESET_0

Offset: 0x20f04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH16_STATUS_0

Offset: 0x20f0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH16_INT_STATUS_0

Offset: 0x20f10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH16_INT_SET_0

Offset: 0x20f18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH16_INT_CLEAR_0

Offset: 0x20f1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH16_CTRL_0

Offset: 0x20f24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH16_CONFIG_0

Offset: 0x20f28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH16_AHUB_FIFO_CTRL_0

Offset: 0x20f2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE2_CH16_TC_STATUS_0

Offset: 0x20f30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH16_LOWER_SOURCE_ADDR_0

Offset: 0x20f34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH16_LOWER_TARGET_ADDR_0

Offset: 0x20f3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH16_TC_0

Offset: 0x20f44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH16_LOWER_DESC_ADDR_0

Offset: 0x20f48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH16_TRANSFER_STATUS_0

Offset: 0x20f54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH16_TIMESTAMP_ADDR_0

Offset: 0x20f58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH16_TIMESTAMP_STATUS_0

Offset: 0x20f5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH16_APR_0

Offset: 0x20f60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH17_CMD_0

Offset: 0x21000
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH17_SOFT_RESET_0

Offset: 0x21004
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH17_STATUS_0

Offset: 0x2100c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH17_INT_STATUS_0

Offset: 0x21010
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH17_INT_SET_0

Offset: 0x21018
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH17_INT_CLEAR_0

Offset: 0x2101c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH17_CTRL_0

Offset: 0x21024
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH17_CONFIG_0

Offset: 0x21028
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH17_AHUB_FIFO_CTRL_0

Offset: 0x2102c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE2_CH17_TC_STATUS_0

Offset: 0x21030
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH17_LOWER_SOURCE_ADDR_0

Offset: 0x21034
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH17_LOWER_TARGET_ADDR_0

Offset: 0x2103c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH17_TC_0

Offset: 0x21044
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH17_LOWER_DESC_ADDR_0

Offset: 0x21048
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH17_TRANSFER_STATUS_0

Offset: 0x21054
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH17_TIMESTAMP_ADDR_0

Offset: 0x21058
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH17_TIMESTAMP_STATUS_0

Offset: 0x2105c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH17_APR_0

Offset: 0x21060
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH18_CMD_0

Offset: 0x21100
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH18_SOFT_RESET_0

Offset: 0x21104
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH18_STATUS_0

Offset: 0x2110c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH18_INT_STATUS_0

Offset: 0x21110
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH18_INT_SET_0

Offset: 0x21118
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH18_INT_CLEAR_0

Offset: 0x2111c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH18_CTRL_0

Offset: 0x21124
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH18_CONFIG_0

Offset: 0x21128
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH18_AHUB_FIFO_CTRL_0

Offset: 0x2112c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE2_CH18_TC_STATUS_0

Offset: 0x21130
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH18_LOWER_SOURCE_ADDR_0

Offset: 0x21134
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH18_LOWER_TARGET_ADDR_0

Offset: 0x2113c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH18_TC_0

Offset: 0x21144
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH18_LOWER_DESC_ADDR_0

Offset: 0x21148
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH18_TRANSFER_STATUS_0

Offset: 0x21154
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH18_TIMESTAMP_ADDR_0

Offset: 0x21158
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH18_TIMESTAMP_STATUS_0

Offset: 0x2115c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH18_APR_0

Offset: 0x21160
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH19_CMD_0

Offset: 0x21200
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH19_SOFT_RESET_0

Offset: 0x21204
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH19_STATUS_0

Offset: 0x2120c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH19_INT_STATUS_0

Offset: 0x21210
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH19_INT_SET_0

Offset: 0x21218
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH19_INT_CLEAR_0

Offset: 0x2121c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH19_CTRL_0

Offset: 0x21224
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH19_CONFIG_0

Offset: 0x21228
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH19_AHUB_FIFO_CTRL_0

Offset: 0x2122c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE2_CH19_TC_STATUS_0

Offset: 0x21230
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH19_LOWER_SOURCE_ADDR_0

Offset: 0x21234
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH19_LOWER_TARGET_ADDR_0

Offset: 0x2123c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH19_TC_0

Offset: 0x21244
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH19_LOWER_DESC_ADDR_0

Offset: 0x21248
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH19_TRANSFER_STATUS_0

Offset: 0x21254
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH19_TIMESTAMP_ADDR_0

Offset: 0x21258
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH19_TIMESTAMP_STATUS_0

Offset: 0x2125c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH19_APR_0

Offset: 0x21260
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH20_CMD_0

Offset: 0x21300
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH20_SOFT_RESET_0

Offset: 0x21304
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH20_STATUS_0

Offset: 0x2130c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH20_INT_STATUS_0

Offset: 0x21310
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH20_INT_SET_0

Offset: 0x21318
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH20_INT_CLEAR_0

Offset: 0x2131c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH20_CTRL_0

Offset: 0x21324
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH20_CONFIG_0

Offset: 0x21328
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH20_AHUB_FIFO_CTRL_0

Offset: 0x2132c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE2_CH20_TC_STATUS_0

Offset: 0x21330
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH20_LOWER_SOURCE_ADDR_0

Offset: 0x21334
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH20_LOWER_TARGET_ADDR_0

Offset: 0x2133c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH20_TC_0

Offset: 0x21344
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH20_LOWER_DESC_ADDR_0

Offset: 0x21348
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH20_TRANSFER_STATUS_0

Offset: 0x21354
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH20_TIMESTAMP_ADDR_0

Offset: 0x21358
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH20_TIMESTAMP_STATUS_0

Offset: 0x2135c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH20_APR_0

Offset: 0x21360
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH21_CMD_0

Offset: 0x21400
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH21_SOFT_RESET_0

Offset: 0x21404
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH21_STATUS_0

Offset: 0x2140c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH21_INT_STATUS_0

Offset: 0x21410
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH21_INT_SET_0

Offset: 0x21418
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH21_INT_CLEAR_0

Offset: 0x2141c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH21_CTRL_0

Offset: 0x21424
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH21_CONFIG_0

Offset: 0x21428
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH21_AHUB_FIFO_CTRL_0

Offset: 0x2142c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE2_CH21_TC_STATUS_0

Offset: 0x21430
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH21_LOWER_SOURCE_ADDR_0

Offset: 0x21434
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH21_LOWER_TARGET_ADDR_0

Offset: 0x2143c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH21_TC_0

Offset: 0x21444
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH21_LOWER_DESC_ADDR_0

Offset: 0x21448
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH21_TRANSFER_STATUS_0

Offset: 0x21454
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH21_TIMESTAMP_ADDR_0

Offset: 0x21458
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH21_TIMESTAMP_STATUS_0

Offset: 0x2145c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH21_APR_0

Offset: 0x21460
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH22_CMD_0

Offset: 0x21500
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH22_SOFT_RESET_0

Offset: 0x21504
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH22_STATUS_0

Offset: 0x2150c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH22_INT_STATUS_0

Offset: 0x21510
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH22_INT_SET_0

Offset: 0x21518
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH22_INT_CLEAR_0

Offset: 0x2151c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH22_CTRL_0

Offset: 0x21524
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH22_CONFIG_0

Offset: 0x21528
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH22_AHUB_FIFO_CTRL_0

Offset: 0x2152c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE2_CH22_TC_STATUS_0

Offset: 0x21530
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH22_LOWER_SOURCE_ADDR_0

Offset: 0x21534
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH22_LOWER_TARGET_ADDR_0

Offset: 0x2153c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH22_TC_0

Offset: 0x21544
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH22_LOWER_DESC_ADDR_0

Offset: 0x21548
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH22_TRANSFER_STATUS_0

Offset: 0x21554
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH22_TIMESTAMP_ADDR_0

Offset: 0x21558
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH22_TIMESTAMP_STATUS_0

Offset: 0x2155c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH22_APR_0

Offset: 0x21560
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH23_CMD_0

Offset: 0x21600
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH23_SOFT_RESET_0

Offset: 0x21604
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH23_STATUS_0

Offset: 0x2160c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH23_INT_STATUS_0

Offset: 0x21610
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH23_INT_SET_0

Offset: 0x21618
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH23_INT_CLEAR_0

Offset: 0x2161c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH23_CTRL_0

Offset: 0x21624
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH23_CONFIG_0

Offset: 0x21628
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH23_AHUB_FIFO_CTRL_0

Offset: 0x2162c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE2_CH23_TC_STATUS_0

Offset: 0x21630
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH23_LOWER_SOURCE_ADDR_0

Offset: 0x21634
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH23_LOWER_TARGET_ADDR_0

Offset: 0x2163c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH23_TC_0

Offset: 0x21644
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH23_LOWER_DESC_ADDR_0

Offset: 0x21648
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH23_TRANSFER_STATUS_0

Offset: 0x21654
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH23_TIMESTAMP_ADDR_0

Offset: 0x21658
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH23_TIMESTAMP_STATUS_0

Offset: 0x2165c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH23_APR_0

Offset: 0x21660
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH24_CMD_0

Offset: 0x21700
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH24_SOFT_RESET_0

Offset: 0x21704
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH24_STATUS_0

Offset: 0x2170c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH24_INT_STATUS_0

Offset: 0x21710
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH24_INT_SET_0

Offset: 0x21718
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH24_INT_CLEAR_0

Offset: 0x2171c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH24_CTRL_0

Offset: 0x21724
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH24_CONFIG_0

Offset: 0x21728
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH24_AHUB_FIFO_CTRL_0

Offset: 0x2172c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE2_CH24_TC_STATUS_0

Offset: 0x21730
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH24_LOWER_SOURCE_ADDR_0

Offset: 0x21734
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH24_LOWER_TARGET_ADDR_0

Offset: 0x2173c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH24_TC_0

Offset: 0x21744
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH24_LOWER_DESC_ADDR_0

Offset: 0x21748
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH24_TRANSFER_STATUS_0

Offset: 0x21754
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH24_TIMESTAMP_ADDR_0

Offset: 0x21758
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH24_TIMESTAMP_STATUS_0

Offset: 0x2175c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH24_APR_0

Offset: 0x21760
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH25_CMD_0

Offset: 0x21800
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH25_SOFT_RESET_0

Offset: 0x21804
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH25_STATUS_0

Offset: 0x2180c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH25_INT_STATUS_0

Offset: 0x21810
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH25_INT_SET_0

Offset: 0x21818
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH25_INT_CLEAR_0

Offset: 0x2181c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH25_CTRL_0

Offset: 0x21824
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH25_CONFIG_0

Offset: 0x21828
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH25_AHUB_FIFO_CTRL_0

Offset: 0x2182c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE2_CH25_TC_STATUS_0

Offset: 0x21830
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH25_LOWER_SOURCE_ADDR_0

Offset: 0x21834
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH25_LOWER_TARGET_ADDR_0

Offset: 0x2183c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH25_TC_0

Offset: 0x21844
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH25_LOWER_DESC_ADDR_0

Offset: 0x21848
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH25_TRANSFER_STATUS_0

Offset: 0x21854
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH25_TIMESTAMP_ADDR_0

Offset: 0x21858
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH25_TIMESTAMP_STATUS_0

Offset: 0x2185c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH25_APR_0

Offset: 0x21860
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH26_CMD_0

Offset: 0x21900
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH26_SOFT_RESET_0

Offset: 0x21904
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH26_STATUS_0

Offset: 0x2190c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH26_INT_STATUS_0

Offset: 0x21910
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH26_INT_SET_0

Offset: 0x21918
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH26_INT_CLEAR_0

Offset: 0x2191c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH26_CTRL_0

Offset: 0x21924
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH26_CONFIG_0

Offset: 0x21928
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH26_AHUB_FIFO_CTRL_0

Offset: 0x2192c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE2_CH26_TC_STATUS_0

Offset: 0x21930
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH26_LOWER_SOURCE_ADDR_0

Offset: 0x21934
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH26_LOWER_TARGET_ADDR_0

Offset: 0x2193c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH26_TC_0

Offset: 0x21944
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH26_LOWER_DESC_ADDR_0

Offset: 0x21948
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH26_TRANSFER_STATUS_0

Offset: 0x21954
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH26_TIMESTAMP_ADDR_0

Offset: 0x21958
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH26_TIMESTAMP_STATUS_0

Offset: 0x2195c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH26_APR_0

Offset: 0x21960
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH27_CMD_0

Offset: 0x21a00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH27_SOFT_RESET_0

Offset: 0x21a04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH27_STATUS_0

Offset: 0x21a0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH27_INT_STATUS_0

Offset: 0x21a10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH27_INT_SET_0

Offset: 0x21a18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH27_INT_CLEAR_0

Offset: 0x21a1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH27_CTRL_0

Offset: 0x21a24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH27_CONFIG_0

Offset: 0x21a28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH27_AHUB_FIFO_CTRL_0

Offset: 0x21a2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE2_CH27_TC_STATUS_0

Offset: 0x21a30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH27_LOWER_SOURCE_ADDR_0

Offset: 0x21a34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH27_LOWER_TARGET_ADDR_0

Offset: 0x21a3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH27_TC_0

Offset: 0x21a44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH27_LOWER_DESC_ADDR_0

Offset: 0x21a48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH27_TRANSFER_STATUS_0

Offset: 0x21a54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH27_TIMESTAMP_ADDR_0

Offset: 0x21a58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH27_TIMESTAMP_STATUS_0

Offset: 0x21a5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH27_APR_0

Offset: 0x21a60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH28_CMD_0

Offset: 0x21b00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH28_SOFT_RESET_0

Offset: 0x21b04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH28_STATUS_0

Offset: 0x21b0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH28_INT_STATUS_0

Offset: 0x21b10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH28_INT_SET_0

Offset: 0x21b18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH28_INT_CLEAR_0

Offset: 0x21b1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH28_CTRL_0

Offset: 0x21b24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH28_CONFIG_0

Offset: 0x21b28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH28_AHUB_FIFO_CTRL_0

Offset: 0x21b2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE2_CH28_TC_STATUS_0

Offset: 0x21b30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH28_LOWER_SOURCE_ADDR_0

Offset: 0x21b34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH28_LOWER_TARGET_ADDR_0

Offset: 0x21b3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH28_TC_0

Offset: 0x21b44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH28_LOWER_DESC_ADDR_0

Offset: 0x21b48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH28_TRANSFER_STATUS_0

Offset: 0x21b54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH28_TIMESTAMP_ADDR_0

Offset: 0x21b58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH28_TIMESTAMP_STATUS_0

Offset: 0x21b5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH28_APR_0

Offset: 0x21b60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH29_CMD_0

Offset: 0x21c00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH29_SOFT_RESET_0

Offset: 0x21c04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH29_STATUS_0

Offset: 0x21c0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH29_INT_STATUS_0

Offset: 0x21c10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH29_INT_SET_0

Offset: 0x21c18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH29_INT_CLEAR_0

Offset: 0x21c1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH29_CTRL_0

Offset: 0x21c24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH29_CONFIG_0

Offset: 0x21c28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH29_AHUB_FIFO_CTRL_0

Offset: 0x21c2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE2_CH29_TC_STATUS_0

Offset: 0x21c30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH29_LOWER_SOURCE_ADDR_0

Offset: 0x21c34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH29_LOWER_TARGET_ADDR_0

Offset: 0x21c3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH29_TC_0

Offset: 0x21c44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH29_LOWER_DESC_ADDR_0

Offset: 0x21c48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH29_TRANSFER_STATUS_0

Offset: 0x21c54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH29_TIMESTAMP_ADDR_0

Offset: 0x21c58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH29_TIMESTAMP_STATUS_0

Offset: 0x21c5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH29_APR_0

Offset: 0x21c60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH30_CMD_0

Offset: 0x21d00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH30_SOFT_RESET_0

Offset: 0x21d04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH30_STATUS_0

Offset: 0x21d0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH30_INT_STATUS_0

Offset: 0x21d10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH30_INT_SET_0

Offset: 0x21d18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH30_INT_CLEAR_0

Offset: 0x21d1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH30_CTRL_0

Offset: 0x21d24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH30_CONFIG_0

Offset: 0x21d28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH30_AHUB_FIFO_CTRL_0

Offset: 0x21d2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE2_CH30_TC_STATUS_0

Offset: 0x21d30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH30_LOWER_SOURCE_ADDR_0

Offset: 0x21d34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH30_LOWER_TARGET_ADDR_0

Offset: 0x21d3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH30_TC_0

Offset: 0x21d44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH30_LOWER_DESC_ADDR_0

Offset: 0x21d48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH30_TRANSFER_STATUS_0

Offset: 0x21d54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH30_TIMESTAMP_ADDR_0

Offset: 0x21d58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH30_TIMESTAMP_STATUS_0

Offset: 0x21d5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH30_APR_0

Offset: 0x21d60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH31_CMD_0

Offset: 0x21e00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH31_SOFT_RESET_0

Offset: 0x21e04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH31_STATUS_0

Offset: 0x21e0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH31_INT_STATUS_0

Offset: 0x21e10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH31_INT_SET_0

Offset: 0x21e18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH31_INT_CLEAR_0

Offset: 0x21e1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH31_CTRL_0

Offset: 0x21e24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH31_CONFIG_0

Offset: 0x21e28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH31_AHUB_FIFO_CTRL_0

Offset: 0x21e2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE2_CH31_TC_STATUS_0

Offset: 0x21e30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH31_LOWER_SOURCE_ADDR_0

Offset: 0x21e34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH31_LOWER_TARGET_ADDR_0

Offset: 0x21e3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH31_TC_0

Offset: 0x21e44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH31_LOWER_DESC_ADDR_0

Offset: 0x21e48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH31_TRANSFER_STATUS_0

Offset: 0x21e54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH31_TIMESTAMP_ADDR_0

Offset: 0x21e58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH31_TIMESTAMP_STATUS_0

Offset: 0x21e5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH31_APR_0

Offset: 0x21e60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH32_CMD_0

Offset: 0x21f00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH32_SOFT_RESET_0

Offset: 0x21f04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH32_STATUS_0

Offset: 0x21f0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH32_INT_STATUS_0

Offset: 0x21f10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH32_INT_SET_0

Offset: 0x21f18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH32_INT_CLEAR_0

Offset: 0x21f1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH32_CTRL_0

Offset: 0x21f24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE2_CH32_CONFIG_0

Offset: 0x21f28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE2_CH32_AHUB_FIFO_CTRL_0

Offset: 0x21f2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE2_CH32_TC_STATUS_0

Offset: 0x21f30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE2_CH32_LOWER_SOURCE_ADDR_0

Offset: 0x21f34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH32_LOWER_TARGET_ADDR_0

Offset: 0x21f3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH32_TC_0

Offset: 0x21f44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE2_CH32_LOWER_DESC_ADDR_0

Offset: 0x21f48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE2_CH32_TRANSFER_STATUS_0

Offset: 0x21f54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE2_CH32_TIMESTAMP_ADDR_0

Offset: 0x21f58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE2_CH32_TIMESTAMP_STATUS_0

Offset: 0x21f5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE2_CH32_APR_0

Offset: 0x21f60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH1_CMD_0

Offset: 0x30000
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH1_SOFT_RESET_0

Offset: 0x30004
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH1_STATUS_0

Offset: 0x3000c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH1_INT_STATUS_0

Offset: 0x30010
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH1_INT_SET_0

Offset: 0x30018
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH1_INT_CLEAR_0

Offset: 0x3001c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH1_CTRL_0

Offset: 0x30024
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH1_CONFIG_0

Offset: 0x30028
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH1_AHUB_FIFO_CTRL_0

Offset: 0x3002c
Read/Write: RW
Parity Protection: N
Reset: 0x01010303 (0b0x00,0001,xx00,0001,xx00,0011,xx00,0011)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x1	**OVERFLOW_THRESHOLD**
21:16	0x1	**STARVATION_THRESHOLD**
13:8	0x3	**TX_FIFO_SIZE**
5:0	0x3	**RX_FIFO_SIZE**

ADMA_PAGE3_CH1_TC_STATUS_0

Offset: 0x30030
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH1_LOWER_SOURCE_ADDR_0

Offset: 0x30034
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH1_LOWER_TARGET_ADDR_0

Offset: 0x3003c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH1_TC_0

Offset: 0x30044
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH1_LOWER_DESC_ADDR_0

Offset: 0x30048
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH1_TRANSFER_STATUS_0

Offset: 0x30054
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH1_TIMESTAMP_ADDR_0

Offset: 0x30058
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH1_TIMESTAMP_STATUS_0

Offset: 0x3005c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH1_APR_0

Offset: 0x30060
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH2_CMD_0

Offset: 0x30100
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH2_SOFT_RESET_0

Offset: 0x30104
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH2_STATUS_0

Offset: 0x3010c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH2_INT_STATUS_0

Offset: 0x30110
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH2_INT_SET_0

Offset: 0x30118
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH2_INT_CLEAR_0

Offset: 0x3011c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH2_CTRL_0

Offset: 0x30124
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH2_CONFIG_0

Offset: 0x30128
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH2_AHUB_FIFO_CTRL_0

Offset: 0x3012c
Read/Write: RW
Parity Protection: N
Reset: 0x01010303 (0b0x00,0001,xx00,0001,xx00,0011,xx00,0011)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x1	**OVERFLOW_THRESHOLD**
21:16	0x1	**STARVATION_THRESHOLD**
13:8	0x3	**TX_FIFO_SIZE**
5:0	0x3	**RX_FIFO_SIZE**

ADMA_PAGE3_CH2_TC_STATUS_0

Offset: 0x30130
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH2_LOWER_SOURCE_ADDR_0

Offset: 0x30134
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH2_LOWER_TARGET_ADDR_0

Offset: 0x3013c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH2_TC_0

Offset: 0x30144
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH2_LOWER_DESC_ADDR_0

Offset: 0x30148
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH2_TRANSFER_STATUS_0

Offset: 0x30154
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH2_TIMESTAMP_ADDR_0

Offset: 0x30158
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH2_TIMESTAMP_STATUS_0

Offset: 0x3015c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH2_APR_0

Offset: 0x30160
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH3_CMD_0

Offset: 0x30200
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH3_SOFT_RESET_0

Offset: 0x30204
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH3_STATUS_0

Offset: 0x3020c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH3_INT_STATUS_0

Offset: 0x30210
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH3_INT_SET_0

Offset: 0x30218
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH3_INT_CLEAR_0

Offset: 0x3021c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH3_CTRL_0

Offset: 0x30224
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH3_CONFIG_0

Offset: 0x30228
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH3_AHUB_FIFO_CTRL_0

Offset: 0x3022c
Read/Write: RW
Parity Protection: N
Reset: 0x01010303 (0b0x00,0001,xx00,0001,xx00,0011,xx00,0011)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x1	**OVERFLOW_THRESHOLD**
21:16	0x1	**STARVATION_THRESHOLD**
13:8	0x3	**TX_FIFO_SIZE**
5:0	0x3	**RX_FIFO_SIZE**

ADMA_PAGE3_CH3_TC_STATUS_0

Offset: 0x30230
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH3_LOWER_SOURCE_ADDR_0

Offset: 0x30234
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH3_LOWER_TARGET_ADDR_0

Offset: 0x3023c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH3_TC_0

Offset: 0x30244
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH3_LOWER_DESC_ADDR_0

Offset: 0x30248
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH3_TRANSFER_STATUS_0

Offset: 0x30254
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH3_TIMESTAMP_ADDR_0

Offset: 0x30258
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH3_TIMESTAMP_STATUS_0

Offset: 0x3025c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH3_APR_0

Offset: 0x30260
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH4_CMD_0

Offset: 0x30300
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH4_SOFT_RESET_0

Offset: 0x30304
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH4_STATUS_0

Offset: 0x3030c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH4_INT_STATUS_0

Offset: 0x30310
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH4_INT_SET_0

Offset: 0x30318
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH4_INT_CLEAR_0

Offset: 0x3031c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH4_CTRL_0

Offset: 0x30324
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH4_CONFIG_0

Offset: 0x30328
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH4_AHUB_FIFO_CTRL_0

Offset: 0x3032c
Read/Write: RW
Parity Protection: N
Reset: 0x01010303 (0b0x00,0001,xx00,0001,xx00,0011,xx00,0011)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x1	**OVERFLOW_THRESHOLD**
21:16	0x1	**STARVATION_THRESHOLD**
13:8	0x3	**TX_FIFO_SIZE**
5:0	0x3	**RX_FIFO_SIZE**

ADMA_PAGE3_CH4_TC_STATUS_0

Offset: 0x30330
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH4_LOWER_SOURCE_ADDR_0

Offset: 0x30334
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH4_LOWER_TARGET_ADDR_0

Offset: 0x3033c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH4_TC_0

Offset: 0x30344
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH4_LOWER_DESC_ADDR_0

Offset: 0x30348
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH4_TRANSFER_STATUS_0

Offset: 0x30354
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH4_TIMESTAMP_ADDR_0

Offset: 0x30358
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH4_TIMESTAMP_STATUS_0

Offset: 0x3035c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH4_APR_0

Offset: 0x30360
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH5_CMD_0

Offset: 0x30400
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH5_SOFT_RESET_0

Offset: 0x30404
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH5_STATUS_0

Offset: 0x3040c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH5_INT_STATUS_0

Offset: 0x30410
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH5_INT_SET_0

Offset: 0x30418
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH5_INT_CLEAR_0

Offset: 0x3041c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH5_CTRL_0

Offset: 0x30424
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH5_CONFIG_0

Offset: 0x30428
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH5_AHUB_FIFO_CTRL_0

Offset: 0x3042c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE3_CH5_TC_STATUS_0

Offset: 0x30430
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH5_LOWER_SOURCE_ADDR_0

Offset: 0x30434
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH5_LOWER_TARGET_ADDR_0

Offset: 0x3043c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH5_TC_0

Offset: 0x30444
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH5_LOWER_DESC_ADDR_0

Offset: 0x30448
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH5_TRANSFER_STATUS_0

Offset: 0x30454
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH5_TIMESTAMP_ADDR_0

Offset: 0x30458
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH5_TIMESTAMP_STATUS_0

Offset: 0x3045c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH5_APR_0

Offset: 0x30460
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH6_CMD_0

Offset: 0x30500
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH6_SOFT_RESET_0

Offset: 0x30504
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH6_STATUS_0

Offset: 0x3050c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH6_INT_STATUS_0

Offset: 0x30510
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH6_INT_SET_0

Offset: 0x30518
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH6_INT_CLEAR_0

Offset: 0x3051c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH6_CTRL_0

Offset: 0x30524
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH6_CONFIG_0

Offset: 0x30528
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH6_AHUB_FIFO_CTRL_0

Offset: 0x3052c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE3_CH6_TC_STATUS_0

Offset: 0x30530
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH6_LOWER_SOURCE_ADDR_0

Offset: 0x30534
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH6_LOWER_TARGET_ADDR_0

Offset: 0x3053c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH6_TC_0

Offset: 0x30544
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH6_LOWER_DESC_ADDR_0

Offset: 0x30548
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH6_TRANSFER_STATUS_0

Offset: 0x30554
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH6_TIMESTAMP_ADDR_0

Offset: 0x30558
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH6_TIMESTAMP_STATUS_0

Offset: 0x3055c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH6_APR_0

Offset: 0x30560
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH7_CMD_0

Offset: 0x30600
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH7_SOFT_RESET_0

Offset: 0x30604
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH7_STATUS_0

Offset: 0x3060c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH7_INT_STATUS_0

Offset: 0x30610
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH7_INT_SET_0

Offset: 0x30618
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH7_INT_CLEAR_0

Offset: 0x3061c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH7_CTRL_0

Offset: 0x30624
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH7_CONFIG_0

Offset: 0x30628
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH7_AHUB_FIFO_CTRL_0

Offset: 0x3062c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE3_CH7_TC_STATUS_0

Offset: 0x30630
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH7_LOWER_SOURCE_ADDR_0

Offset: 0x30634
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH7_LOWER_TARGET_ADDR_0

Offset: 0x3063c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH7_TC_0

Offset: 0x30644
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH7_LOWER_DESC_ADDR_0

Offset: 0x30648
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH7_TRANSFER_STATUS_0

Offset: 0x30654
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH7_TIMESTAMP_ADDR_0

Offset: 0x30658
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH7_TIMESTAMP_STATUS_0

Offset: 0x3065c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH7_APR_0

Offset: 0x30660
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH8_CMD_0

Offset: 0x30700
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH8_SOFT_RESET_0

Offset: 0x30704
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH8_STATUS_0

Offset: 0x3070c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH8_INT_STATUS_0

Offset: 0x30710
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH8_INT_SET_0

Offset: 0x30718
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH8_INT_CLEAR_0

Offset: 0x3071c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH8_CTRL_0

Offset: 0x30724
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH8_CONFIG_0

Offset: 0x30728
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH8_AHUB_FIFO_CTRL_0

Offset: 0x3072c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE3_CH8_TC_STATUS_0

Offset: 0x30730
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH8_LOWER_SOURCE_ADDR_0

Offset: 0x30734
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH8_LOWER_TARGET_ADDR_0

Offset: 0x3073c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH8_TC_0

Offset: 0x30744
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH8_LOWER_DESC_ADDR_0

Offset: 0x30748
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH8_TRANSFER_STATUS_0

Offset: 0x30754
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH8_TIMESTAMP_ADDR_0

Offset: 0x30758
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH8_TIMESTAMP_STATUS_0

Offset: 0x3075c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH8_APR_0

Offset: 0x30760
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH9_CMD_0

Offset: 0x30800
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH9_SOFT_RESET_0

Offset: 0x30804
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH9_STATUS_0

Offset: 0x3080c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH9_INT_STATUS_0

Offset: 0x30810
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH9_INT_SET_0

Offset: 0x30818
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH9_INT_CLEAR_0

Offset: 0x3081c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH9_CTRL_0

Offset: 0x30824
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH9_CONFIG_0

Offset: 0x30828
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH9_AHUB_FIFO_CTRL_0

Offset: 0x3082c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE3_CH9_TC_STATUS_0

Offset: 0x30830
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH9_LOWER_SOURCE_ADDR_0

Offset: 0x30834
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH9_LOWER_TARGET_ADDR_0

Offset: 0x3083c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH9_TC_0

Offset: 0x30844
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH9_LOWER_DESC_ADDR_0

Offset: 0x30848
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH9_TRANSFER_STATUS_0

Offset: 0x30854
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH9_TIMESTAMP_ADDR_0

Offset: 0x30858
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH9_TIMESTAMP_STATUS_0

Offset: 0x3085c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH9_APR_0

Offset: 0x30860
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH10_CMD_0

Offset: 0x30900
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH10_SOFT_RESET_0

Offset: 0x30904
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH10_STATUS_0

Offset: 0x3090c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH10_INT_STATUS_0

Offset: 0x30910
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH10_INT_SET_0

Offset: 0x30918
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH10_INT_CLEAR_0

Offset: 0x3091c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH10_CTRL_0

Offset: 0x30924
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH10_CONFIG_0

Offset: 0x30928
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH10_AHUB_FIFO_CTRL_0

Offset: 0x3092c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE3_CH10_TC_STATUS_0

Offset: 0x30930
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH10_LOWER_SOURCE_ADDR_0

Offset: 0x30934
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH10_LOWER_TARGET_ADDR_0

Offset: 0x3093c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH10_TC_0

Offset: 0x30944
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH10_LOWER_DESC_ADDR_0

Offset: 0x30948
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH10_TRANSFER_STATUS_0

Offset: 0x30954
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH10_TIMESTAMP_ADDR_0

Offset: 0x30958
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH10_TIMESTAMP_STATUS_0

Offset: 0x3095c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH10_APR_0

Offset: 0x30960
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH11_CMD_0

Offset: 0x30a00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH11_SOFT_RESET_0

Offset: 0x30a04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH11_STATUS_0

Offset: 0x30a0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH11_INT_STATUS_0

Offset: 0x30a10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH11_INT_SET_0

Offset: 0x30a18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH11_INT_CLEAR_0

Offset: 0x30a1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH11_CTRL_0

Offset: 0x30a24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH11_CONFIG_0

Offset: 0x30a28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH11_AHUB_FIFO_CTRL_0

Offset: 0x30a2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE3_CH11_TC_STATUS_0

Offset: 0x30a30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH11_LOWER_SOURCE_ADDR_0

Offset: 0x30a34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH11_LOWER_TARGET_ADDR_0

Offset: 0x30a3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH11_TC_0

Offset: 0x30a44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH11_LOWER_DESC_ADDR_0

Offset: 0x30a48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH11_TRANSFER_STATUS_0

Offset: 0x30a54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH11_TIMESTAMP_ADDR_0

Offset: 0x30a58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH11_TIMESTAMP_STATUS_0

Offset: 0x30a5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH11_APR_0

Offset: 0x30a60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH12_CMD_0

Offset: 0x30b00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH12_SOFT_RESET_0

Offset: 0x30b04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH12_STATUS_0

Offset: 0x30b0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH12_INT_STATUS_0

Offset: 0x30b10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH12_INT_SET_0

Offset: 0x30b18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH12_INT_CLEAR_0

Offset: 0x30b1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH12_CTRL_0

Offset: 0x30b24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH12_CONFIG_0

Offset: 0x30b28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH12_AHUB_FIFO_CTRL_0

Offset: 0x30b2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE3_CH12_TC_STATUS_0

Offset: 0x30b30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH12_LOWER_SOURCE_ADDR_0

Offset: 0x30b34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH12_LOWER_TARGET_ADDR_0

Offset: 0x30b3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH12_TC_0

Offset: 0x30b44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH12_LOWER_DESC_ADDR_0

Offset: 0x30b48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH12_TRANSFER_STATUS_0

Offset: 0x30b54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH12_TIMESTAMP_ADDR_0

Offset: 0x30b58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH12_TIMESTAMP_STATUS_0

Offset: 0x30b5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH12_APR_0

Offset: 0x30b60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH13_CMD_0

Offset: 0x30c00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH13_SOFT_RESET_0

Offset: 0x30c04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH13_STATUS_0

Offset: 0x30c0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH13_INT_STATUS_0

Offset: 0x30c10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH13_INT_SET_0

Offset: 0x30c18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH13_INT_CLEAR_0

Offset: 0x30c1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH13_CTRL_0

Offset: 0x30c24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH13_CONFIG_0

Offset: 0x30c28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH13_AHUB_FIFO_CTRL_0

Offset: 0x30c2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE3_CH13_TC_STATUS_0

Offset: 0x30c30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH13_LOWER_SOURCE_ADDR_0

Offset: 0x30c34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH13_LOWER_TARGET_ADDR_0

Offset: 0x30c3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH13_TC_0

Offset: 0x30c44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH13_LOWER_DESC_ADDR_0

Offset: 0x30c48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH13_TRANSFER_STATUS_0

Offset: 0x30c54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH13_TIMESTAMP_ADDR_0

Offset: 0x30c58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH13_TIMESTAMP_STATUS_0

Offset: 0x30c5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH13_APR_0

Offset: 0x30c60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH14_CMD_0

Offset: 0x30d00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH14_SOFT_RESET_0

Offset: 0x30d04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH14_STATUS_0

Offset: 0x30d0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH14_INT_STATUS_0

Offset: 0x30d10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH14_INT_SET_0

Offset: 0x30d18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH14_INT_CLEAR_0

Offset: 0x30d1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH14_CTRL_0

Offset: 0x30d24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH14_CONFIG_0

Offset: 0x30d28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH14_AHUB_FIFO_CTRL_0

Offset: 0x30d2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE3_CH14_TC_STATUS_0

Offset: 0x30d30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH14_LOWER_SOURCE_ADDR_0

Offset: 0x30d34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH14_LOWER_TARGET_ADDR_0

Offset: 0x30d3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH14_TC_0

Offset: 0x30d44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH14_LOWER_DESC_ADDR_0

Offset: 0x30d48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH14_TRANSFER_STATUS_0

Offset: 0x30d54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH14_TIMESTAMP_ADDR_0

Offset: 0x30d58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH14_TIMESTAMP_STATUS_0

Offset: 0x30d5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH14_APR_0

Offset: 0x30d60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH15_CMD_0

Offset: 0x30e00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH15_SOFT_RESET_0

Offset: 0x30e04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH15_STATUS_0

Offset: 0x30e0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH15_INT_STATUS_0

Offset: 0x30e10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH15_INT_SET_0

Offset: 0x30e18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH15_INT_CLEAR_0

Offset: 0x30e1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH15_CTRL_0

Offset: 0x30e24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH15_CONFIG_0

Offset: 0x30e28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH15_AHUB_FIFO_CTRL_0

Offset: 0x30e2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE3_CH15_TC_STATUS_0

Offset: 0x30e30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH15_LOWER_SOURCE_ADDR_0

Offset: 0x30e34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH15_LOWER_TARGET_ADDR_0

Offset: 0x30e3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH15_TC_0

Offset: 0x30e44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH15_LOWER_DESC_ADDR_0

Offset: 0x30e48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH15_TRANSFER_STATUS_0

Offset: 0x30e54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH15_TIMESTAMP_ADDR_0

Offset: 0x30e58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH15_TIMESTAMP_STATUS_0

Offset: 0x30e5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH15_APR_0

Offset: 0x30e60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH16_CMD_0

Offset: 0x30f00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH16_SOFT_RESET_0

Offset: 0x30f04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH16_STATUS_0

Offset: 0x30f0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH16_INT_STATUS_0

Offset: 0x30f10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH16_INT_SET_0

Offset: 0x30f18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH16_INT_CLEAR_0

Offset: 0x30f1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH16_CTRL_0

Offset: 0x30f24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH16_CONFIG_0

Offset: 0x30f28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH16_AHUB_FIFO_CTRL_0

Offset: 0x30f2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE3_CH16_TC_STATUS_0

Offset: 0x30f30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH16_LOWER_SOURCE_ADDR_0

Offset: 0x30f34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH16_LOWER_TARGET_ADDR_0

Offset: 0x30f3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH16_TC_0

Offset: 0x30f44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH16_LOWER_DESC_ADDR_0

Offset: 0x30f48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH16_TRANSFER_STATUS_0

Offset: 0x30f54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH16_TIMESTAMP_ADDR_0

Offset: 0x30f58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH16_TIMESTAMP_STATUS_0

Offset: 0x30f5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH16_APR_0

Offset: 0x30f60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH17_CMD_0

Offset: 0x31000
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH17_SOFT_RESET_0

Offset: 0x31004
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH17_STATUS_0

Offset: 0x3100c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH17_INT_STATUS_0

Offset: 0x31010
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH17_INT_SET_0

Offset: 0x31018
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH17_INT_CLEAR_0

Offset: 0x3101c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH17_CTRL_0

Offset: 0x31024
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH17_CONFIG_0

Offset: 0x31028
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH17_AHUB_FIFO_CTRL_0

Offset: 0x3102c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE3_CH17_TC_STATUS_0

Offset: 0x31030
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH17_LOWER_SOURCE_ADDR_0

Offset: 0x31034
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH17_LOWER_TARGET_ADDR_0

Offset: 0x3103c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH17_TC_0

Offset: 0x31044
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH17_LOWER_DESC_ADDR_0

Offset: 0x31048
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH17_TRANSFER_STATUS_0

Offset: 0x31054
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH17_TIMESTAMP_ADDR_0

Offset: 0x31058
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH17_TIMESTAMP_STATUS_0

Offset: 0x3105c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH17_APR_0

Offset: 0x31060
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH18_CMD_0

Offset: 0x31100
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH18_SOFT_RESET_0

Offset: 0x31104
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH18_STATUS_0

Offset: 0x3110c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH18_INT_STATUS_0

Offset: 0x31110
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH18_INT_SET_0

Offset: 0x31118
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH18_INT_CLEAR_0

Offset: 0x3111c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH18_CTRL_0

Offset: 0x31124
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH18_CONFIG_0

Offset: 0x31128
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH18_AHUB_FIFO_CTRL_0

Offset: 0x3112c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE3_CH18_TC_STATUS_0

Offset: 0x31130
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH18_LOWER_SOURCE_ADDR_0

Offset: 0x31134
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH18_LOWER_TARGET_ADDR_0

Offset: 0x3113c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH18_TC_0

Offset: 0x31144
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH18_LOWER_DESC_ADDR_0

Offset: 0x31148
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH18_TRANSFER_STATUS_0

Offset: 0x31154
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH18_TIMESTAMP_ADDR_0

Offset: 0x31158
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH18_TIMESTAMP_STATUS_0

Offset: 0x3115c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH18_APR_0

Offset: 0x31160
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH19_CMD_0

Offset: 0x31200
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH19_SOFT_RESET_0

Offset: 0x31204
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH19_STATUS_0

Offset: 0x3120c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH19_INT_STATUS_0

Offset: 0x31210
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH19_INT_SET_0

Offset: 0x31218
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH19_INT_CLEAR_0

Offset: 0x3121c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH19_CTRL_0

Offset: 0x31224
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH19_CONFIG_0

Offset: 0x31228
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH19_AHUB_FIFO_CTRL_0

Offset: 0x3122c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE3_CH19_TC_STATUS_0

Offset: 0x31230
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH19_LOWER_SOURCE_ADDR_0

Offset: 0x31234
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH19_LOWER_TARGET_ADDR_0

Offset: 0x3123c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH19_TC_0

Offset: 0x31244
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH19_LOWER_DESC_ADDR_0

Offset: 0x31248
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH19_TRANSFER_STATUS_0

Offset: 0x31254
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH19_TIMESTAMP_ADDR_0

Offset: 0x31258
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH19_TIMESTAMP_STATUS_0

Offset: 0x3125c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH19_APR_0

Offset: 0x31260
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH20_CMD_0

Offset: 0x31300
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH20_SOFT_RESET_0

Offset: 0x31304
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH20_STATUS_0

Offset: 0x3130c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH20_INT_STATUS_0

Offset: 0x31310
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH20_INT_SET_0

Offset: 0x31318
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH20_INT_CLEAR_0

Offset: 0x3131c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH20_CTRL_0

Offset: 0x31324
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH20_CONFIG_0

Offset: 0x31328
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH20_AHUB_FIFO_CTRL_0

Offset: 0x3132c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE3_CH20_TC_STATUS_0

Offset: 0x31330
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH20_LOWER_SOURCE_ADDR_0

Offset: 0x31334
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH20_LOWER_TARGET_ADDR_0

Offset: 0x3133c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH20_TC_0

Offset: 0x31344
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH20_LOWER_DESC_ADDR_0

Offset: 0x31348
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH20_TRANSFER_STATUS_0

Offset: 0x31354
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH20_TIMESTAMP_ADDR_0

Offset: 0x31358
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH20_TIMESTAMP_STATUS_0

Offset: 0x3135c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH20_APR_0

Offset: 0x31360
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH21_CMD_0

Offset: 0x31400
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH21_SOFT_RESET_0

Offset: 0x31404
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH21_STATUS_0

Offset: 0x3140c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH21_INT_STATUS_0

Offset: 0x31410
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH21_INT_SET_0

Offset: 0x31418
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH21_INT_CLEAR_0

Offset: 0x3141c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH21_CTRL_0

Offset: 0x31424
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH21_CONFIG_0

Offset: 0x31428
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH21_AHUB_FIFO_CTRL_0

Offset: 0x3142c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE3_CH21_TC_STATUS_0

Offset: 0x31430
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH21_LOWER_SOURCE_ADDR_0

Offset: 0x31434
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH21_LOWER_TARGET_ADDR_0

Offset: 0x3143c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH21_TC_0

Offset: 0x31444
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH21_LOWER_DESC_ADDR_0

Offset: 0x31448
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH21_TRANSFER_STATUS_0

Offset: 0x31454
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH21_TIMESTAMP_ADDR_0

Offset: 0x31458
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH21_TIMESTAMP_STATUS_0

Offset: 0x3145c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH21_APR_0

Offset: 0x31460
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH22_CMD_0

Offset: 0x31500
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH22_SOFT_RESET_0

Offset: 0x31504
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH22_STATUS_0

Offset: 0x3150c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH22_INT_STATUS_0

Offset: 0x31510
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH22_INT_SET_0

Offset: 0x31518
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH22_INT_CLEAR_0

Offset: 0x3151c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH22_CTRL_0

Offset: 0x31524
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH22_CONFIG_0

Offset: 0x31528
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH22_AHUB_FIFO_CTRL_0

Offset: 0x3152c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE3_CH22_TC_STATUS_0

Offset: 0x31530
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH22_LOWER_SOURCE_ADDR_0

Offset: 0x31534
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH22_LOWER_TARGET_ADDR_0

Offset: 0x3153c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH22_TC_0

Offset: 0x31544
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH22_LOWER_DESC_ADDR_0

Offset: 0x31548
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH22_TRANSFER_STATUS_0

Offset: 0x31554
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH22_TIMESTAMP_ADDR_0

Offset: 0x31558
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH22_TIMESTAMP_STATUS_0

Offset: 0x3155c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH22_APR_0

Offset: 0x31560
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH23_CMD_0

Offset: 0x31600
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH23_SOFT_RESET_0

Offset: 0x31604
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH23_STATUS_0

Offset: 0x3160c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH23_INT_STATUS_0

Offset: 0x31610
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH23_INT_SET_0

Offset: 0x31618
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH23_INT_CLEAR_0

Offset: 0x3161c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH23_CTRL_0

Offset: 0x31624
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH23_CONFIG_0

Offset: 0x31628
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH23_AHUB_FIFO_CTRL_0

Offset: 0x3162c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE3_CH23_TC_STATUS_0

Offset: 0x31630
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH23_LOWER_SOURCE_ADDR_0

Offset: 0x31634
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH23_LOWER_TARGET_ADDR_0

Offset: 0x3163c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH23_TC_0

Offset: 0x31644
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH23_LOWER_DESC_ADDR_0

Offset: 0x31648
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH23_TRANSFER_STATUS_0

Offset: 0x31654
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH23_TIMESTAMP_ADDR_0

Offset: 0x31658
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH23_TIMESTAMP_STATUS_0

Offset: 0x3165c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH23_APR_0

Offset: 0x31660
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH24_CMD_0

Offset: 0x31700
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH24_SOFT_RESET_0

Offset: 0x31704
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH24_STATUS_0

Offset: 0x3170c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH24_INT_STATUS_0

Offset: 0x31710
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH24_INT_SET_0

Offset: 0x31718
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH24_INT_CLEAR_0

Offset: 0x3171c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH24_CTRL_0

Offset: 0x31724
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH24_CONFIG_0

Offset: 0x31728
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH24_AHUB_FIFO_CTRL_0

Offset: 0x3172c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE3_CH24_TC_STATUS_0

Offset: 0x31730
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH24_LOWER_SOURCE_ADDR_0

Offset: 0x31734
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH24_LOWER_TARGET_ADDR_0

Offset: 0x3173c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH24_TC_0

Offset: 0x31744
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH24_LOWER_DESC_ADDR_0

Offset: 0x31748
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH24_TRANSFER_STATUS_0

Offset: 0x31754
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH24_TIMESTAMP_ADDR_0

Offset: 0x31758
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH24_TIMESTAMP_STATUS_0

Offset: 0x3175c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH24_APR_0

Offset: 0x31760
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH25_CMD_0

Offset: 0x31800
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH25_SOFT_RESET_0

Offset: 0x31804
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH25_STATUS_0

Offset: 0x3180c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH25_INT_STATUS_0

Offset: 0x31810
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH25_INT_SET_0

Offset: 0x31818
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH25_INT_CLEAR_0

Offset: 0x3181c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH25_CTRL_0

Offset: 0x31824
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH25_CONFIG_0

Offset: 0x31828
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH25_AHUB_FIFO_CTRL_0

Offset: 0x3182c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE3_CH25_TC_STATUS_0

Offset: 0x31830
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH25_LOWER_SOURCE_ADDR_0

Offset: 0x31834
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH25_LOWER_TARGET_ADDR_0

Offset: 0x3183c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH25_TC_0

Offset: 0x31844
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH25_LOWER_DESC_ADDR_0

Offset: 0x31848
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH25_TRANSFER_STATUS_0

Offset: 0x31854
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH25_TIMESTAMP_ADDR_0

Offset: 0x31858
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH25_TIMESTAMP_STATUS_0

Offset: 0x3185c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH25_APR_0

Offset: 0x31860
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH26_CMD_0

Offset: 0x31900
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH26_SOFT_RESET_0

Offset: 0x31904
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH26_STATUS_0

Offset: 0x3190c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH26_INT_STATUS_0

Offset: 0x31910
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH26_INT_SET_0

Offset: 0x31918
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH26_INT_CLEAR_0

Offset: 0x3191c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH26_CTRL_0

Offset: 0x31924
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH26_CONFIG_0

Offset: 0x31928
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH26_AHUB_FIFO_CTRL_0

Offset: 0x3192c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE3_CH26_TC_STATUS_0

Offset: 0x31930
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH26_LOWER_SOURCE_ADDR_0

Offset: 0x31934
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH26_LOWER_TARGET_ADDR_0

Offset: 0x3193c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH26_TC_0

Offset: 0x31944
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH26_LOWER_DESC_ADDR_0

Offset: 0x31948
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH26_TRANSFER_STATUS_0

Offset: 0x31954
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH26_TIMESTAMP_ADDR_0

Offset: 0x31958
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH26_TIMESTAMP_STATUS_0

Offset: 0x3195c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH26_APR_0

Offset: 0x31960
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH27_CMD_0

Offset: 0x31a00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH27_SOFT_RESET_0

Offset: 0x31a04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH27_STATUS_0

Offset: 0x31a0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH27_INT_STATUS_0

Offset: 0x31a10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH27_INT_SET_0

Offset: 0x31a18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH27_INT_CLEAR_0

Offset: 0x31a1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH27_CTRL_0

Offset: 0x31a24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH27_CONFIG_0

Offset: 0x31a28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH27_AHUB_FIFO_CTRL_0

Offset: 0x31a2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE3_CH27_TC_STATUS_0

Offset: 0x31a30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH27_LOWER_SOURCE_ADDR_0

Offset: 0x31a34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH27_LOWER_TARGET_ADDR_0

Offset: 0x31a3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH27_TC_0

Offset: 0x31a44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH27_LOWER_DESC_ADDR_0

Offset: 0x31a48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH27_TRANSFER_STATUS_0

Offset: 0x31a54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH27_TIMESTAMP_ADDR_0

Offset: 0x31a58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH27_TIMESTAMP_STATUS_0

Offset: 0x31a5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH27_APR_0

Offset: 0x31a60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH28_CMD_0

Offset: 0x31b00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH28_SOFT_RESET_0

Offset: 0x31b04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH28_STATUS_0

Offset: 0x31b0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH28_INT_STATUS_0

Offset: 0x31b10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH28_INT_SET_0

Offset: 0x31b18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH28_INT_CLEAR_0

Offset: 0x31b1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH28_CTRL_0

Offset: 0x31b24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH28_CONFIG_0

Offset: 0x31b28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH28_AHUB_FIFO_CTRL_0

Offset: 0x31b2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE3_CH28_TC_STATUS_0

Offset: 0x31b30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH28_LOWER_SOURCE_ADDR_0

Offset: 0x31b34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH28_LOWER_TARGET_ADDR_0

Offset: 0x31b3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH28_TC_0

Offset: 0x31b44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH28_LOWER_DESC_ADDR_0

Offset: 0x31b48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH28_TRANSFER_STATUS_0

Offset: 0x31b54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH28_TIMESTAMP_ADDR_0

Offset: 0x31b58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH28_TIMESTAMP_STATUS_0

Offset: 0x31b5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH28_APR_0

Offset: 0x31b60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH29_CMD_0

Offset: 0x31c00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH29_SOFT_RESET_0

Offset: 0x31c04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH29_STATUS_0

Offset: 0x31c0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH29_INT_STATUS_0

Offset: 0x31c10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH29_INT_SET_0

Offset: 0x31c18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH29_INT_CLEAR_0

Offset: 0x31c1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH29_CTRL_0

Offset: 0x31c24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH29_CONFIG_0

Offset: 0x31c28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH29_AHUB_FIFO_CTRL_0

Offset: 0x31c2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE3_CH29_TC_STATUS_0

Offset: 0x31c30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH29_LOWER_SOURCE_ADDR_0

Offset: 0x31c34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH29_LOWER_TARGET_ADDR_0

Offset: 0x31c3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH29_TC_0

Offset: 0x31c44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH29_LOWER_DESC_ADDR_0

Offset: 0x31c48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH29_TRANSFER_STATUS_0

Offset: 0x31c54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH29_TIMESTAMP_ADDR_0

Offset: 0x31c58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH29_TIMESTAMP_STATUS_0

Offset: 0x31c5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH29_APR_0

Offset: 0x31c60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH30_CMD_0

Offset: 0x31d00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH30_SOFT_RESET_0

Offset: 0x31d04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH30_STATUS_0

Offset: 0x31d0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH30_INT_STATUS_0

Offset: 0x31d10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH30_INT_SET_0

Offset: 0x31d18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH30_INT_CLEAR_0

Offset: 0x31d1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH30_CTRL_0

Offset: 0x31d24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH30_CONFIG_0

Offset: 0x31d28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH30_AHUB_FIFO_CTRL_0

Offset: 0x31d2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE3_CH30_TC_STATUS_0

Offset: 0x31d30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH30_LOWER_SOURCE_ADDR_0

Offset: 0x31d34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH30_LOWER_TARGET_ADDR_0

Offset: 0x31d3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH30_TC_0

Offset: 0x31d44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH30_LOWER_DESC_ADDR_0

Offset: 0x31d48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH30_TRANSFER_STATUS_0

Offset: 0x31d54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH30_TIMESTAMP_ADDR_0

Offset: 0x31d58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH30_TIMESTAMP_STATUS_0

Offset: 0x31d5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH30_APR_0

Offset: 0x31d60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH31_CMD_0

Offset: 0x31e00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH31_SOFT_RESET_0

Offset: 0x31e04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH31_STATUS_0

Offset: 0x31e0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH31_INT_STATUS_0

Offset: 0x31e10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH31_INT_SET_0

Offset: 0x31e18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH31_INT_CLEAR_0

Offset: 0x31e1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH31_CTRL_0

Offset: 0x31e24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH31_CONFIG_0

Offset: 0x31e28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH31_AHUB_FIFO_CTRL_0

Offset: 0x31e2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE3_CH31_TC_STATUS_0

Offset: 0x31e30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH31_LOWER_SOURCE_ADDR_0

Offset: 0x31e34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH31_LOWER_TARGET_ADDR_0

Offset: 0x31e3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH31_TC_0

Offset: 0x31e44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH31_LOWER_DESC_ADDR_0

Offset: 0x31e48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH31_TRANSFER_STATUS_0

Offset: 0x31e54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH31_TIMESTAMP_ADDR_0

Offset: 0x31e58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH31_TIMESTAMP_STATUS_0

Offset: 0x31e5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH31_APR_0

Offset: 0x31e60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH32_CMD_0

Offset: 0x31f00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH32_SOFT_RESET_0

Offset: 0x31f04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH32_STATUS_0

Offset: 0x31f0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH32_INT_STATUS_0

Offset: 0x31f10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH32_INT_SET_0

Offset: 0x31f18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH32_INT_CLEAR_0

Offset: 0x31f1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH32_CTRL_0

Offset: 0x31f24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE3_CH32_CONFIG_0

Offset: 0x31f28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE3_CH32_AHUB_FIFO_CTRL_0

Offset: 0x31f2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE3_CH32_TC_STATUS_0

Offset: 0x31f30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE3_CH32_LOWER_SOURCE_ADDR_0

Offset: 0x31f34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH32_LOWER_TARGET_ADDR_0

Offset: 0x31f3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH32_TC_0

Offset: 0x31f44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE3_CH32_LOWER_DESC_ADDR_0

Offset: 0x31f48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE3_CH32_TRANSFER_STATUS_0

Offset: 0x31f54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE3_CH32_TIMESTAMP_ADDR_0

Offset: 0x31f58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE3_CH32_TIMESTAMP_STATUS_0

Offset: 0x31f5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE3_CH32_APR_0

Offset: 0x31f60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH1_CMD_0

Offset: 0x40000
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH1_SOFT_RESET_0

Offset: 0x40004
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH1_STATUS_0

Offset: 0x4000c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH1_INT_STATUS_0

Offset: 0x40010
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH1_INT_SET_0

Offset: 0x40018
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH1_INT_CLEAR_0

Offset: 0x4001c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH1_CTRL_0

Offset: 0x40024
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH1_CONFIG_0

Offset: 0x40028
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH1_AHUB_FIFO_CTRL_0

Offset: 0x4002c
Read/Write: RW
Parity Protection: N
Reset: 0x01010303 (0b0x00,0001,xx00,0001,xx00,0011,xx00,0011)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x1	**OVERFLOW_THRESHOLD**
21:16	0x1	**STARVATION_THRESHOLD**
13:8	0x3	**TX_FIFO_SIZE**
5:0	0x3	**RX_FIFO_SIZE**

ADMA_PAGE4_CH1_TC_STATUS_0

Offset: 0x40030
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH1_LOWER_SOURCE_ADDR_0

Offset: 0x40034
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH1_LOWER_TARGET_ADDR_0

Offset: 0x4003c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH1_TC_0

Offset: 0x40044
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH1_LOWER_DESC_ADDR_0

Offset: 0x40048
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH1_TRANSFER_STATUS_0

Offset: 0x40054
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH1_TIMESTAMP_ADDR_0

Offset: 0x40058
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH1_TIMESTAMP_STATUS_0

Offset: 0x4005c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH1_APR_0

Offset: 0x40060
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH2_CMD_0

Offset: 0x40100
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH2_SOFT_RESET_0

Offset: 0x40104
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH2_STATUS_0

Offset: 0x4010c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH2_INT_STATUS_0

Offset: 0x40110
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH2_INT_SET_0

Offset: 0x40118
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH2_INT_CLEAR_0

Offset: 0x4011c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH2_CTRL_0

Offset: 0x40124
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH2_CONFIG_0

Offset: 0x40128
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH2_AHUB_FIFO_CTRL_0

Offset: 0x4012c
Read/Write: RW
Parity Protection: N
Reset: 0x01010303 (0b0x00,0001,xx00,0001,xx00,0011,xx00,0011)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x1	**OVERFLOW_THRESHOLD**
21:16	0x1	**STARVATION_THRESHOLD**
13:8	0x3	**TX_FIFO_SIZE**
5:0	0x3	**RX_FIFO_SIZE**

ADMA_PAGE4_CH2_TC_STATUS_0

Offset: 0x40130
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH2_LOWER_SOURCE_ADDR_0

Offset: 0x40134
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH2_LOWER_TARGET_ADDR_0

Offset: 0x4013c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH2_TC_0

Offset: 0x40144
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH2_LOWER_DESC_ADDR_0

Offset: 0x40148
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH2_TRANSFER_STATUS_0

Offset: 0x40154
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH2_TIMESTAMP_ADDR_0

Offset: 0x40158
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH2_TIMESTAMP_STATUS_0

Offset: 0x4015c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH2_APR_0

Offset: 0x40160
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH3_CMD_0

Offset: 0x40200
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH3_SOFT_RESET_0

Offset: 0x40204
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH3_STATUS_0

Offset: 0x4020c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH3_INT_STATUS_0

Offset: 0x40210
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH3_INT_SET_0

Offset: 0x40218
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH3_INT_CLEAR_0

Offset: 0x4021c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH3_CTRL_0

Offset: 0x40224
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH3_CONFIG_0

Offset: 0x40228
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH3_AHUB_FIFO_CTRL_0

Offset: 0x4022c
Read/Write: RW
Parity Protection: N
Reset: 0x01010303 (0b0x00,0001,xx00,0001,xx00,0011,xx00,0011)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x1	**OVERFLOW_THRESHOLD**
21:16	0x1	**STARVATION_THRESHOLD**
13:8	0x3	**TX_FIFO_SIZE**
5:0	0x3	**RX_FIFO_SIZE**

ADMA_PAGE4_CH3_TC_STATUS_0

Offset: 0x40230
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH3_LOWER_SOURCE_ADDR_0

Offset: 0x40234
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH3_LOWER_TARGET_ADDR_0

Offset: 0x4023c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH3_TC_0

Offset: 0x40244
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH3_LOWER_DESC_ADDR_0

Offset: 0x40248
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH3_TRANSFER_STATUS_0

Offset: 0x40254
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH3_TIMESTAMP_ADDR_0

Offset: 0x40258
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH3_TIMESTAMP_STATUS_0

Offset: 0x4025c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH3_APR_0

Offset: 0x40260
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH4_CMD_0

Offset: 0x40300
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH4_SOFT_RESET_0

Offset: 0x40304
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH4_STATUS_0

Offset: 0x4030c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH4_INT_STATUS_0

Offset: 0x40310
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH4_INT_SET_0

Offset: 0x40318
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH4_INT_CLEAR_0

Offset: 0x4031c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH4_CTRL_0

Offset: 0x40324
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH4_CONFIG_0

Offset: 0x40328
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH4_AHUB_FIFO_CTRL_0

Offset: 0x4032c
Read/Write: RW
Parity Protection: N
Reset: 0x01010303 (0b0x00,0001,xx00,0001,xx00,0011,xx00,0011)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x1	**OVERFLOW_THRESHOLD**
21:16	0x1	**STARVATION_THRESHOLD**
13:8	0x3	**TX_FIFO_SIZE**
5:0	0x3	**RX_FIFO_SIZE**

ADMA_PAGE4_CH4_TC_STATUS_0

Offset: 0x40330
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH4_LOWER_SOURCE_ADDR_0

Offset: 0x40334
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH4_LOWER_TARGET_ADDR_0

Offset: 0x4033c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH4_TC_0

Offset: 0x40344
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH4_LOWER_DESC_ADDR_0

Offset: 0x40348
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH4_TRANSFER_STATUS_0

Offset: 0x40354
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH4_TIMESTAMP_ADDR_0

Offset: 0x40358
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH4_TIMESTAMP_STATUS_0

Offset: 0x4035c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH4_APR_0

Offset: 0x40360
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH5_CMD_0

Offset: 0x40400
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH5_SOFT_RESET_0

Offset: 0x40404
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH5_STATUS_0

Offset: 0x4040c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH5_INT_STATUS_0

Offset: 0x40410
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH5_INT_SET_0

Offset: 0x40418
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH5_INT_CLEAR_0

Offset: 0x4041c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH5_CTRL_0

Offset: 0x40424
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH5_CONFIG_0

Offset: 0x40428
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH5_AHUB_FIFO_CTRL_0

Offset: 0x4042c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE4_CH5_TC_STATUS_0

Offset: 0x40430
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH5_LOWER_SOURCE_ADDR_0

Offset: 0x40434
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH5_LOWER_TARGET_ADDR_0

Offset: 0x4043c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH5_TC_0

Offset: 0x40444
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH5_LOWER_DESC_ADDR_0

Offset: 0x40448
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH5_TRANSFER_STATUS_0

Offset: 0x40454
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH5_TIMESTAMP_ADDR_0

Offset: 0x40458
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH5_TIMESTAMP_STATUS_0

Offset: 0x4045c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH5_APR_0

Offset: 0x40460
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH6_CMD_0

Offset: 0x40500
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH6_SOFT_RESET_0

Offset: 0x40504
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH6_STATUS_0

Offset: 0x4050c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH6_INT_STATUS_0

Offset: 0x40510
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH6_INT_SET_0

Offset: 0x40518
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH6_INT_CLEAR_0

Offset: 0x4051c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH6_CTRL_0

Offset: 0x40524
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH6_CONFIG_0

Offset: 0x40528
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH6_AHUB_FIFO_CTRL_0

Offset: 0x4052c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE4_CH6_TC_STATUS_0

Offset: 0x40530
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH6_LOWER_SOURCE_ADDR_0

Offset: 0x40534
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH6_LOWER_TARGET_ADDR_0

Offset: 0x4053c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH6_TC_0

Offset: 0x40544
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH6_LOWER_DESC_ADDR_0

Offset: 0x40548
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH6_TRANSFER_STATUS_0

Offset: 0x40554
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH6_TIMESTAMP_ADDR_0

Offset: 0x40558
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH6_TIMESTAMP_STATUS_0

Offset: 0x4055c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH6_APR_0

Offset: 0x40560
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH7_CMD_0

Offset: 0x40600
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH7_SOFT_RESET_0

Offset: 0x40604
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH7_STATUS_0

Offset: 0x4060c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH7_INT_STATUS_0

Offset: 0x40610
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH7_INT_SET_0

Offset: 0x40618
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH7_INT_CLEAR_0

Offset: 0x4061c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH7_CTRL_0

Offset: 0x40624
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH7_CONFIG_0

Offset: 0x40628
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH7_AHUB_FIFO_CTRL_0

Offset: 0x4062c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE4_CH7_TC_STATUS_0

Offset: 0x40630
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH7_LOWER_SOURCE_ADDR_0

Offset: 0x40634
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH7_LOWER_TARGET_ADDR_0

Offset: 0x4063c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH7_TC_0

Offset: 0x40644
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH7_LOWER_DESC_ADDR_0

Offset: 0x40648
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH7_TRANSFER_STATUS_0

Offset: 0x40654
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH7_TIMESTAMP_ADDR_0

Offset: 0x40658
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH7_TIMESTAMP_STATUS_0

Offset: 0x4065c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH7_APR_0

Offset: 0x40660
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH8_CMD_0

Offset: 0x40700
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH8_SOFT_RESET_0

Offset: 0x40704
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH8_STATUS_0

Offset: 0x4070c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH8_INT_STATUS_0

Offset: 0x40710
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH8_INT_SET_0

Offset: 0x40718
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH8_INT_CLEAR_0

Offset: 0x4071c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH8_CTRL_0

Offset: 0x40724
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH8_CONFIG_0

Offset: 0x40728
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH8_AHUB_FIFO_CTRL_0

Offset: 0x4072c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE4_CH8_TC_STATUS_0

Offset: 0x40730
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH8_LOWER_SOURCE_ADDR_0

Offset: 0x40734
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH8_LOWER_TARGET_ADDR_0

Offset: 0x4073c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH8_TC_0

Offset: 0x40744
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH8_LOWER_DESC_ADDR_0

Offset: 0x40748
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH8_TRANSFER_STATUS_0

Offset: 0x40754
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH8_TIMESTAMP_ADDR_0

Offset: 0x40758
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH8_TIMESTAMP_STATUS_0

Offset: 0x4075c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH8_APR_0

Offset: 0x40760
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH9_CMD_0

Offset: 0x40800
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH9_SOFT_RESET_0

Offset: 0x40804
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH9_STATUS_0

Offset: 0x4080c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH9_INT_STATUS_0

Offset: 0x40810
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH9_INT_SET_0

Offset: 0x40818
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH9_INT_CLEAR_0

Offset: 0x4081c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH9_CTRL_0

Offset: 0x40824
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH9_CONFIG_0

Offset: 0x40828
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH9_AHUB_FIFO_CTRL_0

Offset: 0x4082c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE4_CH9_TC_STATUS_0

Offset: 0x40830
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH9_LOWER_SOURCE_ADDR_0

Offset: 0x40834
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH9_LOWER_TARGET_ADDR_0

Offset: 0x4083c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH9_TC_0

Offset: 0x40844
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH9_LOWER_DESC_ADDR_0

Offset: 0x40848
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH9_TRANSFER_STATUS_0

Offset: 0x40854
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH9_TIMESTAMP_ADDR_0

Offset: 0x40858
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH9_TIMESTAMP_STATUS_0

Offset: 0x4085c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH9_APR_0

Offset: 0x40860
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH10_CMD_0

Offset: 0x40900
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH10_SOFT_RESET_0

Offset: 0x40904
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH10_STATUS_0

Offset: 0x4090c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH10_INT_STATUS_0

Offset: 0x40910
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH10_INT_SET_0

Offset: 0x40918
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH10_INT_CLEAR_0

Offset: 0x4091c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH10_CTRL_0

Offset: 0x40924
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH10_CONFIG_0

Offset: 0x40928
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH10_AHUB_FIFO_CTRL_0

Offset: 0x4092c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE4_CH10_TC_STATUS_0

Offset: 0x40930
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH10_LOWER_SOURCE_ADDR_0

Offset: 0x40934
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH10_LOWER_TARGET_ADDR_0

Offset: 0x4093c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH10_TC_0

Offset: 0x40944
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH10_LOWER_DESC_ADDR_0

Offset: 0x40948
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH10_TRANSFER_STATUS_0

Offset: 0x40954
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH10_TIMESTAMP_ADDR_0

Offset: 0x40958
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH10_TIMESTAMP_STATUS_0

Offset: 0x4095c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH10_APR_0

Offset: 0x40960
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH11_CMD_0

Offset: 0x40a00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH11_SOFT_RESET_0

Offset: 0x40a04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH11_STATUS_0

Offset: 0x40a0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH11_INT_STATUS_0

Offset: 0x40a10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH11_INT_SET_0

Offset: 0x40a18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH11_INT_CLEAR_0

Offset: 0x40a1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH11_CTRL_0

Offset: 0x40a24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH11_CONFIG_0

Offset: 0x40a28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH11_AHUB_FIFO_CTRL_0

Offset: 0x40a2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE4_CH11_TC_STATUS_0

Offset: 0x40a30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH11_LOWER_SOURCE_ADDR_0

Offset: 0x40a34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH11_LOWER_TARGET_ADDR_0

Offset: 0x40a3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH11_TC_0

Offset: 0x40a44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH11_LOWER_DESC_ADDR_0

Offset: 0x40a48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH11_TRANSFER_STATUS_0

Offset: 0x40a54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH11_TIMESTAMP_ADDR_0

Offset: 0x40a58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH11_TIMESTAMP_STATUS_0

Offset: 0x40a5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH11_APR_0

Offset: 0x40a60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH12_CMD_0

Offset: 0x40b00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH12_SOFT_RESET_0

Offset: 0x40b04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH12_STATUS_0

Offset: 0x40b0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH12_INT_STATUS_0

Offset: 0x40b10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH12_INT_SET_0

Offset: 0x40b18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH12_INT_CLEAR_0

Offset: 0x40b1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH12_CTRL_0

Offset: 0x40b24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH12_CONFIG_0

Offset: 0x40b28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH12_AHUB_FIFO_CTRL_0

Offset: 0x40b2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE4_CH12_TC_STATUS_0

Offset: 0x40b30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH12_LOWER_SOURCE_ADDR_0

Offset: 0x40b34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH12_LOWER_TARGET_ADDR_0

Offset: 0x40b3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH12_TC_0

Offset: 0x40b44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH12_LOWER_DESC_ADDR_0

Offset: 0x40b48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH12_TRANSFER_STATUS_0

Offset: 0x40b54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH12_TIMESTAMP_ADDR_0

Offset: 0x40b58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH12_TIMESTAMP_STATUS_0

Offset: 0x40b5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH12_APR_0

Offset: 0x40b60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH13_CMD_0

Offset: 0x40c00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH13_SOFT_RESET_0

Offset: 0x40c04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH13_STATUS_0

Offset: 0x40c0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH13_INT_STATUS_0

Offset: 0x40c10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH13_INT_SET_0

Offset: 0x40c18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH13_INT_CLEAR_0

Offset: 0x40c1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH13_CTRL_0

Offset: 0x40c24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH13_CONFIG_0

Offset: 0x40c28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH13_AHUB_FIFO_CTRL_0

Offset: 0x40c2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE4_CH13_TC_STATUS_0

Offset: 0x40c30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH13_LOWER_SOURCE_ADDR_0

Offset: 0x40c34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH13_LOWER_TARGET_ADDR_0

Offset: 0x40c3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH13_TC_0

Offset: 0x40c44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH13_LOWER_DESC_ADDR_0

Offset: 0x40c48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH13_TRANSFER_STATUS_0

Offset: 0x40c54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH13_TIMESTAMP_ADDR_0

Offset: 0x40c58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH13_TIMESTAMP_STATUS_0

Offset: 0x40c5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH13_APR_0

Offset: 0x40c60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH14_CMD_0

Offset: 0x40d00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH14_SOFT_RESET_0

Offset: 0x40d04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH14_STATUS_0

Offset: 0x40d0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH14_INT_STATUS_0

Offset: 0x40d10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH14_INT_SET_0

Offset: 0x40d18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH14_INT_CLEAR_0

Offset: 0x40d1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH14_CTRL_0

Offset: 0x40d24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH14_CONFIG_0

Offset: 0x40d28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH14_AHUB_FIFO_CTRL_0

Offset: 0x40d2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE4_CH14_TC_STATUS_0

Offset: 0x40d30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH14_LOWER_SOURCE_ADDR_0

Offset: 0x40d34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH14_LOWER_TARGET_ADDR_0

Offset: 0x40d3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH14_TC_0

Offset: 0x40d44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH14_LOWER_DESC_ADDR_0

Offset: 0x40d48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH14_TRANSFER_STATUS_0

Offset: 0x40d54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH14_TIMESTAMP_ADDR_0

Offset: 0x40d58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH14_TIMESTAMP_STATUS_0

Offset: 0x40d5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH14_APR_0

Offset: 0x40d60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH15_CMD_0

Offset: 0x40e00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH15_SOFT_RESET_0

Offset: 0x40e04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH15_STATUS_0

Offset: 0x40e0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH15_INT_STATUS_0

Offset: 0x40e10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH15_INT_SET_0

Offset: 0x40e18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH15_INT_CLEAR_0

Offset: 0x40e1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH15_CTRL_0

Offset: 0x40e24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH15_CONFIG_0

Offset: 0x40e28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH15_AHUB_FIFO_CTRL_0

Offset: 0x40e2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE4_CH15_TC_STATUS_0

Offset: 0x40e30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH15_LOWER_SOURCE_ADDR_0

Offset: 0x40e34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH15_LOWER_TARGET_ADDR_0

Offset: 0x40e3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH15_TC_0

Offset: 0x40e44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH15_LOWER_DESC_ADDR_0

Offset: 0x40e48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH15_TRANSFER_STATUS_0

Offset: 0x40e54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH15_TIMESTAMP_ADDR_0

Offset: 0x40e58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH15_TIMESTAMP_STATUS_0

Offset: 0x40e5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH15_APR_0

Offset: 0x40e60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH16_CMD_0

Offset: 0x40f00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH16_SOFT_RESET_0

Offset: 0x40f04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH16_STATUS_0

Offset: 0x40f0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH16_INT_STATUS_0

Offset: 0x40f10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH16_INT_SET_0

Offset: 0x40f18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH16_INT_CLEAR_0

Offset: 0x40f1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH16_CTRL_0

Offset: 0x40f24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH16_CONFIG_0

Offset: 0x40f28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH16_AHUB_FIFO_CTRL_0

Offset: 0x40f2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE4_CH16_TC_STATUS_0

Offset: 0x40f30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH16_LOWER_SOURCE_ADDR_0

Offset: 0x40f34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH16_LOWER_TARGET_ADDR_0

Offset: 0x40f3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH16_TC_0

Offset: 0x40f44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH16_LOWER_DESC_ADDR_0

Offset: 0x40f48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH16_TRANSFER_STATUS_0

Offset: 0x40f54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH16_TIMESTAMP_ADDR_0

Offset: 0x40f58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH16_TIMESTAMP_STATUS_0

Offset: 0x40f5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH16_APR_0

Offset: 0x40f60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH17_CMD_0

Offset: 0x41000
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH17_SOFT_RESET_0

Offset: 0x41004
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH17_STATUS_0

Offset: 0x4100c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH17_INT_STATUS_0

Offset: 0x41010
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH17_INT_SET_0

Offset: 0x41018
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH17_INT_CLEAR_0

Offset: 0x4101c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH17_CTRL_0

Offset: 0x41024
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH17_CONFIG_0

Offset: 0x41028
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH17_AHUB_FIFO_CTRL_0

Offset: 0x4102c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE4_CH17_TC_STATUS_0

Offset: 0x41030
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH17_LOWER_SOURCE_ADDR_0

Offset: 0x41034
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH17_LOWER_TARGET_ADDR_0

Offset: 0x4103c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH17_TC_0

Offset: 0x41044
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH17_LOWER_DESC_ADDR_0

Offset: 0x41048
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH17_TRANSFER_STATUS_0

Offset: 0x41054
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH17_TIMESTAMP_ADDR_0

Offset: 0x41058
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH17_TIMESTAMP_STATUS_0

Offset: 0x4105c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH17_APR_0

Offset: 0x41060
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH18_CMD_0

Offset: 0x41100
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH18_SOFT_RESET_0

Offset: 0x41104
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH18_STATUS_0

Offset: 0x4110c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH18_INT_STATUS_0

Offset: 0x41110
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH18_INT_SET_0

Offset: 0x41118
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH18_INT_CLEAR_0

Offset: 0x4111c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH18_CTRL_0

Offset: 0x41124
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH18_CONFIG_0

Offset: 0x41128
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH18_AHUB_FIFO_CTRL_0

Offset: 0x4112c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE4_CH18_TC_STATUS_0

Offset: 0x41130
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH18_LOWER_SOURCE_ADDR_0

Offset: 0x41134
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH18_LOWER_TARGET_ADDR_0

Offset: 0x4113c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH18_TC_0

Offset: 0x41144
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH18_LOWER_DESC_ADDR_0

Offset: 0x41148
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH18_TRANSFER_STATUS_0

Offset: 0x41154
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH18_TIMESTAMP_ADDR_0

Offset: 0x41158
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH18_TIMESTAMP_STATUS_0

Offset: 0x4115c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH18_APR_0

Offset: 0x41160
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH19_CMD_0

Offset: 0x41200
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH19_SOFT_RESET_0

Offset: 0x41204
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH19_STATUS_0

Offset: 0x4120c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH19_INT_STATUS_0

Offset: 0x41210
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH19_INT_SET_0

Offset: 0x41218
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH19_INT_CLEAR_0

Offset: 0x4121c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH19_CTRL_0

Offset: 0x41224
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH19_CONFIG_0

Offset: 0x41228
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH19_AHUB_FIFO_CTRL_0

Offset: 0x4122c
Read/Write: RW
Parity Protection: N
Reset: 0x00000202 (0b0x00,0000,xx00,0000,xx00,0010,xx00,0010)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x2	**TX_FIFO_SIZE**
5:0	0x2	**RX_FIFO_SIZE**

ADMA_PAGE4_CH19_TC_STATUS_0

Offset: 0x41230
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH19_LOWER_SOURCE_ADDR_0

Offset: 0x41234
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH19_LOWER_TARGET_ADDR_0

Offset: 0x4123c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH19_TC_0

Offset: 0x41244
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH19_LOWER_DESC_ADDR_0

Offset: 0x41248
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH19_TRANSFER_STATUS_0

Offset: 0x41254
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH19_TIMESTAMP_ADDR_0

Offset: 0x41258
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH19_TIMESTAMP_STATUS_0

Offset: 0x4125c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH19_APR_0

Offset: 0x41260
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH20_CMD_0

Offset: 0x41300
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH20_SOFT_RESET_0

Offset: 0x41304
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH20_STATUS_0

Offset: 0x4130c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH20_INT_STATUS_0

Offset: 0x41310
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH20_INT_SET_0

Offset: 0x41318
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH20_INT_CLEAR_0

Offset: 0x4131c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH20_CTRL_0

Offset: 0x41324
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH20_CONFIG_0

Offset: 0x41328
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH20_AHUB_FIFO_CTRL_0

Offset: 0x4132c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE4_CH20_TC_STATUS_0

Offset: 0x41330
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH20_LOWER_SOURCE_ADDR_0

Offset: 0x41334
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH20_LOWER_TARGET_ADDR_0

Offset: 0x4133c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH20_TC_0

Offset: 0x41344
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH20_LOWER_DESC_ADDR_0

Offset: 0x41348
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH20_TRANSFER_STATUS_0

Offset: 0x41354
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH20_TIMESTAMP_ADDR_0

Offset: 0x41358
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH20_TIMESTAMP_STATUS_0

Offset: 0x4135c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH20_APR_0

Offset: 0x41360
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH21_CMD_0

Offset: 0x41400
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH21_SOFT_RESET_0

Offset: 0x41404
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH21_STATUS_0

Offset: 0x4140c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH21_INT_STATUS_0

Offset: 0x41410
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH21_INT_SET_0

Offset: 0x41418
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH21_INT_CLEAR_0

Offset: 0x4141c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH21_CTRL_0

Offset: 0x41424
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH21_CONFIG_0

Offset: 0x41428
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH21_AHUB_FIFO_CTRL_0

Offset: 0x4142c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE4_CH21_TC_STATUS_0

Offset: 0x41430
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH21_LOWER_SOURCE_ADDR_0

Offset: 0x41434
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH21_LOWER_TARGET_ADDR_0

Offset: 0x4143c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH21_TC_0

Offset: 0x41444
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH21_LOWER_DESC_ADDR_0

Offset: 0x41448
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH21_TRANSFER_STATUS_0

Offset: 0x41454
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH21_TIMESTAMP_ADDR_0

Offset: 0x41458
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH21_TIMESTAMP_STATUS_0

Offset: 0x4145c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH21_APR_0

Offset: 0x41460
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH22_CMD_0

Offset: 0x41500
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH22_SOFT_RESET_0

Offset: 0x41504
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH22_STATUS_0

Offset: 0x4150c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH22_INT_STATUS_0

Offset: 0x41510
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH22_INT_SET_0

Offset: 0x41518
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH22_INT_CLEAR_0

Offset: 0x4151c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH22_CTRL_0

Offset: 0x41524
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH22_CONFIG_0

Offset: 0x41528
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH22_AHUB_FIFO_CTRL_0

Offset: 0x4152c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE4_CH22_TC_STATUS_0

Offset: 0x41530
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH22_LOWER_SOURCE_ADDR_0

Offset: 0x41534
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH22_LOWER_TARGET_ADDR_0

Offset: 0x4153c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH22_TC_0

Offset: 0x41544
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH22_LOWER_DESC_ADDR_0

Offset: 0x41548
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH22_TRANSFER_STATUS_0

Offset: 0x41554
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH22_TIMESTAMP_ADDR_0

Offset: 0x41558
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH22_TIMESTAMP_STATUS_0

Offset: 0x4155c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH22_APR_0

Offset: 0x41560
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH23_CMD_0

Offset: 0x41600
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH23_SOFT_RESET_0

Offset: 0x41604
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH23_STATUS_0

Offset: 0x4160c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH23_INT_STATUS_0

Offset: 0x41610
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH23_INT_SET_0

Offset: 0x41618
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH23_INT_CLEAR_0

Offset: 0x4161c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH23_CTRL_0

Offset: 0x41624
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH23_CONFIG_0

Offset: 0x41628
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH23_AHUB_FIFO_CTRL_0

Offset: 0x4162c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE4_CH23_TC_STATUS_0

Offset: 0x41630
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH23_LOWER_SOURCE_ADDR_0

Offset: 0x41634
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH23_LOWER_TARGET_ADDR_0

Offset: 0x4163c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH23_TC_0

Offset: 0x41644
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH23_LOWER_DESC_ADDR_0

Offset: 0x41648
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH23_TRANSFER_STATUS_0

Offset: 0x41654
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH23_TIMESTAMP_ADDR_0

Offset: 0x41658
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH23_TIMESTAMP_STATUS_0

Offset: 0x4165c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH23_APR_0

Offset: 0x41660
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH24_CMD_0

Offset: 0x41700
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH24_SOFT_RESET_0

Offset: 0x41704
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH24_STATUS_0

Offset: 0x4170c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH24_INT_STATUS_0

Offset: 0x41710
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH24_INT_SET_0

Offset: 0x41718
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH24_INT_CLEAR_0

Offset: 0x4171c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH24_CTRL_0

Offset: 0x41724
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH24_CONFIG_0

Offset: 0x41728
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH24_AHUB_FIFO_CTRL_0

Offset: 0x4172c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE4_CH24_TC_STATUS_0

Offset: 0x41730
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH24_LOWER_SOURCE_ADDR_0

Offset: 0x41734
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH24_LOWER_TARGET_ADDR_0

Offset: 0x4173c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH24_TC_0

Offset: 0x41744
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH24_LOWER_DESC_ADDR_0

Offset: 0x41748
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH24_TRANSFER_STATUS_0

Offset: 0x41754
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH24_TIMESTAMP_ADDR_0

Offset: 0x41758
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH24_TIMESTAMP_STATUS_0

Offset: 0x4175c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH24_APR_0

Offset: 0x41760
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH25_CMD_0

Offset: 0x41800
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH25_SOFT_RESET_0

Offset: 0x41804
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH25_STATUS_0

Offset: 0x4180c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH25_INT_STATUS_0

Offset: 0x41810
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH25_INT_SET_0

Offset: 0x41818
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH25_INT_CLEAR_0

Offset: 0x4181c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH25_CTRL_0

Offset: 0x41824
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH25_CONFIG_0

Offset: 0x41828
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH25_AHUB_FIFO_CTRL_0

Offset: 0x4182c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE4_CH25_TC_STATUS_0

Offset: 0x41830
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH25_LOWER_SOURCE_ADDR_0

Offset: 0x41834
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH25_LOWER_TARGET_ADDR_0

Offset: 0x4183c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH25_TC_0

Offset: 0x41844
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH25_LOWER_DESC_ADDR_0

Offset: 0x41848
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH25_TRANSFER_STATUS_0

Offset: 0x41854
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH25_TIMESTAMP_ADDR_0

Offset: 0x41858
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH25_TIMESTAMP_STATUS_0

Offset: 0x4185c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH25_APR_0

Offset: 0x41860
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH26_CMD_0

Offset: 0x41900
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH26_SOFT_RESET_0

Offset: 0x41904
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH26_STATUS_0

Offset: 0x4190c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH26_INT_STATUS_0

Offset: 0x41910
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH26_INT_SET_0

Offset: 0x41918
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH26_INT_CLEAR_0

Offset: 0x4191c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH26_CTRL_0

Offset: 0x41924
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH26_CONFIG_0

Offset: 0x41928
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH26_AHUB_FIFO_CTRL_0

Offset: 0x4192c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE4_CH26_TC_STATUS_0

Offset: 0x41930
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH26_LOWER_SOURCE_ADDR_0

Offset: 0x41934
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH26_LOWER_TARGET_ADDR_0

Offset: 0x4193c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH26_TC_0

Offset: 0x41944
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH26_LOWER_DESC_ADDR_0

Offset: 0x41948
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH26_TRANSFER_STATUS_0

Offset: 0x41954
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH26_TIMESTAMP_ADDR_0

Offset: 0x41958
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH26_TIMESTAMP_STATUS_0

Offset: 0x4195c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH26_APR_0

Offset: 0x41960
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH27_CMD_0

Offset: 0x41a00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH27_SOFT_RESET_0

Offset: 0x41a04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH27_STATUS_0

Offset: 0x41a0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH27_INT_STATUS_0

Offset: 0x41a10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH27_INT_SET_0

Offset: 0x41a18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH27_INT_CLEAR_0

Offset: 0x41a1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH27_CTRL_0

Offset: 0x41a24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH27_CONFIG_0

Offset: 0x41a28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH27_AHUB_FIFO_CTRL_0

Offset: 0x41a2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE4_CH27_TC_STATUS_0

Offset: 0x41a30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH27_LOWER_SOURCE_ADDR_0

Offset: 0x41a34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH27_LOWER_TARGET_ADDR_0

Offset: 0x41a3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH27_TC_0

Offset: 0x41a44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH27_LOWER_DESC_ADDR_0

Offset: 0x41a48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH27_TRANSFER_STATUS_0

Offset: 0x41a54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH27_TIMESTAMP_ADDR_0

Offset: 0x41a58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH27_TIMESTAMP_STATUS_0

Offset: 0x41a5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH27_APR_0

Offset: 0x41a60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH28_CMD_0

Offset: 0x41b00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH28_SOFT_RESET_0

Offset: 0x41b04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH28_STATUS_0

Offset: 0x41b0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH28_INT_STATUS_0

Offset: 0x41b10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH28_INT_SET_0

Offset: 0x41b18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH28_INT_CLEAR_0

Offset: 0x41b1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH28_CTRL_0

Offset: 0x41b24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH28_CONFIG_0

Offset: 0x41b28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH28_AHUB_FIFO_CTRL_0

Offset: 0x41b2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE4_CH28_TC_STATUS_0

Offset: 0x41b30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH28_LOWER_SOURCE_ADDR_0

Offset: 0x41b34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH28_LOWER_TARGET_ADDR_0

Offset: 0x41b3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH28_TC_0

Offset: 0x41b44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH28_LOWER_DESC_ADDR_0

Offset: 0x41b48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH28_TRANSFER_STATUS_0

Offset: 0x41b54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH28_TIMESTAMP_ADDR_0

Offset: 0x41b58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH28_TIMESTAMP_STATUS_0

Offset: 0x41b5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH28_APR_0

Offset: 0x41b60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH29_CMD_0

Offset: 0x41c00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH29_SOFT_RESET_0

Offset: 0x41c04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH29_STATUS_0

Offset: 0x41c0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH29_INT_STATUS_0

Offset: 0x41c10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH29_INT_SET_0

Offset: 0x41c18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH29_INT_CLEAR_0

Offset: 0x41c1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH29_CTRL_0

Offset: 0x41c24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH29_CONFIG_0

Offset: 0x41c28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH29_AHUB_FIFO_CTRL_0

Offset: 0x41c2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE4_CH29_TC_STATUS_0

Offset: 0x41c30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH29_LOWER_SOURCE_ADDR_0

Offset: 0x41c34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH29_LOWER_TARGET_ADDR_0

Offset: 0x41c3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH29_TC_0

Offset: 0x41c44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH29_LOWER_DESC_ADDR_0

Offset: 0x41c48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH29_TRANSFER_STATUS_0

Offset: 0x41c54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH29_TIMESTAMP_ADDR_0

Offset: 0x41c58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH29_TIMESTAMP_STATUS_0

Offset: 0x41c5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH29_APR_0

Offset: 0x41c60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH30_CMD_0

Offset: 0x41d00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH30_SOFT_RESET_0

Offset: 0x41d04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH30_STATUS_0

Offset: 0x41d0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH30_INT_STATUS_0

Offset: 0x41d10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH30_INT_SET_0

Offset: 0x41d18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH30_INT_CLEAR_0

Offset: 0x41d1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH30_CTRL_0

Offset: 0x41d24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH30_CONFIG_0

Offset: 0x41d28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH30_AHUB_FIFO_CTRL_0

Offset: 0x41d2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE4_CH30_TC_STATUS_0

Offset: 0x41d30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH30_LOWER_SOURCE_ADDR_0

Offset: 0x41d34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH30_LOWER_TARGET_ADDR_0

Offset: 0x41d3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH30_TC_0

Offset: 0x41d44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH30_LOWER_DESC_ADDR_0

Offset: 0x41d48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH30_TRANSFER_STATUS_0

Offset: 0x41d54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH30_TIMESTAMP_ADDR_0

Offset: 0x41d58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH30_TIMESTAMP_STATUS_0

Offset: 0x41d5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH30_APR_0

Offset: 0x41d60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH31_CMD_0

Offset: 0x41e00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH31_SOFT_RESET_0

Offset: 0x41e04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH31_STATUS_0

Offset: 0x41e0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH31_INT_STATUS_0

Offset: 0x41e10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH31_INT_SET_0

Offset: 0x41e18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH31_INT_CLEAR_0

Offset: 0x41e1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH31_CTRL_0

Offset: 0x41e24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH31_CONFIG_0

Offset: 0x41e28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH31_AHUB_FIFO_CTRL_0

Offset: 0x41e2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE4_CH31_TC_STATUS_0

Offset: 0x41e30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH31_LOWER_SOURCE_ADDR_0

Offset: 0x41e34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH31_LOWER_TARGET_ADDR_0

Offset: 0x41e3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH31_TC_0

Offset: 0x41e44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH31_LOWER_DESC_ADDR_0

Offset: 0x41e48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH31_TRANSFER_STATUS_0

Offset: 0x41e54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH31_TIMESTAMP_ADDR_0

Offset: 0x41e58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH31_TIMESTAMP_STATUS_0

Offset: 0x41e5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH31_APR_0

Offset: 0x41e60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH32_CMD_0

Offset: 0x41f00
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH32_SOFT_RESET_0

Offset: 0x41f04
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH32_STATUS_0

Offset: 0x41f0c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,x000,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
22:20	0x0	**CURRENT_SOURCE_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
18:16	0x0	**CURRENT_TARGET_MEMORY_BUFFER**: **0** = BUFFER_1 **1** = BUFFER_2 **2** = BUFFER_3 **3** = BUFFER_4 **4** = BUFFER_5 **5** = BUFFER_6 **6** = BUFFER_7 **7** = BUFFER_8
2	0x0	**OUTSTANDING_TRANSFERS**: **0** = FALSE **1** = TRUE
1	0x0	**TRANSFER_PAUSED**: **0** = FALSE **1** = TRUE
0	0x0	**TRANSFER_ENABLED**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH32_INT_STATUS_0

Offset: 0x41f10
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH32_INT_SET_0

Offset: 0x41f18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH32_INT_CLEAR_0

Offset: 0x41f1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TRANSFER_DONE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH32_CTRL_0

Offset: 0x41f24
Read/Write: RW
Parity Protection: N
Reset: 0x00004102 (0b0000,0000,0000,0000,0100,x001,xx00,0010)

Bit	Reset	Description
31:27	0x0	**TX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_TXFIFO_DMA_REQUEST **22** = I2C3_TXFIFO_DMA_REQUEST **23** = I2C8_TXFIFO_DMA_REQUEST
26:22	0x0	**RX_REQUEST_SELECT**: **0** = MEMORY **1** = AHUB_CH1_REQUEST **2** = AHUB_CH2_REQUEST **3** = AHUB_CH3_REQUEST **4** = AHUB_CH4_REQUEST **5** = AHUB_CH5_REQUEST **6** = AHUB_CH6_REQUEST **7** = AHUB_CH7_REQUEST **8** = AHUB_CH8_REQUEST **9** = AHUB_CH9_REQUEST **10** = AHUB_CH10_REQUEST **11** = AHUB_CH11_REQUEST **12** = AHUB_CH12_REQUEST **13** = AHUB_CH13_REQUEST **14** = AHUB_CH14_REQUEST **15** = AHUB_CH15_REQUEST **16** = AHUB_CH16_REQUEST **17** = AHUB_CH17_REQUEST **18** = AHUB_CH18_REQUEST **19** = AHUB_CH19_REQUEST **20** = AHUB_CH20_REQUEST **21** = I2C1_RXFIFO_DMA_REQUEST **22** = I2C3_RXFIFO_DMA_REQUEST **23** = I2C8_RXFIFO_DMA_REQUEST **24** = VI_RXFIFO_DMA_REQUEST

Bit	Reset	Description
21:16	0x0	**TRIGGER_SELECT**: **1** = CH1_TRANSFER_DONE **2** = CH2_TRANSFER_DONE **3** = CH3_TRANSFER_DONE **4** = CH4_TRANSFER_DONE **5** = CH5_TRANSFER_DONE **6** = CH6_TRANSFER_DONE **7** = CH7_TRANSFER_DONE **8** = CH8_TRANSFER_DONE **9** = CH9_TRANSFER_DONE **10** = CH10_TRANSFER_DONE **11** = CH11_TRANSFER_DONE **12** = CH12_TRANSFER_DONE **13** = CH13_TRANSFER_DONE **14** = CH14_TRANSFER_DONE **15** = CH15_TRANSFER_DONE **16** = CH16_TRANSFER_DONE **17** = CH17_TRANSFER_DONE **18** = CH18_TRANSFER_DONE **19** = CH19_TRANSFER_DONE **20** = CH20_TRANSFER_DONE **21** = CH21_TRANSFER_DONE **22** = CH22_TRANSFER_DONE **23** = CH23_TRANSFER_DONE **24** = CH24_TRANSFER_DONE **25** = CH25_TRANSFER_DONE **26** = CH26_TRANSFER_DONE **27** = CH27_TRANSFER_DONE **28** = CH28_TRANSFER_DONE **29** = CH29_TRANSFER_DONE **30** = CH30_TRANSFER_DONE **31** = CH31_TRANSFER_DONE **32** = CH32_TRANSFER_DONE
15:12	0x4	**TRANSFER_DIRECTION**: **1** = MEMORY_TO_MEMORY **2** = AHUB_TO_MEMORY **4** = MEMORY_TO_AHUB **8** = AHUB_TO_AHUB
10:8	0x1	**TRANSFER_MODE**: **1** = ONCE **2** = CONTINUOUS **4** = LINKED_LIST
5:4	0x0	**TIMESTAMP_CAPTURE_ENABLE**: **1** = MEMORY_WRITE **2** = REGISTER_WRITE
3	0x0	**FLOWCTRL_TYPE**: **0** = CREDIT_BASED **1** = THRESHOLD_BASED
2	0x0	**TRIGGER_ENABLE**: **0** = FALSE **1** = TRUE
1	0x1	**FLOWCTRL_ENABLE**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x0	**TRANSFER_PAUSE**: **0** = FALSE **1** = TRUE

ADMA_PAGE4_CH32_CONFIG_0

Offset: 0x41f28
Read/Write: RW
Parity Protection: N
Reset: 0x00f00081 (0bx000,x000,1111,0000,0000,xxxx,1000,0001)

Bit	Reset	Description
30:28	0x0	**SOURCE_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
26:24	0x0	**TARGET_MEMORY_BUFFERS**: **0** = BUFFER_1 **1** = BUFFERS_2 **2** = BUFFERS_3 **3** = BUFFERS_4 **4** = BUFFERS_5 **5** = BUFFERS_6 **6** = BUFFERS_7 **7** = BUFFERS_8
23:20	0xf	**BURST_SIZE**: **0** = WORD_1 **1** = WORDS_2 **2** = WORDS_3 **3** = WORDS_4 **4** = WORDS_5 **5** = WORDS_6 **6** = WORDS_7 **7** = WORDS_8 **8** = WORDS_9 **9** = WORDS_10 **10** = WORDS_11 **11** = WORDS_12 **12** = WORDS_13 **13** = WORDS_14 **14** = WORDS_15 **15** = WORDS_16

Bit	Reset	Description
19:16	0x0	**SOURCE_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
15:12	0x0	**TARGET_ADDR_WRAP**: **0** = NO_WRAP **1** = WRAP_ON_1_WORD **2** = WRAP_ON_2_WORDS **3** = WRAP_ON_4_WORDS **4** = WRAP_ON_8_WORDS **5** = WRAP_ON_16_WORDS **6** = WRAP_ON_32_WORDS **7** = WRAP_ON_64_WORDS **8** = WRAP_ON_128_WORDS **9** = WRAP_ON_256_WORDS **10** = WRAP_ON_512_WORDS **11** = WRAP_ON_1K_WORDS
7:4	0x8	**OUTSTANDING_REQUESTS**
3:0	0x1	**WEIGHT_FOR_WRR**

ADMA_PAGE4_CH32_AHUB_FIFO_CTRL_0

Offset: 0x41f2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0x00,0000,xx00,0000,xx00,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**FETCHING_POLICY**: **0** = BURST_BASED **1** = THRESHOLD_BASED
29:24	0x0	**OVERFLOW_THRESHOLD**
21:16	0x0	**STARVATION_THRESHOLD**
13:8	0x0	**TX_FIFO_SIZE**
5:0	0x0	**RX_FIFO_SIZE**

ADMA_PAGE4_CH32_TC_STATUS_0

Offset: 0x41f30
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**CURRENT_TRANSFER_COUNT**

ADMA_PAGE4_CH32_LOWER_SOURCE_ADDR_0

Offset: 0x41f34
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH32_LOWER_TARGET_ADDR_0

Offset: 0x41f3c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH32_TC_0

Offset: 0x41f44
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:2	0x0	**TRANSFER_COUNT**

ADMA_PAGE4_CH32_LOWER_DESC_ADDR_0

Offset: 0x41f48
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**NEXT_DESCRIPTOR_ADDR**

ADMA_PAGE4_CH32_TRANSFER_STATUS_0

Offset: 0x41f54
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_DONE_COUNT**

ADMA_PAGE4_CH32_TIMESTAMP_ADDR_0

Offset: 0x41f58
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**BASE_ADDR**

ADMA_PAGE4_CH32_TIMESTAMP_STATUS_0

Offset: 0x41f5c
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EOT_TIMESTAMP**

ADMA_PAGE4_CH32_APR_0

Offset: 0x41f60
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ENABLE**: **0** = FALSE **1** = TRUE

7.7.4.19 Audio Memory Control (AMC) Registers

AMC_CONFIG_0

Offset: 0x0
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000,00x0)

Bit	Reset	Description
6	FALSE	**APERTURE_NS_DIS**: **0** = FALSE **1** = TRUE
5	FALSE	**APR_NS_DIS**: **0** = FALSE **1** = TRUE
4	FALSE	**CO_NS_DIS**: **0** = FALSE **1** = TRUE
3	FALSE	**EVP_LOCK**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
2	FALSE	**ERR_RESPONSE_DISABLE**: **0** = FALSE **1** = TRUE
0	FALSE	**ARAM_ADDR_ALIAS_ENABLE**: **0** = FALSE **1** = TRUE

AMC_INT_STATUS_0

Offset: 0x4
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	FALSE	**INVALID_APR_ACCESS**: **0** = FALSE **1** = TRUE
4	FALSE	**INVALID_CO_ACCESS**: **0** = FALSE **1** = TRUE
3	FALSE	**INVALID_CONFIG_ADDR**: **0** = FALSE **1** = TRUE
2	FALSE	**INVALID_BURST_TYPE**: **0** = FALSE **1** = TRUE
1	FALSE	**INVALID_REQUEST_TYPE**: **0** = FALSE **1** = TRUE
0	FALSE	**INVALID_AMEM_ACCESS**: **0** = FALSE **1** = TRUE

AMC_INT_MASK_0

Offset: 0x8
Read/Write: RW
Parity Protection: N
Reset: 0x0000003f (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx11,1111)

Bit	Reset	Description
5	MASK	**INVALID_APR_ACCESS**: **0** = UNMASK **1** = MASK
4	MASK	**INVALID_CO_ACCESS**: **0** = UNMASK **1** = MASK
3	MASK	**INVALID_CONFIG_ADDR**: **0** = UNMASK **1** = MASK
2	MASK	**INVALID_BURST_TYPE**: **0** = UNMASK **1** = MASK
1	MASK	**INVALID_REQUEST_TYPE**: **0** = UNMASK **1** = MASK
0	MASK	**INVALID_AMEM_ACCESS**: **0** = UNMASK **1** = MASK

AMC_INT_SET_0

Offset: 0xc
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	FALSE	**INVALID_APR_ACCESS**: **0** = FALSE **1** = TRUE
4	FALSE	**INVALID_CO_ACCESS**: **0** = FALSE **1** = TRUE
3	FALSE	**INVALID_CONFIG_ADDR**: **0** = FALSE **1** = TRUE
2	FALSE	**INVALID_BURST_TYPE**: **0** = FALSE **1** = TRUE
1	FALSE	**INVALID_REQUEST_TYPE**: **0** = FALSE **1** = TRUE
0	FALSE	**INVALID_AMEM_ACCESS**: **0** = FALSE **1** = TRUE

AMC_INT_CLEAR_0

Offset: 0x10
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	FALSE	**INVALID_APR_ACCESS**: **0** = FALSE **1** = TRUE
4	FALSE	**INVALID_CO_ACCESS**: **0** = FALSE **1** = TRUE
3	FALSE	**INVALID_CONFIG_ADDR**: **0** = FALSE **1** = TRUE
2	FALSE	**INVALID_BURST_TYPE**: **0** = FALSE **1** = TRUE
1	FALSE	**INVALID_REQUEST_TYPE**: **0** = FALSE **1** = TRUE
0	FALSE	**INVALID_AMEM_ACCESS**: **0** = FALSE **1** = TRUE

AMC_ERROR_ADDR_0

Offset: 0x14
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ERROR_ADDR**

AMC_APR_CONFIG_0

Offset: 0x18
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**ENABLE**: **0** = FALSE **1** = TRUE

AMC_APR_START_ADDR_0

Offset: 0x1c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**APR_START_ADDR**

AMC_APR_SIZE_0

Offset: 0x20
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
27:0	0x0	**APR_SIZE**

AMC_CO_CONFIG_0

Offset: 0x24
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**ENABLE**: **0** = FALSE **1** = TRUE

AMC_CO_START_ADDR_0

Offset: 0x28
Read/Write: RW

Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CO_START_ADDR**

AMC_CO_SIZE_0

Offset: 0x2c
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16:0	0x0	**CO_SIZE**

AMC_APERTURE_BASE_0

Offset: 0x30
Read/Write: RW
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxx0,0000,0000,0xxx,xxxx,xxxx)

Bit	Reset	Description
20:11	0x0	**APERTURE_BASE**

AMC_EVP_RESET_VEC_0

Offset: 0x700
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESET_VECTOR**

AMC_EVP_UNDEF_VEC_0

Offset: 0x704
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**UNDEF_VECTOR**

AMC_EVP_SWI_VEC_0

Offset: 0x708
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**SWI_VECTOR**

AMC_EVP_PREFETCH_ABORT_VEC_0

Offset: 0x70c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**PREFETCH_ABORT_VECTOR**

AMC_EVP_DATA_ABORT_VEC_0

Offset: 0x710
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA_ABORT_VECTOR**

AMC_EVP_RSVD_VEC_0

Offset: 0x714
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RSVD_VECTOR**

AMC_EVP_IRQ_VEC_0

Offset: 0x718
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**IRQ_VECTOR**

AMC_EVP_FIQ_VEC_0

Offset: 0x71c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**FIQ_VECTOR**

AMC_EVP_RESET_ADDR_0

Offset: 0x720
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESET_ADDR**

AMC_EVP_UNDEF_ADDR_0

Offset: 0x724
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**UNDEF_ADDR**

AMC_EVP_SWI_ADDR_0

Offset: 0x728
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**SWI_ADDR**

AMC_EVP_PREFETCH_ABORT_ADDR_0

Offset: 0x72c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**PREFETCH_ABORT_ADDR**

AMC_EVP_DATA_ABORT_ADDR_0

Offset: 0x730
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA_ABORT_ADDR**

AMC_EVP_RSVD_ADDR_0

Offset: 0x734
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RSVD_ADDR**

AMC_EVP_IRQ_ADDR_0

Offset: 0x738
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**IRQ_ADDR**

AMC_EVP_FIQ_ADDR_0

Offset: 0x73c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**FIQ_ADDR**

AMC_APERTURE_DATA_0

This is an array of 512 identical register entries; the register fields below apply to each entry.
Offset: 0x800,..,0xffc
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**DATA**

7.7.4.20 Audio Digital Signal Processor Peripheral (ADSP_PERIPH) Control Registers

ADSP_PERIPH_CPUIF_ICCICR_0

Offset: 0x100
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICCICR**: CPU Interface Control Register

ADSP_PERIPH_CPUIF_ICCPMR_0

Offset: 0x104
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICCPMR**: Interrupt Priority Mask Register

ADSP_PERIPH_CPUIF_ICCBPR_0

Offset: 0x108
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICCBPR**: Binary Point Register

ADSP_PERIPH_CPUIF_ICCIAR_0

Offset: 0x10c
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICCIAR**: Interrupt Acknowledge Register

ADSP_PERIPH_CPUIF_ICCEOIR_0

Offset: 0x110
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICCEOIR**: End of Interrupt Register

ADSP_PERIPH_CPUIF_ICCRPR_0

Offset: 0x114
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICCRPR**: Running Priority Register

ADSP_PERIPH_CPUIF_ICCHPPIR_0

Offset: 0x118
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICCHPPIR**: Highest Priority Pending Interrupt Register

ADSP_PERIPH_CPUIF_ICCABPR_0

Offset: 0x11c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICC_ABPR**: Aliased Binary Point Register

ADSP_PERIPH_CPUIF_ICCIDR_0

Offset: 0x1fc
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICCIDR**: CPU Interface Implementer Identification Register

ADSP_PERIPH_DISTRIBUTOR_ICDDCR_0

Offset: 0x1000
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDDCR**: Distributor Control Register

ADSP_PERIPH_DISTRIBUTOR_ICDICTR_0

Offset: 0x1004
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICTR**: Interrupt Controller Type Register

ADSP_PERIPH_DISTRIBUTOR_ICDIIDR_0

Offset: 0x1008
Read/Write: RO

Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIIDR**: Distributor Implementer Identification Register

ADSP_PERIPH_DISTRIBUTOR_ICDISR0_0

Offset: 0x1080
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDISR0**: Interrupt Group Registers

ADSP_PERIPH_DISTRIBUTOR_ICDISR1_0

Offset: 0x1084
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDISR1**: Interrupt Group Registers

ADSP_PERIPH_DISTRIBUTOR_ICDISR2_0

Offset: 0x1088
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDISR2**: Interrupt Group Registers

ADSP_PERIPH_DISTRIBUTOR_ICDISR3_0

Offset: 0x108c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDISR3**: Interrupt Group Registers

ADSP_PERIPH_DISTRIBUTOR_ICDISR4_0

Offset: 0x1090
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDISR4**: Interrupt Group Registers

ADSP_PERIPH_DISTRIBUTOR_ICDISR5_0

Offset: 0x1094
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDISR5**: Interrupt Group Registers

ADSP_PERIPH_DISTRIBUTOR_ICDISR6_0

Offset: 0x1098
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDISR6**: Interrupt Group Registers

ADSP_PERIPH_DISTRIBUTOR_ICDISR7_0

Offset: 0x109c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDISR7**: Interrupt Group Registers

ADSP_PERIPH_DISTRIBUTOR_ICDISER0_0

Offset: 0x1100
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDISER0**: Interrupt Set-Enable Registers

ADSP_PERIPH_DISTRIBUTOR_ICDISER1_0

Offset: 0x1104
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDISER1**: Interrupt Set-Enable Registers

ADSP_PERIPH_DISTRIBUTOR_ICDISER2_0

```
ICDISER2 offset(decimal)=4360 size=32
```

Offset: 0x1108
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDISER2**: Interrupt Set-Enable Registers

ADSP_PERIPH_DISTRIBUTOR_ICDISER3_0

Offset: 0x110c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDISER3**: Interrupt Set-Enable Registers

ADSP_PERIPH_DISTRIBUTOR_ICDISER4_0

Offset: 0x1110
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDISER4**: Interrupt Set-Enable Registers

ADSP_PERIPH_DISTRIBUTOR_ICDISER5_0

Offset: 0x1114
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDISER5**: Interrupt Set-Enable Registers

ADSP_PERIPH_DISTRIBUTOR_ICDISER6_0

Offset: 0x1118
Read/Write: RW

Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDISER6**: Interrupt Set-Enable Registers

ADSP_PERIPH_DISTRIBUTOR_ICDISER7_0

Offset: 0x111c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDISER7**: Interrupt Set-Enable Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICER0_0

Offset: 0x1180
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICER0**: Interrupt Clear-Enable Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICER1_0

Offset: 0x1184
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICER1**: Interrupt Clear-Enable Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICER2_0

Offset: 0x1188
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICER2**: Interrupt Clear-Enable Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICER3_0

Offset: 0x118c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICER3**: Interrupt Clear-Enable Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICER4_0

Offset: 0x1190
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICER4**: Interrupt Clear-Enable Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICER5_0

Offset: 0x1194
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICER5**: Interrupt Clear-Enable Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICER6_0

Offset: 0x1198
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICER6**: Interrupt Clear-Enable Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICER7_0

Offset: 0x119c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICER7**: Interrupt Clear-Enable Registers

ADSP_PERIPH_DISTRIBUTOR_ICDISPR0_0

Offset: 0x1200
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDISPR0**: Interrupt Set-Pending Registers

ADSP_PERIPH_DISTRIBUTOR_ICDISPR1_0

Offset: 0x1204
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDISPR1**: Interrupt Set-Pending Registers

ADSP_PERIPH_DISTRIBUTOR_ICDISPR2_0

Offset: 0x1208
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDISPR2**: Interrupt Set-Pending Registers

ADSP_PERIPH_DISTRIBUTOR_ICDISPR3_0

Offset: 0x120c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDISPR3**: Interrupt Set-Pending Registers

ADSP_PERIPH_DISTRIBUTOR_ICDISPR4_0

Offset: 0x1210
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDISPR4**: Interrupt Set-Pending Registers

ADSP_PERIPH_DISTRIBUTOR_ICDISPR5_0

Offset: 0x1214
Read/Write: RW

Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDISPR5**: Interrupt Set-Pending Registers

ADSP_PERIPH_DISTRIBUTOR_ICDISPR6_0

Offset: 0x1218
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDISPR6**: Interrupt Set-Pending Registers

ADSP_PERIPH_DISTRIBUTOR_ICDISPR7_0

Offset: 0x121c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDISPR7**: Interrupt Set-Pending Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICPR0_0

Offset: 0x1280
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICPR0**: Interrupt Clear-Pending Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICPR1_0

Offset: 0x1284
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICPR1**: Interrupt Clear-Pending Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICPR2_0

Offset: 0x1288
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICPR2**: Interrupt Clear-Pending Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICPR3_0

Offset: 0x128c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICPR3**: Interrupt Clear-Pending Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICPR4_0

Offset: 0x1290
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICPR4**: Interrupt Clear-Pending Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICPR5_0

Offset: 0x1294
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICPR5**: Interrupt Clear-Pending Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICPR6_0

Offset: 0x1298
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICPR6**: Interrupt Clear-Pending Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICPR7_0

Offset: 0x129c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICPR7**: Interrupt Clear-Pending Registers

ADSP_PERIPH_DISTRIBUTOR_ICDABR0_0

Offset: 0x1300
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDABR0**: Interrupt Active Bit Registers

ADSP_PERIPH_DISTRIBUTOR_ICDABR1_0

Offset: 0x1304
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDABR1**: Interrupt Active Bit Registers

ADSP_PERIPH_DISTRIBUTOR_ICDABR2_0

Offset: 0x1308
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDABR2**: Interrupt Active Bit Registers

ADSP_PERIPH_DISTRIBUTOR_ICDABR3_0

Offset: 0x130c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDABR3**: Interrupt Active Bit Registers

ADSP_PERIPH_DISTRIBUTOR_ICDABR4_0

Offset: 0x1310
Read/Write: RW

Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDABR4**: Interrupt Active Bit Registers

ADSP_PERIPH_DISTRIBUTOR_ICDABR5_0

Offset: 0x1314
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDABR5**: Interrupt Active Bit Registers

ADSP_PERIPH_DISTRIBUTOR_ICDABR6_0

Offset: 0x1318
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDABR6**: Interrupt Active Bit Registers

ADSP_PERIPH_DISTRIBUTOR_ICDABR7_0

Offset: 0x131c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDABR7**: Interrupt Active Bit Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR0_0

Offset: 0x1400
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR0**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR1_0

Offset: 0x1404
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR1**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR2_0

Offset: 0x1408
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR2**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR3_0

Offset: 0x140c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR3**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR4_0

Offset: 0x1410
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR4**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR5_0

Offset: 0x1414
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR5**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR6_0

Offset: 0x1418
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR6**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR7_0

Offset: 0x141c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR7**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR8_0

Offset: 0x1420
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR8**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR9_0

Offset: 0x1424
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR9**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR10_0

Offset: 0x1428
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR10**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR11_0

Offset: 0x142c
Read/Write: RW

Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR11**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR12_0

Offset: 0x1430
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR12**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR13_0

Offset: 0x1434
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR13**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR14_0

Offset: 0x1438
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR14**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR15_0

Offset: 0x143c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR15**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR16_0

Offset: 0x1440
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR16**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR17_0

Offset: 0x1444
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR17**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR18_0

Offset: 0x1448
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR18**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR19_0

Offset: 0x144c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR19**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR20_0

Offset: 0x1450
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR20**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR21_0

Offset: 0x1454
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR21**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR22_0

Offset: 0x1458
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR22**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR23_0

Offset: 0x145c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR23**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR24_0

Offset: 0x1460
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR24**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR25_0

Offset: 0x1464
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR25**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR26_0

Offset: 0x1468
Read/Write: RW

Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR26**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR27_0

Offset: 0x146c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR27**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR28_0

Offset: 0x1470
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR28**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR29_0

Offset: 0x1474
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR29**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR30_0

Offset: 0x1478
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR30**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR31_0

Offset: 0x147c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR31**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR32_0

Offset: 0x1480
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR32**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR33_0

Offset: 0x1484
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR33**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR34_0

Offset: 0x1488
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR34**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR35_0

Offset: 0x148c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR35**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR36_0

Offset: 0x1490
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR36**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR37_0

Offset: 0x1494
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR37**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR38_0

Offset: 0x1498
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR38**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR39_0

Offset: 0x149c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR39**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR40_0

Offset: 0x14a0
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR40**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR41_0

Offset: 0x14a4
Read/Write: RW

Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR41**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR42_0

Offset: 0x14a8
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR42**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR43_0

Offset: 0x14ac
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR43**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR44_0

Offset: 0x14b0
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR44**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR45_0

Offset: 0x14b4
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR45**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR46_0

Offset: 0x14b8
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR46**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR47_0

Offset: 0x14bc
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR47**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR48_0

Offset: 0x14c0
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR48**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR49_0

Offset: 0x14c4
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR49**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR50_0

Offset: 0x14c8
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR50**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR51_0

Offset: 0x14cc
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR51**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR52_0

Offset: 0x14d0
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR52**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR53_0

Offset: 0x14d4
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR53**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR54_0

Offset: 0x14d8
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR54**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR55_0

Offset: 0x14dc
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR55**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR56_0

Offset: 0x14e0
Read/Write: RW

Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR56**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR57_0

Offset: 0x14e4
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR57**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR58_0

Offset: 0x14e8
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR58**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR59_0

Offset: 0x14ec
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR59**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR60_0

Offset: 0x14f0
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR60**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR61_0

Offset: 0x14f4
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR61**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR62_0

Offset: 0x14f8
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR62**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPR63_0

Offset: 0x14fc
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPR63**: Interrupt Priority Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR0_0

Offset: 0x1800
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR0**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR1_0

Offset: 0x1804
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR1**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR2_0

Offset: 0x1808
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR2**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR3_0

Offset: 0x180c
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR3**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR4_0

Offset: 0x1810
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR4**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR5_0

Offset: 0x1814
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR5**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR6_0

Offset: 0x1818
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR6**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR7_0

Offset: 0x181c
Read/Write: RO

Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR7**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR8_0

Offset: 0x1820
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR8**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR9_0

Offset: 0x1824
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR9**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR10_0

Offset: 0x1828
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR10**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR11_0

Offset: 0x182c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR11**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR12_0

Offset: 0x1830
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR12**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR13_0

Offset: 0x1834
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR13**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR14_0

Offset: 0x1838
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR14**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR15_0

Offset: 0x183c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR15**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR16_0

Offset: 0x1840
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR16**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR17_0

Offset: 0x1844
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR17**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR18_0

Offset: 0x1848
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR18**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR19_0

Offset: 0x184c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR19**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR20_0

Offset: 0x1850
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR20**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR21_0

Offset: 0x1854
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR21**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR22_0

Offset: 0x1858
Read/Write: RW

Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR22**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR23_0

Offset: 0x185c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR23**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR24_0

Offset: 0x1860
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR24**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR25_0

Offset: 0x1864
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR25**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR26_0

Offset: 0x1868
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR26**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR27_0

Offset: 0x186c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR27**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR28_0

Offset: 0x1870
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR28**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR29_0

Offset: 0x1874
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR29**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR30_0

Offset: 0x1878
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR30**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR31_0

Offset: 0x187c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR31**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR32_0

Offset: 0x1880
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR32**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR33_0

Offset: 0x1884
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR33**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR34_0

Offset: 0x1888
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR34**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR35_0

Offset: 0x188c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR35**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR36_0

Offset: 0x1890
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR36**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR37_0

Offset: 0x1894
Read/Write: RW

Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR37**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR38_0

Offset: 0x1898
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR38**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR39_0

Offset: 0x189c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR39**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR40_0

Offset: 0x18a0
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR40**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR41_0

Offset: 0x18a4
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR41**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR42_0

Offset: 0x18a8
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR42**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR43_0

Offset: 0x18ac
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR43**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR44_0

Offset: 0x18b0
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR44**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR45_0

Offset: 0x18b4
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR45**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR46_0

Offset: 0x18b8
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR46**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR47_0

Offset: 0x18bc
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR47**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR48_0

Offset: 0x18c0
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR48**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR49_0

Offset: 0x18c4
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR49**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR50_0

Offset: 0x18c8
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR50**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR51_0

Offset: 0x18cc
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR51**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR52_0

Offset: 0x18d0
Read/Write: RW

Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR52**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR53_0

Offset: 0x18d4
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR53**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR54_0

Offset: 0x18d8
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR54**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR55_0

Offset: 0x18dc
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR55**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR56_0

Offset: 0x18e0
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR56**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR57_0

Offset: 0x18e4
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR57**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR58_0

Offset: 0x18e8
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR58**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR59_0

Offset: 0x18ec
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR59**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR60_0

Offset: 0x18f0
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR60**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR61_0

Offset: 0x18f4
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR61**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR62_0

Offset: 0x18f8
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR62**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDIPTR63_0

Offset: 0x18fc
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDIPTR63**: Interrupt Processor Targets Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICFR0_0

Offset: 0x1c00
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICFR0**: Interrupt Configuration Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICFR1_0

Offset: 0x1c04
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICFR1**: Interrupt Configuration Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICFR2_0

Offset: 0x1c08
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICFR2**: Interrupt Configuration Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICFR3_0

Offset: 0x1c0c
Read/Write: RW

Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICFR3**: Interrupt Configuration Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICFR4_0

Offset: 0x1c10
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICFR4**: Interrupt Configuration Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICFR5_0

Offset: 0x1c14
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICFR5**: Interrupt Configuration Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICFR6_0

Offset: 0x1c18
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICFR6**: Interrupt Configuration Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICFR7_0

Offset: 0x1c1c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICFR7**: Interrupt Configuration Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICFR8_0

Offset: 0x1c20
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICFR8**: Interrupt Configuration Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICFR9_0

Offset: 0x1c24
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICFR9**: Interrupt Configuration Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICFR10_0

Offset: 0x1c28
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICFR10**: Interrupt Configuration Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICFR11_0

Offset: 0x1c2c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICFR11**: Interrupt Configuration Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICFR12_0

Offset: 0x1c30
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICFR12**: Interrupt Configuration Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICFR13_0

Offset: 0x1c34
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICFR13**: Interrupt Configuration Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICFR14_0

Offset: 0x1c38
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICFR14**: Interrupt Configuration Registers

ADSP_PERIPH_DISTRIBUTOR_ICDICFR15_0

Offset: 0x1c3c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDICFR15**: Interrupt Configuration Registers

ADSP_PERIPH_DISTRIBUTOR_ICPPISR_0

Offset: 0x1d00
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICPPISR**: Private Peripheral Interrupt Status Register

ADSP_PERIPH_DISTRIBUTOR_ICSPISR0_0

Offset: 0x1d04
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICSPISR0**: Shared Peripheral Interrupt Status Registers

ADSP_PERIPH_DISTRIBUTOR_ICSPISR1_0

Offset: 0x1d08
Read/Write: RO

Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICSPISR1**: Shared Peripheral Interrupt Status Registers

ADSP_PERIPH_DISTRIBUTOR_ICSPISR2_0

Offset: 0x1d0c
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICSPISR2**: Shared Peripheral Interrupt Status Registers

ADSP_PERIPH_DISTRIBUTOR_ICSPISR3_0

Offset: 0x1d10
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICSPISR3**: Shared Peripheral Interrupt Status Registers

ADSP_PERIPH_DISTRIBUTOR_ICSPISR4_0

Offset: 0x1d14
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICSPISR4**: Shared Peripheral Interrupt Status Registers

ADSP_PERIPH_DISTRIBUTOR_ICSPISR5_0

Offset: 0x1d18
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICSPISR5**: Shared Peripheral Interrupt Status Registers

ADSP_PERIPH_DISTRIBUTOR_ICSPISR6_0

Offset: 0x1d1c
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICSPISR6**: Shared Peripheral Interrupt Status Registers

ADSP_PERIPH_DISTRIBUTOR_ICDSGIR_0

Offset: 0x1f00
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICDSGIR**: Software Generated Interrupt Register

ADSP_PERIPH_DISTRIBUTOR_ICPIDR0_0

Offset: 0x1fd0
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICPIDR0**: Peripheral ID4 Register

ADSP_PERIPH_DISTRIBUTOR_ICPIDR1_0

Offset: 0x1fd4
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICPIDR1**: Peripheral ID5 Register

ADSP_PERIPH_DISTRIBUTOR_ICPIDR2_0

Offset: 0x1fd8
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICPIDR2**: Peripheral ID6 Register

ADSP_PERIPH_DISTRIBUTOR_ICPIDR3_0

Offset: 0x1fdc
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICPIDR3**: Peripheral ID7 Register

ADSP_PERIPH_DISTRIBUTOR_ICPIDR4_0

Offset: 0x1fe0
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICPIDR4**: Peripheral ID0 Register

ADSP_PERIPH_DISTRIBUTOR_ICPIDR5_0

Offset: 0x1fe4
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICPIDR5**: Peripheral ID1 Register

ADSP_PERIPH_DISTRIBUTOR_ICPIDR6_0

Offset: 0x1fe8
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICPIDR6**: Peripheral ID2 Register

ADSP_PERIPH_DISTRIBUTOR_ICPIDR7_0

Offset: 0x1fec
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICPIDR7**: Peripheral ID3 Register

ADSP_PERIPH_DISTRIBUTOR_ICCIDR0_0

Offset: 0x1ff0
Read/Write: RO

Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICCIDR0**: Component ID0 Register

ADSP_PERIPH_DISTRIBUTOR_ICCIDR1_0

Offset: 0x1ff4
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICCIDR1**: Component ID1 Register

ADSP_PERIPH_DISTRIBUTOR_ICCIDR2_0

Offset: 0x1ff8
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICCIDR2**: Component ID2 Register

ADSP_PERIPH_DISTRIBUTOR_ICCIDR3_0

Offset: 0x1ffc
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**ICCIDR3**: Component ID3 Register

7.7.4.21 Audio Miscellaneous (AMISC) Control Registers

AMISC_AMISC_CONFIG_0

Offset: 0x0
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DISABLE_ERROR_RESPONSE**: Do not send error on axi bus on invalid access

AMISC_ADSP_CONFIG_0

Offset: 0x4
Read/Write: R/W
Parity Protection: N
Reset: 0xe0800000 (0b111x,x000,1x00,00xx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:29	0x7	**MAXCLKLATENCY**:
26:23	0x1	**CLUSTERID**
21	0x0	**VINITHI**
20	0x0	**CFGSDISABLE**
19	0x0	**CP15SDISABLE**
18	0x0	**EVENTI**

AMISC_ADSP_PERIPHBASE_0

Offset: 0x8
Read/Write: R/W
Parity Protection: N
Reset: 0x00002000 (0b0000,0000,0000,0000,001x,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:13	0x1	**PERIPHBASE**

AMISC_ADSP_L2_CONFIG_0

Offset: 0xc
Read/Write: R/W
Parity Protection: N
Reset: 0x02000000 (0bx000,001x,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
30:25	0x1	**CACHEID**

AMISC_ADSP_L2_REGFILEBASE_0

Offset: 0x10
Read/Write: R/W
Parity Protection: N
Reset: 0x00004000 (0b0000,0000,0000,0000,0100,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:12	0x4	**REGFILEBASE**

AMISC_ADSP_STATUS_0

Offset: 0x14
Read/Write: RO
Parity Protection: N
Reset: 0xX0000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	X	**L2_IDLE**
30	X	**L2_CLKSTOPPED**
29	X	**DBGNOPWRDWN**

AMISC_ADSP_AGIC_CONFIG_0

Offset: 0x18
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0b0xx0,0000,0000,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**CFGSDISABLE**
28:24	0x0	**NS_ENABLE**

Bit	Reset	Description
23:20	0x0	**GICD_MASK**

AMISC_CONFIG_LOCK_0

Offset: 0x1c
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**ADSP**

AMISC_TSC_LOW_0

Offset: 0x4c
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**TIMESTAMP**

AMISC_TSC_HIGH_0

Offset: 0x50
Read/Write: RO
Parity Protection: N
Reset: 0x00XXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
23:0	X	**TIMESTAMP**

AMISC_IDLE_0

Offset: 0x54
Read/Write: R/W
Parity Protection: N
Reset: 0x80000000 (0b1xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x1	**IDLE_EN**

AMISC_ACTMON_0

Offset: 0x58
Read/Write: R/W
Parity Protection: N
Reset: 0x7f800000 (0b0111,1111,1xxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**CNT_ENABLE**: Enable the adsp side counter
30:23	0xff	**CNT_TARGET**: Count target minus 1 before signal to actmon. Actmon will be signalled when the counter = CNT_TARGET and another ~standbywfi cycle is detected (i.e. CNT_TARGET+1 events).

AMISC_MC_STATS_READ_0

This is an array of 2 identical register entries; the register fields below apply to each entry.
Full register list is: AMISC_MC_STATS_READ_<*i*>, where <*i*> = 0, 1.
Note: the register with (<*i*> = 0) is for ACAST,
the register with (<*i*> = 1) is for ADAST.

Offset: 0x60,0x64
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**COUNT**

AMISC_MC_STATS_WRITE_0

This is an array of 2 identical register entries; the register fields below apply to each entry.
Full register list is: AMISC_MC_STATS_WRITE_<*i*>, where <*i*> = 0, 1.
Note: the register with (*i* = 0) is for ACAST,
the register with (*i* = 1) is for ADAST.

Offset: 0x68,0x6c
Read/Write: RO

Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**COUNT**

AMISC_MC_STATS_CLEAR_0

This is an array of 2 identical register entries; the register fields below apply to each entry.
Full register list is: AMISC_MC_STATS_CLEAR_<*i*>, where <*i*> = 0, 1.

Offset: 0x70, 0x74
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**CLEAR_WRITE_COUNT**
0	0x0	**CLEAR_READ_COUNT**

AMISC_KEYSLOT_NS_0

Offset: 0x7c
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	0x0	**DISABLE**

AMISC_KEYSLOT_KEY<*i*>_<*j*>,

where <*i*> = 0, 1, 2, 3 and

<*j*> = 0, 1, 2, 3.

AMISC_KEYSLOT_KEY0_0
Offset: 0x80

AMISC_KEYSLOT_KEY0_1
Offset: 0x84

AMISC_KEYSLOT_KEY0_2
Offset: 0x88

AMISC_KEYSLOT_KEY0_3
Offset: 0x8c

AMISC_KEYSLOT_KEY1_0
Offset: 0x90

AMISC_KEYSLOT_KEY1_1
Offset: 0x94

AMISC_KEYSLOT_KEY1_2
Offset: 0x98

AMISC_KEYSLOT_KEY1_3
Offset: 0x9c

AMISC_KEYSLOT_KEY2_0
Offset: 0xa0

AMISC_KEYSLOT_KEY2_1
Offset: 0xa4

AMISC_KEYSLOT_KEY2_2
Offset: 0xa8

AMISC_KEYSLOT_KEY2_3
Offset: 0xac

AMISC_KEYSLOT_KEY3_0
Offset: 0xb0

AMISC_KEYSLOT_KEY3_1
Offset: 0xb4

AMISC_KEYSLOT_KEY3_2
Offset: 0xb8

AMISC_KEYSLOT_KEY3_3
Offset: 0xbc

Read/Write: R/W
Parity Protection: N
Secure: Trust Zone Protected
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**KEY**

The following 4 registers are for EAVB-APE clock synchronization.

AMISC_APE_TSC_CTRL_0_0

Offset: 0xc0
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**N_MODULO**: Modulo value after which fractional value will be added to integer value.
15:0	0x0	**N_FRACT**: Fractional value of the Timestamp counter

AMISC_APE_TSC_CTRL_1_0

Offset: 0xc4
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**N_INT**: Integer value to increment for every clock

AMISC_APE_TSC_CTRL_2_0

Offset: 0xc8
Read/Write: R/W
Parity Protection: N
Reset: 0x3b9aca00 (0b0011,1011,1001,1010,1100,1010,0000,0000)

Bit	Reset	Description
31:0	0x3b9aca00	**N_SEC**: Second counter will be incremented for every N_SEC nanoseconds.

AMISC_APE_TSC_CTRL_3_0

Offset: 0xcc
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
31	0x0	**ENABLE**: Enable signals for clock synchronization.Loads all initialization values on 0 to 1.
2	0x0	**COPY**: Copy value of EAVB timestamps to APE timestamps. Clears FRAC and to 0.,auto clear bit
1	0x0	**RST**: When written 1 to this field will reset the APE timestamping counter logic,auto clear bit
0	0x0	**TRIGGER**: When written 1 to this field will trigger the snapshot capture for both EAVB/APE timestamps,auto clear bit

AMISC_APE_RT_TSC_NS_0

Offset: 0xd0
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**NS**: Nano second time stamp value.

AMISC_APE_RT_TSC_SEC_0

Offset: 0xd4
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SEC**: Second time stamp value

AMISC_APE_SNAP_TSC_NS_0

Offset: 0xd8
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**NS**: Nano second time stamp value.

AMISC_APE_SNAP_TSC_SEC_0

Offset: 0xdc
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SEC**: Second time stamp value

AMISC_EAVB_SNAP_TSC_NS_0

Offset: 0xe0
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**NS**: Nano second time stamp value.

AMISC_EAVB_SNAP_TSC_SEC_0

Offset: 0xe4
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SEC**: Second time stamp value

AMISC_SPARE_0

Offset: 0xe8
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SPARE**

AMISC_ACONNECT_CLOCK_GATING_DISABLE_0

Offset: 0xec
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0b0000,0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**APE_GATING_DISABLE**
30	0x0	**ADMA_GATING_DISABLE**
29	0x0	**ADSP_GATING_DISABLE**
28	0x0	**AHUB_GATING_DISABLE**
27	0x0	**APBP_GATING_DISABLE**

AMISC_APE_SLCG_OVERRIDE_0

Offset: 0xf0
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**ape_clk_ovr_on**

AMISC_ACTMON_INPUT_CONFIG_0

This is an array of 3 identical register entries; the register fields below apply to each entry. Full register list is: AMISC_ACTMON_INPUT_CONFIG_<*i*>, where <*i*> = 0, 1, 2.

Offset: 0x100,..,0x108
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	0x0	**INPUT_SEL**: **0** = ADSP_ACTIVE **1** = ADSP_MEM_STALL **2** = AST_ACTIVE **5** = OUTSTANDING_REQ_THRESHOLD

AMISC_ACTMON_MEM_ACTIVITY_0

This is an array of 3 identical register entries; the register fields below apply to each entry. Full register list is: AMISC_ACTMON_MEM_ACTIVITY_<*i*>, where <*i*> = 0, 1, 2.

Offset: 0x10c,..,0x114
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,xxxx,0000)

Bit	Reset	Description
8	0x0	**MEM_ACT_MODE**: **1** = REQUEST_COUNT **0** = TRANSFER_COUNT
3	0x0	**ADAST_WR_ACT_EN**: **0** = DISABLE **1** = ENABLE
2	0x0	**ADAST_RD_ACT_EN**: **0** = DISABLE **1** = ENABLE
1	0x0	**ACAST_WR_ACT_EN**: **0** = DISABLE **1** = ENABLE
0	0x0	**ACAST_RD_ACT_EN**: **0** = DISABLE **1** = ENABLE

AMISC_ACTMON_ADSP_PMU_0

This is an array of 3 identical register entries; the register fields below apply to each entry. Full register list is: AMISC_ACTMON_ADSP_PMU_<*i*>, where <*i*> = 0, 1, 2.

Offset: 0x118,..,0x120
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**ADSP_STALL_SEL**: **0** = DISABLED **1** = INSTR_STALL **2** = DATA_STALL **3** = INSTR_DATA_STALL

AMISC_ACTMON_REQ_THRSHOLD_0

This is an array of 3 identical register entries; the register fields below apply to each entry. Full register list is: AMISC_ACTMON_REQ_THRSHOLD_<*i*>, where <*i*> = 0, 1, 2.

Offset: 0x124,..,0x12c
Read/Write: R/W
Parity Protection: N
Reset: 0x0000000f (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,x000,1111)

Bit	Reset	Description
11	0x0	**REQ_THRSHLD_ADAST_RD_EN**: **0** = DISABLE **1** = ENABLE
10	0x0	**REQ_THRSHLD_ADAST_WR_EN**: **0** = DISABLE **1** = ENABLE
9	0x0	**REQ_THRSHLD_ACAST_WR_EN**: **0** = DISABLE **1** = ENABLE
8	0x0	**REQ_THRSHLD_ACAST_RD_EN**: **0** = DISABLE **1** = ENABLE
6:0	0xf	**REQ_THRSHLD_LEVEL**

AMISC_ACTMON_MEM_ACTIVITY_PULSE_N_0

This is an array of 3 identical register entries; the register fields below apply to each entry. Full register list is: AMISC_ACTMON_MEM_ACTIVITY_PULSE_N_<*i*>, where <*i*> = 0, 1, 2.

Offset: 0x130,..,0x138
Read/Write: R/W

Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,x000,0000)

Bit	Reset	Description
8	0x0	**ENABLE**: **0** = DISABLE **1** = ENABLE
6:0	0x0	**PULSE_N**

AMISC_ACTMON_COMMON_ADSP_PULSE_N_0

Offset: 0x13c
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
8	0x0	**ENABLE**: **0** = DISABLE **1** = ENABLE
7:0	0x0	**PULSE_N**

AMISC_ACTMON_COMMON_REQ_THRSHOLD_PULSE_N_0

Offset: 0x140
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,x000,0000)

Bit	Reset	Description
8	0x0	**ENABLE**: **0** = DISABLE **1** = ENABLE
6:0	0x0	**PULSE_N**

7.7.4.22 Audio Generic Interrupt Control (AGIC) Registers

AGIC_DISTRIBUTOR_GICD_CTLR_0

Offset: 0x1000
Read/Write: RW

Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICD_CTLR**: Distributor Control Register

AGIC_DISTRIBUTOR_GICD_TYPER_0

Offset: 0x1004
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICD_TYPER**: Interrupt Controller Type Register

AGIC_DISTRIBUTOR_GICD_IIDR_0

Offset: 0x1008
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICD_IIDR**: Distributor Implementer Identification Register

AGIC_DISTRIBUTOR_GICD_IGROUPR<*i*>_0,

where <*i*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15.

AGIC_DISTRIBUTOR_GICD_IGROUPR0_0
Offset: 0x1080

AGIC_DISTRIBUTOR_GICD_IGROUPR1_0
Offset: 0x1084

AGIC_DISTRIBUTOR_GICD_IGROUPR2_0
Offset: 0x1088

AGIC_DISTRIBUTOR_GICD_IGROUPR3_0
Offset: 0x108c

AGIC_DISTRIBUTOR_GICD_IGROUPR4_0
Offset: 0x1090

AGIC_DISTRIBUTOR_GICD_IGROUPR5_0
Offset: 0x1094

AGIC_DISTRIBUTOR_GICD_IGROUPR6_0
Offset: 0x1098

AGIC_DISTRIBUTOR_GICD_IGROUPR7_0
Offset: 0x109c

AGIC_DISTRIBUTOR_GICD_IGROUPR8_0
Offset: 0x10a0

AGIC_DISTRIBUTOR_GICD_IGROUPR9_0
Offset: 0x10a4

AGIC_DISTRIBUTOR_GICD_IGROUPR10_0
Offset: 0x10a8

AGIC_DISTRIBUTOR_GICD_IGROUPR11_0
Offset: 0x10ac

AGIC_DISTRIBUTOR_GICD_IGROUPR12_0
Offset: 0x10b0

AGIC_DISTRIBUTOR_GICD_IGROUPR13_0
Offset: 0x10b4

AGIC_DISTRIBUTOR_GICD_IGROUPR14_0
Offset: 0x10b8

AGIC_DISTRIBUTOR_GICD_IGROUPR15_0
Offset: 0x10bc

Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICD_IGROUPR**: Interrupt Group Registers

AGIC_DISTRIBUTOR_GICD_ISENABLER<*i*>_0,

where <*i*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15.

AGIC_DISTRIBUTOR_GICD_ISENABLER0_0
Offset: 0x1100

AGIC_DISTRIBUTOR_GICD_ISENABLER1_0
Offset: 0x1104

AGIC_DISTRIBUTOR_GICD_ISENABLER2_0
Offset: 0x1108

AGIC_DISTRIBUTOR_GICD_ISENABLER3_0
Offset: 0x110c

AGIC_DISTRIBUTOR_GICD_ISENABLER4_0
Offset: 0x1110

AGIC_DISTRIBUTOR_GICD_ISENABLER5_0
Offset: 0x1114

AGIC_DISTRIBUTOR_GICD_ISENABLER6_0
Offset: 0x1118

AGIC_DISTRIBUTOR_GICD_ISENABLER7_0
Offset: 0x111c

AGIC_DISTRIBUTOR_GICD_ISENABLER8_0
Offset: 0x1120

AGIC_DISTRIBUTOR_GICD_ISENABLER9_0
Offset: 0x1124

AGIC_DISTRIBUTOR_GICD_ISENABLER10_0
Offset: 0x1128

AGIC_DISTRIBUTOR_GICD_ISENABLER11_0
Offset: 0x112c

AGIC_DISTRIBUTOR_GICD_ISENABLER12_0
Offset: 0x1130

AGIC_DISTRIBUTOR_GICD_ISENABLER13_0
Offset: 0x1134

AGIC_DISTRIBUTOR_GICD_ISENABLER14_0
Offset: 0x1138

AGIC_DISTRIBUTOR_GICD_ISENABLER15_0
Offset: 0x113c

Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICD_ISENABLER**: Interrupt Set-Enable Registers

AGIC_DISTRIBUTOR_GICD_ICENABLER<*i*>_0,

where <*i*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15.

AGIC_DISTRIBUTOR_GICD_ICENABLER0_0
Offset: 0x1180

AGIC_DISTRIBUTOR_GICD_ICENABLER1_0
Offset: 0x1184

AGIC_DISTRIBUTOR_GICD_ICENABLER2_0
Offset: 0x1188

AGIC_DISTRIBUTOR_GICD_ICENABLER3_0
Offset: 0x118c

AGIC_DISTRIBUTOR_GICD_ICENABLER4_0
Offset: 0x1190

AGIC_DISTRIBUTOR_GICD_ICENABLER5_0
Offset: 0x1194

AGIC_DISTRIBUTOR_GICD_ICENABLER6_0
Offset: 0x1198

AGIC_DISTRIBUTOR_GICD_ICENABLER7_0
Offset: 0x119c

AGIC_DISTRIBUTOR_GICD_ICENABLER8_0
Offset: 0x11a0

AGIC_DISTRIBUTOR_GICD_ICENABLER9_0
Offset: 0x11a4

AGIC_DISTRIBUTOR_GICD_ICENABLER10_0
Offset: 0x11a8

AGIC_DISTRIBUTOR_GICD_ICENABLER11_0
Offset: 0x11ac

AGIC_DISTRIBUTOR_GICD_ICENABLER12_0
Offset: 0x11b0

AGIC_DISTRIBUTOR_GICD_ICENABLER13_0
Offset: 0x11b4

AGIC_DISTRIBUTOR_GICD_ICENABLER14_0
Offset: 0x11b8

AGIC_DISTRIBUTOR_GICD_ICENABLER15_0
Offset: 0x11bc

Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICD_ICENABLER**: Interrupt Clear-Enable Registers

AGIC_DISTRIBUTOR_GICD_ISPENDR<*i*>_0,

where <*i*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15.

AGIC_DISTRIBUTOR_GICD_ISPENDR0_0
Offset: 0x1200

AGIC_DISTRIBUTOR_GICD_ISPENDR1_0
Offset: 0x1204

AGIC_DISTRIBUTOR_GICD_ISPENDR2_0
Offset: 0x1208

AGIC_DISTRIBUTOR_GICD_ISPENDR3_0
Offset: 0x120c

AGIC_DISTRIBUTOR_GICD_ISPENDR4_0
Offset: 0x1210

AGIC_DISTRIBUTOR_GICD_ISPENDR5_0
Offset: 0x1214

AGIC_DISTRIBUTOR_GICD_ISPENDR6_0
Offset: 0x1218

AGIC_DISTRIBUTOR_GICD_ISPENDR7_0
Offset: 0x121c

AGIC_DISTRIBUTOR_GICD_ISPENDR8_0
Offset: 0x1220

AGIC_DISTRIBUTOR_GICD_ISPENDR9_0
Offset: 0x1224

AGIC_DISTRIBUTOR_GICD_ISPENDR10_0
Offset: 0x1228

AGIC_DISTRIBUTOR_GICD_ISPENDR11_0
Offset: 0x122c

AGIC_DISTRIBUTOR_GICD_ISPENDR12_0
Offset: 0x1230

AGIC_DISTRIBUTOR_GICD_ISPENDR13_0

Offset: 0x1234

AGIC_DISTRIBUTOR_GICD_ISPENDR14_0
Offset: 0x1238

AGIC_DISTRIBUTOR_GICD_ISPENDR15_0
Offset: 0x123c

Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICD_ISPENDR**: Interrupt Set-Pending Registers

AGIC_DISTRIBUTOR_GICD_ICPENDR<*i*>_0,

where <*i*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15.

AGIC_DISTRIBUTOR_GICD_ICPENDR0_0
Offset: 0x1280

AGIC_DISTRIBUTOR_GICD_ICPENDR1_0
Offset: 0x1284

AGIC_DISTRIBUTOR_GICD_ICPENDR2_0
Offset: 0x1288

AGIC_DISTRIBUTOR_GICD_ICPENDR3_0
Offset: 0x128c

AGIC_DISTRIBUTOR_GICD_ICPENDR4_0
Offset: 0x1290

AGIC_DISTRIBUTOR_GICD_ICPENDR5_0

Offset: 0x1294

AGIC_DISTRIBUTOR_GICD_ICPENDR6_0

Offset: 0x1298

AGIC_DISTRIBUTOR_GICD_ICPENDR7_0

Offset: 0x129c

AGIC_DISTRIBUTOR_GICD_ICPENDR8_0

Offset: 0x12a0

AGIC_DISTRIBUTOR_GICD_ICPENDR9_0

Offset: 0x12a4

AGIC_DISTRIBUTOR_GICD_ICPENDR10_0

Offset: 0x12a8

AGIC_DISTRIBUTOR_GICD_ICPENDR11_0

Offset: 0x12ac

AGIC_DISTRIBUTOR_GICD_ICPENDR12_0
Offset: 0x12b0

AGIC_DISTRIBUTOR_GICD_ICPENDR13_0
Offset: 0x12b4

AGIC_DISTRIBUTOR_GICD_ICPENDR14_0

Offset: 0x12b8

AGIC_DISTRIBUTOR_GICD_ICPENDR15_0
Offset: 0x12bc

Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICD_ICPENDR**: Interrupt Clear-Pending Registers

AGIC_DISTRIBUTOR_GICD_ISACTIVER<*i*>_0,

where <*i*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15.

AGIC_DISTRIBUTOR_GICD_ISACTIVER0_0
Offset: 0x1300

AGIC_DISTRIBUTOR_GICD_ISACTIVER1_0
Offset: 0x1304

AGIC_DISTRIBUTOR_GICD_ISACTIVER2_0
Offset: 0x1308

AGIC_DISTRIBUTOR_GICD_ISACTIVER3_0
Offset: 0x130c

AGIC_DISTRIBUTOR_GICD_ISACTIVER4_0
Offset: 0x1310

AGIC_DISTRIBUTOR_GICD_ISACTIVER5_0
Offset: 0x1314

AGIC_DISTRIBUTOR_GICD_ISACTIVER6_0
Offset: 0x1318

AGIC_DISTRIBUTOR_GICD_ISACTIVER7_0
Offset: 0x131c

AGIC_DISTRIBUTOR_GICD_ISACTIVER8_0
Offset: 0x1320

AGIC_DISTRIBUTOR_GICD_ISACTIVER9_0
Offset: 0x1324

AGIC_DISTRIBUTOR_GICD_ISACTIVER10_0
Offset: 0x1328

AGIC_DISTRIBUTOR_GICD_ISACTIVER11_0
Offset: 0x132c

AGIC_DISTRIBUTOR_GICD_ISACTIVER12_0
Offset: 0x1330

AGIC_DISTRIBUTOR_GICD_ISACTIVER13_0
Offset: 0x1334

AGIC_DISTRIBUTOR_GICD_ISACTIVER14_0
Offset: 0x1338

AGIC_DISTRIBUTOR_GICD_ISACTIVER15_0
Offset: 0x133c

Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICD_ISACTIVER**: Interrupt Set-Active Registers

AGIC_DISTRIBUTOR_GICD_ICACTIVER<*i*>_0,

where <*i*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15.

AGIC_DISTRIBUTOR_GICD_ICACTIVER0_0
Offset: 0x1380

AGIC_DISTRIBUTOR_GICD_ICACTIVER1_0
Offset: 0x1384

AGIC_DISTRIBUTOR_GICD_ICACTIVER2_0
Offset: 0x1388

AGIC_DISTRIBUTOR_GICD_ICACTIVER3_0
Offset: 0x138c

AGIC_DISTRIBUTOR_GICD_ICACTIVER4_0
Offset: 0x1390

AGIC_DISTRIBUTOR_GICD_ICACTIVER5_0
Offset: 0x1394

AGIC_DISTRIBUTOR_GICD_ICACTIVER6_0
Offset: 0x1398

AGIC_DISTRIBUTOR_GICD_ICACTIVER7_0
Offset: 0x139c

AGIC_DISTRIBUTOR_GICD_ICACTIVER8_0
Offset: 0x13a0

AGIC_DISTRIBUTOR_GICD_ICACTIVER9_0
Offset: 0x13a4

AGIC_DISTRIBUTOR_GICD_ICACTIVER10_0
Offset: 0x13a8

AGIC_DISTRIBUTOR_GICD_ICACTIVER11_0
Offset: 0x13ac

AGIC_DISTRIBUTOR_GICD_ICACTIVER12_0
Offset: 0x13b0

AGIC_DISTRIBUTOR_GICD_ICACTIVER13_0
Offset: 0x13b4

AGIC_DISTRIBUTOR_GICD_ICACTIVER14_0
Offset: 0x13b8

AGIC_DISTRIBUTOR_GICD_ICACTIVER15_0
Offset: 0x13bc

Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICD_ICACTIVER**: Interrupt Clear-Active Registers

AGIC_DISTRIBUTOR_GICD_IPRIORITYR<*j*>_0,

where <*j*> = 0, 1, 2, 3, ..., 127.

AGIC_DISTRIBUTOR_GICD_IPRIORITYR0_0
Offset: 0x1400

AGIC_DISTRIBUTOR_GICD_IPRIORITYR1_0

Offset: 0x1404

AGIC_DISTRIBUTOR_GICD_IPRIORITYR2_0

Offset: 0x1408

AGIC_DISTRIBUTOR_GICD_IPRIORITYR3_0
Offset: 0x140c

AGIC_DISTRIBUTOR_GICD_IPRIORITYR4_0
Offset: 0x1410

AGIC_DISTRIBUTOR_GICD_IPRIORITYR5_0
Offset: 0x1414

AGIC_DISTRIBUTOR_GICD_IPRIORITYR6_0
Offset: 0x1418

AGIC_DISTRIBUTOR_GICD_IPRIORITYR7_0
Offset: 0x141c

AGIC_DISTRIBUTOR_GICD_IPRIORITYR8_0
Offset: 0x1420

AGIC_DISTRIBUTOR_GICD_IPRIORITYR9_0
Offset: 0x1424

AGIC_DISTRIBUTOR_GICD_IPRIORITYR10_0
Offset: 0x1428

AGIC_DISTRIBUTOR_GICD_IPRIORITYR11_0
Offset: 0x142c

AGIC_DISTRIBUTOR_GICD_IPRIORITYR12_0
Offset: 0x1430

AGIC_DISTRIBUTOR_GICD_IPRIORITYR13_0
Offset: 0x1434

AGIC_DISTRIBUTOR_GICD_IPRIORITYR14_0
Offset: 0x1438

AGIC_DISTRIBUTOR_GICD_IPRIORITYR15_0
Offset: 0x143c

AGIC_DISTRIBUTOR_GICD_IPRIORITYR16_0
Offset: 0x1440

AGIC_DISTRIBUTOR_GICD_IPRIORITYR17_0
Offset: 0x1444

AGIC_DISTRIBUTOR_GICD_IPRIORITYR18_0
Offset: 0x1448

AGIC_DISTRIBUTOR_GICD_IPRIORITYR19_0
Offset: 0x144c

AGIC_DISTRIBUTOR_GICD_IPRIORITYR20_0
Offset: 0x1450

AGIC_DISTRIBUTOR_GICD_IPRIORITYR21_0
Offset: 0x1454

AGIC_DISTRIBUTOR_GICD_IPRIORITYR22_0
Offset: 0x1458

AGIC_DISTRIBUTOR_GICD_IPRIORITYR23_0
Offset: 0x145c

AGIC_DISTRIBUTOR_GICD_IPRIORITYR24_0|
Offset: 0x1460

AGIC_DISTRIBUTOR_GICD_IPRIORITYR25_0
Offset: 0x1464

AGIC_DISTRIBUTOR_GICD_IPRIORITYR26_0
Offset: 0x1468

AGIC_DISTRIBUTOR_GICD_IPRIORITYR27_0
Offset: 0x146c

AGIC_DISTRIBUTOR_GICD_IPRIORITYR28_0
Offset: 0x1470

AGIC_DISTRIBUTOR_GICD_IPRIORITYR29_0
Offset: 0x1474

AGIC_DISTRIBUTOR_GICD_IPRIORITYR30_0
Offset: 0x1478

AGIC_DISTRIBUTOR_GICD_IPRIORITYR31_0
Offset: 0x147c

AGIC_DISTRIBUTOR_GICD_IPRIORITYR32_0
Offset: 0x1480

AGIC_DISTRIBUTOR_GICD_IPRIORITYR33_0
Offset: 0x1484

AGIC_DISTRIBUTOR_GICD_IPRIORITYR34_0
Offset: 0x1488

AGIC_DISTRIBUTOR_GICD_IPRIORITYR35_0
Offset: 0x148c

AGIC_DISTRIBUTOR_GICD_IPRIORITYR36_0
Offset: 0x1490

AGIC_DISTRIBUTOR_GICD_IPRIORITYR37_0
Offset: 0x1494

AGIC_DISTRIBUTOR_GICD_IPRIORITYR38_0
Offset: 0x1498

AGIC_DISTRIBUTOR_GICD_IPRIORITYR39_0
Offset: 0x149c

AGIC_DISTRIBUTOR_GICD_IPRIORITYR40_0
Offset: 0x14a0

AGIC_DISTRIBUTOR_GICD_IPRIORITYR41_0
Offset: 0x14a4

AGIC_DISTRIBUTOR_GICD_IPRIORITYR42_0
Offset: 0x14a8

AGIC_DISTRIBUTOR_GICD_IPRIORITYR43_0
Offset: 0x14ac

AGIC_DISTRIBUTOR_GICD_IPRIORITYR44_0
Offset: 0x14b0

AGIC_DISTRIBUTOR_GICD_IPRIORITYR45_0
Offset: 0x14b4

AGIC_DISTRIBUTOR_GICD_IPRIORITYR46_0
Offset: 0x14b8

AGIC_DISTRIBUTOR_GICD_IPRIORITYR47_0
Offset: 0x14bc

AGIC_DISTRIBUTOR_GICD_IPRIORITYR48_0
Offset: 0x14c0

AGIC_DISTRIBUTOR_GICD_IPRIORITYR49_0
Offset: 0x14c4

AGIC_DISTRIBUTOR_GICD_IPRIORITYR50_0
Offset: 0x14c8

AGIC_DISTRIBUTOR_GICD_IPRIORITYR51_0
Offset: 0x14cc

AGIC_DISTRIBUTOR_GICD_IPRIORITYR52_0
Offset: 0x14d0

AGIC_DISTRIBUTOR_GICD_IPRIORITYR53_0
Offset: 0x14d4

AGIC_DISTRIBUTOR_GICD_IPRIORITYR54_0
Offset: 0x14d8

AGIC_DISTRIBUTOR_GICD_IPRIORITYR55_0
Offset: 0x14dc

AGIC_DISTRIBUTOR_GICD_IPRIORITYR56_0
Offset: 0x14e0

AGIC_DISTRIBUTOR_GICD_IPRIORITYR57_0
Offset: 0x14e4

AGIC_DISTRIBUTOR_GICD_IPRIORITYR58_0
Offset: 0x14e8

AGIC_DISTRIBUTOR_GICD_IPRIORITYR59_0
Offset: 0x14ec

AGIC_DISTRIBUTOR_GICD_IPRIORITYR60_0
Offset: 0x14f0

AGIC_DISTRIBUTOR_GICD_IPRIORITYR61_0
Offset: 0x14f4

AGIC_DISTRIBUTOR_GICD_IPRIORITYR62_0
Offset: 0x14f8

AGIC_DISTRIBUTOR_GICD_IPRIORITYR63_0
Offset: 0x14fc

AGIC_DISTRIBUTOR_GICD_IPRIORITYR64_0
Offset: 0x1500

AGIC_DISTRIBUTOR_GICD_IPRIORITYR65_0
Offset: 0x1504

AGIC_DISTRIBUTOR_GICD_IPRIORITYR66_0
Offset: 0x1508

AGIC_DISTRIBUTOR_GICD_IPRIORITYR67_0
Offset: 0x150c

AGIC_DISTRIBUTOR_GICD_IPRIORITYR68_0
Offset: 0x1510

AGIC_DISTRIBUTOR_GICD_IPRIORITYR69_0
Offset: 0x1514

AGIC_DISTRIBUTOR_GICD_IPRIORITYR70_0
Offset: 0x1518

AGIC_DISTRIBUTOR_GICD_IPRIORITYR71_0
Offset: 0x151c

AGIC_DISTRIBUTOR_GICD_IPRIORITYR72_0
Offset: 0x1520

AGIC_DISTRIBUTOR_GICD_IPRIORITYR73_0
Offset: 0x1524

AGIC_DISTRIBUTOR_GICD_IPRIORITYR74_0
Offset: 0x1528

AGIC_DISTRIBUTOR_GICD_IPRIORITYR75_0
Offset: 0x152c

AGIC_DISTRIBUTOR_GICD_IPRIORITYR76_0
Offset: 0x1530

AGIC_DISTRIBUTOR_GICD_IPRIORITYR77_0
Offset: 0x1534

AGIC_DISTRIBUTOR_GICD_IPRIORITYR78_0
Offset: 0x1538

AGIC_DISTRIBUTOR_GICD_IPRIORITYR79_0
Offset: 0x153c

AGIC_DISTRIBUTOR_GICD_IPRIORITYR80_0
Offset: 0x1540

AGIC_DISTRIBUTOR_GICD_IPRIORITYR81_0
Offset: 0x1544

AGIC_DISTRIBUTOR_GICD_IPRIORITYR82_0
Offset: 0x1548

AGIC_DISTRIBUTOR_GICD_IPRIORITYR83_0
Offset: 0x154c

AGIC_DISTRIBUTOR_GICD_IPRIORITYR84_0
Offset: 0x1550

AGIC_DISTRIBUTOR_GICD_IPRIORITYR85_0
Offset: 0x1554

AGIC_DISTRIBUTOR_GICD_IPRIORITYR86_0
Offset: 0x1558

AGIC_DISTRIBUTOR_GICD_IPRIORITYR87_0
Offset: 0x155c

AGIC_DISTRIBUTOR_GICD_IPRIORITYR88_0
Offset: 0x1560

AGIC_DISTRIBUTOR_GICD_IPRIORITYR89_0
Offset: 0x1564

AGIC_DISTRIBUTOR_GICD_IPRIORITYR90_0
Offset: 0x1568

AGIC_DISTRIBUTOR_GICD_IPRIORITYR91_0
Offset: 0x156c

AGIC_DISTRIBUTOR_GICD_IPRIORITYR92_0
Offset: 0x1570

AGIC_DISTRIBUTOR_GICD_IPRIORITYR93_0
Offset: 0x1574

AGIC_DISTRIBUTOR_GICD_IPRIORITYR94_0
Offset: 0x1578

AGIC_DISTRIBUTOR_GICD_IPRIORITYR95_0
Offset: 0x157c

AGIC_DISTRIBUTOR_GICD_IPRIORITYR96_0
Offset: 0x1580

AGIC_DISTRIBUTOR_GICD_IPRIORITYR97_0
Offset: 0x1584

AGIC_DISTRIBUTOR_GICD_IPRIORITYR98_0
Offset: 0x1588

AGIC_DISTRIBUTOR_GICD_IPRIORITYR99_0
Offset: 0x158c

AGIC_DISTRIBUTOR_GICD_IPRIORITYR100_0
Offset: 0x1590

AGIC_DISTRIBUTOR_GICD_IPRIORITYR101_0
Offset: 0x1594

AGIC_DISTRIBUTOR_GICD_IPRIORITYR102_0
Offset: 0x1598

AGIC_DISTRIBUTOR_GICD_IPRIORITYR103_0
Offset: 0x159c

AGIC_DISTRIBUTOR_GICD_IPRIORITYR104_0
Offset: 0x15a0

AGIC_DISTRIBUTOR_GICD_IPRIORITYR105_0
Offset: 0x15a4

AGIC_DISTRIBUTOR_GICD_IPRIORITYR106_0
Offset: 0x15a8

AGIC_DISTRIBUTOR_GICD_IPRIORITYR107_0
Offset: 0x15ac

AGIC_DISTRIBUTOR_GICD_IPRIORITYR108_0
Offset: 0x15b0

AGIC_DISTRIBUTOR_GICD_IPRIORITYR109_0
Offset: 0x15b4

AGIC_DISTRIBUTOR_GICD_IPRIORITYR110_0
Offset: 0x15b8

AGIC_DISTRIBUTOR_GICD_IPRIORITYR111_0
Offset: 0x15bc

AGIC_DISTRIBUTOR_GICD_IPRIORITYR112_0
Offset: 0x15c0

AGIC_DISTRIBUTOR_GICD_IPRIORITYR113_0
Offset: 0x15c4

AGIC_DISTRIBUTOR_GICD_IPRIORITYR114_0
Offset: 0x15c8

AGIC_DISTRIBUTOR_GICD_IPRIORITYR115_0
Offset: 0x15cc

AGIC_DISTRIBUTOR_GICD_IPRIORITYR116_0
Offset: 0x15d0

AGIC_DISTRIBUTOR_GICD_IPRIORITYR117_0
Offset: 0x15d4

AGIC_DISTRIBUTOR_GICD_IPRIORITYR118_0
Offset: 0x15d8

AGIC_DISTRIBUTOR_GICD_IPRIORITYR119_0
Offset: 0x15dc

AGIC_DISTRIBUTOR_GICD_IPRIORITYR120_0
Offset: 0x15e0

AGIC_DISTRIBUTOR_GICD_IPRIORITYR121_0
Offset: 0x15e4

AGIC_DISTRIBUTOR_GICD_IPRIORITYR122_0
Offset: 0x15e8

AGIC_DISTRIBUTOR_GICD_IPRIORITYR123_0
Offset: 0x15ec

AGIC_DISTRIBUTOR_GICD_IPRIORITYR124_0
Offset: 0x15f0

AGIC_DISTRIBUTOR_GICD_IPRIORITYR125_0
Offset: 0x15f4

AGIC_DISTRIBUTOR_GICD_IPRIORITYR126_0
Offset: 0x15f8

AGIC_DISTRIBUTOR_GICD_IPRIORITYR127_0
Offset: 0x15fc

Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICD_IPRIORITYR**: Interrupt Priority Registers

AGIC_DISTRIBUTOR_GICD_ITARGETSR<*j*>_0,

where <*j*> = 0, 1, 2, 3, ..., 127.

AGIC_DISTRIBUTOR_GICD_ITARGETSR0_0
Offset: 0x1800

AGIC_DISTRIBUTOR_GICD_ITARGETSR1_0
Offset: 0x1804

AGIC_DISTRIBUTOR_GICD_ITARGETSR2_0
Offset: 0x1808

AGIC_DISTRIBUTOR_GICD_ITARGETSR3_0
Offset: 0x180c

AGIC_DISTRIBUTOR_GICD_ITARGETSR4_0
Offset: 0x1810

AGIC_DISTRIBUTOR_GICD_ITARGETSR5_0
Offset: 0x1814

AGIC_DISTRIBUTOR_GICD_ITARGETSR6_0
Offset: 0x1818

AGIC_DISTRIBUTOR_GICD_ITARGETSR7_0
Offset: 0x181c

AGIC_DISTRIBUTOR_GICD_ITARGETSR8_0
Offset: 0x1820

AGIC_DISTRIBUTOR_GICD_ITARGETSR9_0
Offset: 0x1824

AGIC_DISTRIBUTOR_GICD_ITARGETSR10_0
Offset: 0x1828

AGIC_DISTRIBUTOR_GICD_ITARGETSR11_0
Offset: 0x182c

AGIC_DISTRIBUTOR_GICD_ITARGETSR12_0
Offset: 0x1830

AGIC_DISTRIBUTOR_GICD_ITARGETSR13_0
Offset: 0x1834

AGIC_DISTRIBUTOR_GICD_ITARGETSR14_0
Offset: 0x1838

AGIC_DISTRIBUTOR_GICD_ITARGETSR15_0
Offset: 0x183c

AGIC_DISTRIBUTOR_GICD_ITARGETSR16_0
Offset: 0x1840

AGIC_DISTRIBUTOR_GICD_ITARGETSR17_0
Offset: 0x1844

AGIC_DISTRIBUTOR_GICD_ITARGETSR18_0
Offset: 0x1848

AGIC_DISTRIBUTOR_GICD_ITARGETSR19_0
Offset: 0x184c

AGIC_DISTRIBUTOR_GICD_ITARGETSR20_0
Offset: 0x1850

AGIC_DISTRIBUTOR_GICD_ITARGETSR21_0
Offset: 0x1854

AGIC_DISTRIBUTOR_GICD_ITARGETSR22_0
Offset: 0x1858

AGIC_DISTRIBUTOR_GICD_ITARGETSR23_0
Offset: 0x185c

AGIC_DISTRIBUTOR_GICD_ITARGETSR24_0
Offset: 0x1860

AGIC_DISTRIBUTOR_GICD_ITARGETSR25_0
Offset: 0x1864

AGIC_DISTRIBUTOR_GICD_ITARGETSR26_0
Offset: 0x1868

AGIC_DISTRIBUTOR_GICD_ITARGETSR27_0
Offset: 0x186c

AGIC_DISTRIBUTOR_GICD_ITARGETSR28_0
Offset: 0x1870

AGIC_DISTRIBUTOR_GICD_ITARGETSR29_0
Offset: 0x1874

AGIC_DISTRIBUTOR_GICD_ITARGETSR30_0
Offset: 0x1878

AGIC_DISTRIBUTOR_GICD_ITARGETSR31_0
Offset: 0x187c

AGIC_DISTRIBUTOR_GICD_ITARGETSR32_0
Offset: 0x1880

AGIC_DISTRIBUTOR_GICD_ITARGETSR33_0
Offset: 0x1884

AGIC_DISTRIBUTOR_GICD_ITARGETSR34_0
Offset: 0x1888

AGIC_DISTRIBUTOR_GICD_ITARGETSR35_0
Offset: 0x188c

AGIC_DISTRIBUTOR_GICD_ITARGETSR36_0
Offset: 0x1890

AGIC_DISTRIBUTOR_GICD_ITARGETSR37_0
Offset: 0x1894

AGIC_DISTRIBUTOR_GICD_ITARGETSR38_0
Offset: 0x1898

AGIC_DISTRIBUTOR_GICD_ITARGETSR39_0
Offset: 0x189c

AGIC_DISTRIBUTOR_GICD_ITARGETSR40_0
Offset: 0x18a0

AGIC_DISTRIBUTOR_GICD_ITARGETSR41_0
Offset: 0x18a4

AGIC_DISTRIBUTOR_GICD_ITARGETSR42_0
Offset: 0x18a8

AGIC_DISTRIBUTOR_GICD_ITARGETSR43_0
Offset: 0x18ac

AGIC_DISTRIBUTOR_GICD_ITARGETSR44_0
Offset: 0x18b0

AGIC_DISTRIBUTOR_GICD_ITARGETSR45_0
Offset: 0x18b4

AGIC_DISTRIBUTOR_GICD_ITARGETSR46_0
Offset: 0x18b8

AGIC_DISTRIBUTOR_GICD_ITARGETSR47_0
Offset: 0x18bc

AGIC_DISTRIBUTOR_GICD_ITARGETSR48_0
Offset: 0x18c0

AGIC_DISTRIBUTOR_GICD_ITARGETSR49_0
Offset: 0x18c4

AGIC_DISTRIBUTOR_GICD_ITARGETSR50_0
Offset: 0x18c8

AGIC_DISTRIBUTOR_GICD_ITARGETSR51_0
Offset: 0x18cc

AGIC_DISTRIBUTOR_GICD_ITARGETSR52_0
Offset: 0x18d0

AGIC_DISTRIBUTOR_GICD_ITARGETSR53_0
Offset: 0x18d4

AGIC_DISTRIBUTOR_GICD_ITARGETSR54_0
Offset: 0x18d8

AGIC_DISTRIBUTOR_GICD_ITARGETSR55_0
Offset: 0x18dc

AGIC_DISTRIBUTOR_GICD_ITARGETSR56_0
Offset: 0x18e0

AGIC_DISTRIBUTOR_GICD_ITARGETSR57_0
Offset: 0x18e4

AGIC_DISTRIBUTOR_GICD_ITARGETSR58_0
Offset: 0x18e8

AGIC_DISTRIBUTOR_GICD_ITARGETSR59_0
Offset: 0x18ec

AGIC_DISTRIBUTOR_GICD_ITARGETSR60_0
Offset: 0x18f0

AGIC_DISTRIBUTOR_GICD_ITARGETSR61_0
Offset: 0x18f4

AGIC_DISTRIBUTOR_GICD_ITARGETSR62_0
Offset: 0x18f8

AGIC_DISTRIBUTOR_GICD_ITARGETSR63_0
Offset: 0x18fc

AGIC_DISTRIBUTOR_GICD_ITARGETSR64_0
Offset: 0x1900

AGIC_DISTRIBUTOR_GICD_ITARGETSR65_0
Offset: 0x1904

AGIC_DISTRIBUTOR_GICD_ITARGETSR66_0
Offset: 0x1908

AGIC_DISTRIBUTOR_GICD_ITARGETSR67_0
Offset: 0x190c

AGIC_DISTRIBUTOR_GICD_ITARGETSR68_0
Offset: 0x1910

AGIC_DISTRIBUTOR_GICD_ITARGETSR69_0
Offset: 0x1914

AGIC_DISTRIBUTOR_GICD_ITARGETSR70_0
Offset: 0x1918

AGIC_DISTRIBUTOR_GICD_ITARGETSR71_0
Offset: 0x191c

AGIC_DISTRIBUTOR_GICD_ITARGETSR72_0
Offset: 0x1920

AGIC_DISTRIBUTOR_GICD_ITARGETSR73_0
Offset: 0x1924

AGIC_DISTRIBUTOR_GICD_ITARGETSR74_0
Offset: 0x1928

AGIC_DISTRIBUTOR_GICD_ITARGETSR75_0
Offset: 0x192c

AGIC_DISTRIBUTOR_GICD_ITARGETSR76_0
Offset: 0x1930

AGIC_DISTRIBUTOR_GICD_ITARGETSR77_0
Offset: 0x1934

AGIC_DISTRIBUTOR_GICD_ITARGETSR78_0
Offset: 0x1938

AGIC_DISTRIBUTOR_GICD_ITARGETSR79_0
Offset: 0x193c

AGIC_DISTRIBUTOR_GICD_ITARGETSR80_0
Offset: 0x1940

AGIC_DISTRIBUTOR_GICD_ITARGETSR81_0
Offset: 0x1944

AGIC_DISTRIBUTOR_GICD_ITARGETSR82_0
Offset: 0x1948

AGIC_DISTRIBUTOR_GICD_ITARGETSR83_0
Offset: 0x194c

AGIC_DISTRIBUTOR_GICD_ITARGETSR84_0
Offset: 0x1950

AGIC_DISTRIBUTOR_GICD_ITARGETSR85_0
Offset: 0x1954

AGIC_DISTRIBUTOR_GICD_ITARGETSR86_0
Offset: 0x1958

AGIC_DISTRIBUTOR_GICD_ITARGETSR87_0
Offset: 0x195c

AGIC_DISTRIBUTOR_GICD_ITARGETSR88_0
Offset: 0x1960

AGIC_DISTRIBUTOR_GICD_ITARGETSR89_0
Offset: 0x1964

AGIC_DISTRIBUTOR_GICD_ITARGETSR90_0
Offset: 0x1968

AGIC_DISTRIBUTOR_GICD_ITARGETSR91_0
Offset: 0x196c

AGIC_DISTRIBUTOR_GICD_ITARGETSR92_0
Offset: 0x1970

AGIC_DISTRIBUTOR_GICD_ITARGETSR93_0
Offset: 0x1974

AGIC_DISTRIBUTOR_GICD_ITARGETSR94_0
Offset: 0x1978

AGIC_DISTRIBUTOR_GICD_ITARGETSR95_0
Offset: 0x197c

AGIC_DISTRIBUTOR_GICD_ITARGETSR96_0
Offset: 0x1980

AGIC_DISTRIBUTOR_GICD_ITARGETSR97_0
Offset: 0x1984

AGIC_DISTRIBUTOR_GICD_ITARGETSR98_0
Offset: 0x1988

AGIC_DISTRIBUTOR_GICD_ITARGETSR99_0
Offset: 0x198c

AGIC_DISTRIBUTOR_GICD_ITARGETSR100_0
Offset: 0x1990

AGIC_DISTRIBUTOR_GICD_ITARGETSR101_0
Offset: 0x1994

AGIC_DISTRIBUTOR_GICD_ITARGETSR102_0
Offset: 0x1998

AGIC_DISTRIBUTOR_GICD_ITARGETSR103_0
Offset: 0x199c

AGIC_DISTRIBUTOR_GICD_ITARGETSR104_0
Offset: 0x19a0

AGIC_DISTRIBUTOR_GICD_ITARGETSR105_0
Offset: 0x19a4

AGIC_DISTRIBUTOR_GICD_ITARGETSR106_0
Offset: 0x19a8

AGIC_DISTRIBUTOR_GICD_ITARGETSR107_0
Offset: 0x19ac

AGIC_DISTRIBUTOR_GICD_ITARGETSR108_0
Offset: 0x19b0

AGIC_DISTRIBUTOR_GICD_ITARGETSR109_0
Offset: 0x19b4

AGIC_DISTRIBUTOR_GICD_ITARGETSR110_0
Offset: 0x19b8

AGIC_DISTRIBUTOR_GICD_ITARGETSR111_0
Offset: 0x19bc

AGIC_DISTRIBUTOR_GICD_ITARGETSR112_0
Offset: 0x19c0

AGIC_DISTRIBUTOR_GICD_ITARGETSR113_0
Offset: 0x19c4

AGIC_DISTRIBUTOR_GICD_ITARGETSR114_0
Offset: 0x19c8

AGIC_DISTRIBUTOR_GICD_ITARGETSR115_0
Offset: 0x19cc

AGIC_DISTRIBUTOR_GICD_ITARGETSR116_0
Offset: 0x19d0

AGIC_DISTRIBUTOR_GICD_ITARGETSR117_0
Offset: 0x19d4

AGIC_DISTRIBUTOR_GICD_ITARGETSR118_0
Offset: 0x19d8

AGIC_DISTRIBUTOR_GICD_ITARGETSR119_0
Offset: 0x19dc

AGIC_DISTRIBUTOR_GICD_ITARGETSR120_0
Offset: 0x19e0

AGIC_DISTRIBUTOR_GICD_ITARGETSR121_0
Offset: 0x19e4

AGIC_DISTRIBUTOR_GICD_ITARGETSR122_0
Offset: 0x19e8

AGIC_DISTRIBUTOR_GICD_ITARGETSR123_0
Offset: 0x19ec

AGIC_DISTRIBUTOR_GICD_ITARGETSR124_0
Offset: 0x19f0

AGIC_DISTRIBUTOR_GICD_ITARGETSR125_0
Offset: 0x19f4

AGIC_DISTRIBUTOR_GICD_ITARGETSR126_0
Offset: 0x19f8

AGIC_DISTRIBUTOR_GICD_ITARGETSR127_0
Offset: 0x19fc

Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICD_ITARGETSR**: Interrupt Processor Targets Registers

AGIC_DISTRIBUTOR_GICD_ICFGR<*k*>_0,

where <*k*> = 0, 1, 2, 3, ..., 31.

AGIC_DISTRIBUTOR_GICD_ICFGR0_0
Offset: 0x1c00

AGIC_DISTRIBUTOR_GICD_ICFGR1_0
Offset: 0x1c04

AGIC_DISTRIBUTOR_GICD_ICFGR2_0
Offset: 0x1c08

AGIC_DISTRIBUTOR_GICD_ICFGR3_0
Offset: 0x1c0c

AGIC_DISTRIBUTOR_GICD_ICFGR4_0
Offset: 0x1c10

AGIC_DISTRIBUTOR_GICD_ICFGR5_0
Offset: 0x1c14

AGIC_DISTRIBUTOR_GICD_ICFGR6_0
Offset: 0x1c18

AGIC_DISTRIBUTOR_GICD_ICFGR7_0
Offset: 0x1c1c

AGIC_DISTRIBUTOR_GICD_ICFGR8_0
Offset: 0x1c20

AGIC_DISTRIBUTOR_GICD_ICFGR9_0
Offset: 0x1c24

AGIC_DISTRIBUTOR_GICD_ICFGR10_0
Offset: 0x1c28

AGIC_DISTRIBUTOR_GICD_ICFGR11_0
Offset: 0x1c2c

AGIC_DISTRIBUTOR_GICD_ICFGR12_0
Offset: 0x1c30

AGIC_DISTRIBUTOR_GICD_ICFGR13_0
Offset: 0x1c34

AGIC_DISTRIBUTOR_GICD_ICFGR14_0
Offset: 0x1c38

AGIC_DISTRIBUTOR_GICD_ICFGR15_0
Offset: 0x1c3c

AGIC_DISTRIBUTOR_GICD_ICFGR16_0
Offset: 0x1c40

AGIC_DISTRIBUTOR_GICD_ICFGR17_0
Offset: 0x1c44

AGIC_DISTRIBUTOR_GICD_ICFGR18_0
Offset: 0x1c48

AGIC_DISTRIBUTOR_GICD_ICFGR19_0
Offset: 0x1c4c

AGIC_DISTRIBUTOR_GICD_ICFGR20_0
Offset: 0x1c50

AGIC_DISTRIBUTOR_GICD_ICFGR21_0
Offset: 0x1c54

AGIC_DISTRIBUTOR_GICD_ICFGR22_0
Offset: 0x1c58

AGIC_DISTRIBUTOR_GICD_ICFGR23_0
Offset: 0x1c5c

AGIC_DISTRIBUTOR_GICD_ICFGR24_0
Offset: 0x1c60

AGIC_DISTRIBUTOR_GICD_ICFGR25_0
Offset: 0x1c64

AGIC_DISTRIBUTOR_GICD_ICFGR26_0
Offset: 0x1c68

AGIC_DISTRIBUTOR_GICD_ICFGR27_0
Offset: 0x1c6c

AGIC_DISTRIBUTOR_GICD_ICFGR28_0
Offset: 0x1c70

AGIC_DISTRIBUTOR_GICD_ICFGR29_0
Offset: 0x1c74

AGIC_DISTRIBUTOR_GICD_ICFGR30_0
Offset: 0x1c78

AGIC_DISTRIBUTOR_GICD_ICFGR31_0
Offset: 0x1c7c

Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICD_ICFGR**: Interrupt Configuration Registers

AGIC_DISTRIBUTOR_GICD_PPISR_0

Offset: 0x1d00
Read/Write: RO

Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICD_PPISR**: Private Peripheral Interrupt Status Register

AGIC_DISTRIBUTOR_GICD_SPISR<*m*>_0,

where <*m*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14.

AGIC_DISTRIBUTOR_GICD_SPISR0_0
Offset: 0x1d04

AGIC_DISTRIBUTOR_GICD_SPISR1_0
Offset: 0x1d08

AGIC_DISTRIBUTOR_GICD_SPISR2_0
Offset: 0x1d0c

AGIC_DISTRIBUTOR_GICD_SPISR3_0
Offset: 0x1d10

AGIC_DISTRIBUTOR_GICD_SPISR4_0
Offset: 0x1d14

AGIC_DISTRIBUTOR_GICD_SPISR5_0
Offset: 0x1d18

AGIC_DISTRIBUTOR_GICD_SPISR6_0
Offset: 0x1d1c

AGIC_DISTRIBUTOR_GICD_SPISR7_0
Offset: 0x1d20

AGIC_DISTRIBUTOR_GICD_SPISR8_0
Offset: 0x1d24

AGIC_DISTRIBUTOR_GICD_SPISR9_0
Offset: 0x1d28

AGIC_DISTRIBUTOR_GICD_SPISR10_0
Offset: 0x1d2c

AGIC_DISTRIBUTOR_GICD_SPISR11_0
Offset: 0x1d30

AGIC_DISTRIBUTOR_GICD_SPISR12_0
Offset: 0x1d34

AGIC_DISTRIBUTOR_GICD_SPISR13_0
Offset: 0x1d38

AGIC_DISTRIBUTOR_GICD_SPISR14_0
Offset: 0x1d3c

Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICD_SPISR**: Shared Peripheral Interrupt Status Registers

AGIC_DISTRIBUTOR_GICD_SGIR_0

Offset: 0x1f00
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICD_SGIR**: Software Generated Interrupt Register

AGIC_DISTRIBUTOR_GICD_CPENDSGIR<*n*>_0,

where <*n*> = 0, 1, 2, 3.

AGIC_DISTRIBUTOR_GICD_CPENDSGIR0_0
Offset: 0x1f10

AGIC_DISTRIBUTOR_GICD_CPENDSGIR1_0
Offset: 0x1f14

AGIC_DISTRIBUTOR_GICD_CPENDSGIR2_0
Offset: 0x1f18

AGIC_DISTRIBUTOR_GICD_CPENDSGIR3_0
Offset: 0x1f1c

Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICD_CPENDSGIR**: SGI Clear-Pending Registers

AGIC_DISTRIBUTOR_GICD_SPENDSGIR<*n*>_0,

where <*n*> = 0, 1, 2, 3.

AGIC_DISTRIBUTOR_GICD_SPENDSGIR0_0
Offset: 0x1f20

AGIC_DISTRIBUTOR_GICD_SPENDSGIR1_0
Offset: 0x1f24

AGIC_DISTRIBUTOR_GICD_SPENDSGIR2_0
Offset: 0x1f28

AGIC_DISTRIBUTOR_GICD_SPENDSGIR3_0
Offset: 0x1f2c

Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICD_SPENDSGIR**: SGI Set-Pending Registers

AGIC_DISTRIBUTOR_GICD_PIDR<*p*>_0,

where <*p*> = 0, 1, 2, 3, 4, 5, 6, 7.

AGIC_DISTRIBUTOR_GICD_PIDR0_0
Offset: 0x1fe0

AGIC_DISTRIBUTOR_GICD_PIDR1_0
Offset: 0x1fe4

AGIC_DISTRIBUTOR_GICD_PIDR2_0
Offset: 0x1fe8

AGIC_DISTRIBUTOR_GICD_PIDR3_0
Offset: 0x1fec

AGIC_DISTRIBUTOR_GICD_PIDR4_0
Offset: 0x1fd0

AGIC_DISTRIBUTOR_GICD_PIDR5_0
Offset: 0x1fd4

AGIC_DISTRIBUTOR_GICD_PIDR6_0
Offset: 0x1fd8

AGIC_DISTRIBUTOR_GICD_PIDR7_0
Offset: 0x1fdc

Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICD_PIDR**: Peripheral ID Register

AGIC_DISTRIBUTOR_GICD_CIDR<*p*>_0,

where <*p*> = 0, 1, 2, 3.

AGIC_DISTRIBUTOR_GICD_CIDR0_0
Offset: 0x1ff0

AGIC_DISTRIBUTOR_GICD_CIDR1_0
Offset: 0x1ff4

AGIC_DISTRIBUTOR_GICD_CIDR2_0
Offset: 0x1ff8

AGIC_DISTRIBUTOR_GICD_CIDR3_0
Offset: 0x1ffc

Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICD_CIDR**: Component ID Register

AGIC_CPUIF_GICC_CTLR_0

Offset: 0x2000
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICC_CTLR**: CPU Interface Control Register

AGIC_CPUIF_GICC_PMR_0

Offset: 0x2004
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICC_PMR**: Interrupt Priority Mask Register

AGIC_CPUIF_GICC_BPR_0

Offset: 0x2008
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICC_BPR**: Binary Point Register

AGIC_CPUIF_GICC_IAR_0

Offset: 0x200c
Read/Write: RO

Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICC_IAR**: Interrupt Acknowledge Register

AGIC_CPUIF_GICC_EOIR_0

Offset: 0x2010
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICC_EOIR**: End of Interrupt Register

AGIC_CPUIF_GICC_RPR_0

Offset: 0x2014
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICC_RPR**: Running Priority Register

AGIC_CPUIF_GICC_HPPIR_0

Offset: 0x2018
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICC_HPPIR**: Highest Priority Pending Interrupt Register

AGIC_CPUIF_GICC_ABPR_0

Offset: 0x201c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICC_ABPR**: Aliased Binary Point Register

AGIC_CPUIF_GICC_AIAR_0

Offset: 0x2020
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICC_AIAR**: Aliased Interrupt Acknowledge Register

AGIC_CPUIF_GICC_AEOIR_0

Offset: 0x2024
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICC_AEOIR**: Aliased End of Interrupt Register

AGIC_CPUIF_GICC_AHPPIR_0

Offset: 0x2028
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICC_AHPPIR**: Aliased Highest Priority Pending Interrupt Register

AGIC_CPUIF_GICC_APR0_0

Offset: 0x20d0
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICC_APR0**: Active Priority Register

AGIC_CPUIF_GICC_NSAPR0_0

Offset: 0x20e0
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICC_NSAPR0**: Non-Secure Active Priority Register

AGIC_CPUIF_GICC_IIDR_0

Offset: 0x20fc
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICC_IIDR**: CPU Interface Identification Register

AGIC_CPUIF_GICC_DIR_0

Offset: 0x3000
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICC_DIR**: Deactivate Interrupt Register

AGIC_VCPUIFHYP_GICH_HCR_0

Offset: 0x4000
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICH_HCR**: Hypervisor Control Register

AGIC_VCPUIFHYPALIAS<*p*>_GICH_HCR_0,

where <*p*> = 0, 1, 2, 3, 4, 5, 6, 7.

AGIC_VCPUIFHYPALIAS0_GICH_HCR_0
Offset: 0x5000

AGIC_VCPUIFHYPALIAS1_GICH_HCR_0
Offset: 0x5200

AGIC_VCPUIFHYPALIAS2_GICH_HCR_0
Offset: 0x5400

AGIC_VCPUIFHYPALIAS3_GICH_HCR_0
Offset: 0x5600

AGIC_VCPUIFHYPALIAS4_GICH_HCR_0
Offset: 0x5800

AGIC_VCPUIFHYPALIAS5_GICH_HCR_0
Offset: 0x5a00

AGIC_VCPUIFHYPALIAS6_GICH_HCR_0
Offset: 0x5c00

AGIC_VCPUIFHYPALIAS7_GICH_HCR_0
Offset: 0x5e00

Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICH_HCR**: Hypervisor Control Register

AGIC_VCPUIFHYP_GICH_VTR_0

Offset: 0x4004
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICH_VTR**: VGIC Type Register

AGIC_VCPUIFHYPALIAS<*p*>_GICH_VTR_0,

where <*p*> = 0, 1, 2, 3, 4, 5, 6, 7.

AGIC_VCPUIFHYPALIAS0_GICH_VTR_0
Offset: 0x5004

AGIC_VCPUIFHYPALIAS1_GICH_VTR_0

Offset: 0x5204

AGIC_VCPUIFHYPALIAS2_GICH_VTR_0
Offset: 0x5404

AGIC_VCPUIFHYPALIAS3_GICH_VTR_0
Offset: 0x5604

AGIC_VCPUIFHYPALIAS4_GICH_VTR_0
Offset: 0x5804

AGIC_VCPUIFHYPALIAS5_GICH_VTR_0
Offset: 0x5a04

AGIC_VCPUIFHYPALIAS6_GICH_VTR_0
Offset: 0x5c04

AGIC_VCPUIFHYPALIAS7_GICH_VTR_0
Offset: 0x5e04

Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICH_VTR**: VGIC Type Register

AGIC_VCPUIFHYP_GICH_VMCR_0

Offset: 0x4008
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICH_VMCR**: Virtual Machine Control Register

AGIC_VCPUIFHYPALIAS<*p*>_GICH_VMCR_0,

where <*p*> = 0, 1, 2, 3, 4, 5, 6, 7.

AGIC_VCPUIFHYPALIAS0_GICH_VMCR_0
Offset: 0x5008

AGIC_VCPUIFHYPALIAS1_GICH_VMCR_0
Offset: 0x5208

AGIC_VCPUIFHYPALIAS2_GICH_VMCR_0
Offset: 0x5408

AGIC_VCPUIFHYPALIAS3_GICH_VMCR_0
Offset: 0x5608

AGIC_VCPUIFHYPALIAS4_GICH_VMCR_0
Offset: 0x5808

AGIC_VCPUIFHYPALIAS5_GICH_VMCR_0
Offset: 0x5a08

AGIC_VCPUIFHYPALIAS6_GICH_VMCR_0
Offset: 0x5c08

AGIC_VCPUIFHYPALIAS7_GICH_VMCR_0
Offset: 0x5e08

Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICH_VMCR**: Virtual Machine Control Register

AGIC_VCPUIFHYP_GICH_MISR_0

Offset: 0x4010
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICH_MISR**: Maintenance Interrupt Status Register

AGIC_VCPUIFHYPALIAS<*p*>_GICH_MISR_0,

where <*p*> = 0, 1, 2, 3, 4, 5, 6, 7.

AGIC_VCPUIFHYPALIAS0_GICH_MISR_0
Offset: 0x5010

AGIC_VCPUIFHYPALIAS1_GICH_MISR_0
Offset: 0x5210

AGIC_VCPUIFHYPALIAS2_GICH_MISR_0
Offset: 0x5410

AGIC_VCPUIFHYPALIAS3_GICH_MISR_0
Offset: 0x5610

AGIC_VCPUIFHYPALIAS4_GICH_MISR_0
Offset: 0x5810

AGIC_VCPUIFHYPALIAS5_GICH_MISR_0
Offset: 0x5a10

AGIC_VCPUIFHYPALIAS6_GICH_MISR_0
Offset: 0x5c10

AGIC_VCPUIFHYPALIAS7_GICH_MISR_0
Offset: 0x5e10

Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICH_MISR**: Maintenance Interrupt Status Register

AGIC_VCPUIFHYP_GICH_EISR0_0

Offset: 0x4020
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICH_EISR0**: End of Interrupt Status Register

AGIC_VCPUIFHYPALIAS<*p*>_GICH_EISR0_0,

where <*p*> = 0, 1, 2, 3, 4, 5, 6, 7.

AGIC_VCPUIFHYPALIAS0_GICH_EISR0_0
Offset: 0x5020

AGIC_VCPUIFHYPALIAS1_GICH_EISR0_0
Offset: 0x5220

AGIC_VCPUIFHYPALIAS2_GICH_EISR0_0
Offset: 0x5420

AGIC_VCPUIFHYPALIAS3_GICH_EISR0_0
Offset: 0x5620

AGIC_VCPUIFHYPALIAS4_GICH_EISR0_0
Offset: 0x5820

AGIC_VCPUIFHYPALIAS5_GICH_EISR0_0
Offset: 0x5a20

AGIC_VCPUIFHYPALIAS6_GICH_EISR0_0
Offset: 0x5c20

AGIC_VCPUIFHYPALIAS7_GICH_EISR0_0
Offset: 0x5e20

Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICH_EISR0**: End of Interrupt Status Register

AGIC_VCPUIFHYP_GICH_ELSR0_0

Offset: 0x4030
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICH_ELSR0**: Empty List register Status Register

AGIC_VCPUIFHYPALIAS<*p*>_GICH_ELSR0_0,

where <*p*> = 0, 1, 2, 3, 4, 5, 6, 7.

AGIC_VCPUIFHYPALIAS0_GICH_ELSR0_0
Offset: 0x5030

AGIC_VCPUIFHYPALIAS1_GICH_ELSR0_0
Offset: 0x5230

AGIC_VCPUIFHYPALIAS2_GICH_ELSR0_0
Offset: 0x5430

AGIC_VCPUIFHYPALIAS3_GICH_ELSR0_0
Offset: 0x5630

AGIC_VCPUIFHYPALIAS4_GICH_ELSR0_0
Offset: 0x5830

AGIC_VCPUIFHYPALIAS5_GICH_ELSR0_0
Offset: 0x5a30

AGIC_VCPUIFHYPALIAS6_GICH_ELSR0_0
Offset: 0x5c30

AGIC_VCPUIFHYPALIAS7_GICH_ELSR0_0
Offset: 0x5e30

Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICH_ELSR0**: Empty List register Status Register

AGIC_VCPUIFHYP_GICH_APR0_0

Offset: 0x40f0
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICH_APR0**: Active Priority Register

AGIC_VCPUIFHYPALIAS<*p*>_GICH_APR0_0,

where <*p*> = 0, 1, 2, 3, 4, 5, 6, 7.

AGIC_VCPUIFHYPALIAS0_GICH_APR0_0
Offset: 0x50f0

AGIC_VCPUIFHYPALIAS1_GICH_APR0_0
Offset: 0x52f0

AGIC_VCPUIFHYPALIAS2_GICH_APR0_0
Offset: 0x54f0

AGIC_VCPUIFHYPALIAS3_GICH_APR0_0
Offset: 0x56f0

AGIC_VCPUIFHYPALIAS4_GICH_APR0_0
Offset: 0x58f0

AGIC_VCPUIFHYPALIAS5_GICH_APR0_0
Offset: 0x5af0

AGIC_VCPUIFHYPALIAS6_GICH_APR0_0
Offset: 0x5cf0

AGIC_VCPUIFHYPALIAS7_GICH_APR0_0
Offset: 0x5ef0

Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICH_APR0**: Active Priority Register

AGIC_VCPUIFHYP_GICH_LR<*n*>_0,

where <*n*> = 0, 1, 2, 3.

AGIC_VCPUIFHYP_GICH_LR0_0
Offset: 0x4100

AGIC_VCPUIFHYP_GICH_LR1_0
Offset: 0x4104

AGIC_VCPUIFHYP_GICH_LR2_0
Offset: 0x4108

AGIC_VCPUIFHYP_GICH_LR3_0
Offset: 0x410c

Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICH_LR**: List Register

AGIC_VCPUIFHYPALIAS<*p*>_GICH_LR<*n*>_0,

where <*p*> = 0, 1, 2, 3, 4, 5, 6, 7 and

<*n*> = 0, 1, 2, 3.

AGIC_VCPUIFHYPALIAS0_GICH_LR0_0
Offset: 0x5100

AGIC_VCPUIFHYPALIAS0_GICH_LR1_0
Offset: 0x5104

AGIC_VCPUIFHYPALIAS0_GICH_LR2_0
Offset: 0x5108

AGIC_VCPUIFHYPALIAS0_GICH_LR3_0
Offset: 0x510c

AGIC_VCPUIFHYPALIAS1_GICH_LR0_0
Offset: 0x5300

AGIC_VCPUIFHYPALIAS1_GICH_LR1_0
Offset: 0x5304

AGIC_VCPUIFHYPALIAS1_GICH_LR2_0
Offset: 0x5308

AGIC_VCPUIFHYPALIAS1_GICH_LR3_0
Offset: 0x530c

AGIC_VCPUIFHYPALIAS2_GICH_LR0_0
Offset: 0x5500

AGIC_VCPUIFHYPALIAS2_GICH_LR1_0
Offset: 0x5504

AGIC_VCPUIFHYPALIAS2_GICH_LR2_0
Offset: 0x5508

AGIC_VCPUIFHYPALIAS2_GICH_LR3_0
Offset: 0x550c

AGIC_VCPUIFHYPALIAS3_GICH_LR0_0
Offset: 0x5700

AGIC_VCPUIFHYPALIAS3_GICH_LR1_0
Offset: 0x5704

AGIC_VCPUIFHYPALIAS3_GICH_LR2_0
Offset: 0x5708

AGIC_VCPUIFHYPALIAS3_GICH_LR3_0
Offset: 0x570c

AGIC_VCPUIFHYPALIAS4_GICH_LR0_0
Offset: 0x5900

AGIC_VCPUIFHYPALIAS4_GICH_LR1_0
Offset: 0x5904

AGIC_VCPUIFHYPALIAS4_GICH_LR2_0
Offset: 0x5908

AGIC_VCPUIFHYPALIAS4_GICH_LR3_0
Offset: 0x590c

AGIC_VCPUIFHYPALIAS5_GICH_LR0_0
Offset: 0x5b00

AGIC_VCPUIFHYPALIAS5_GICH_LR1_0
Offset: 0x5b04

AGIC_VCPUIFHYPALIAS5_GICH_LR2_0
Offset: 0x5b08

AGIC_VCPUIFHYPALIAS5_GICH_LR3_0
Offset: 0x5b0c

AGIC_VCPUIFHYPALIAS6_GICH_LR0_0
Offset: 0x5d00

AGIC_VCPUIFHYPALIAS6_GICH_LR1_0
Offset: 0x5d04

AGIC_VCPUIFHYPALIAS6_GICH_LR2_0
Offset: 0x5d08

AGIC_VCPUIFHYPALIAS6_GICH_LR3_0
Offset: 0x5d0c

AGIC_VCPUIFHYPALIAS7_GICH_LR0_0
Offset: 0x5f00

AGIC_VCPUIFHYPALIAS7_GICH_LR1_0
Offset: 0x5f04

AGIC_VCPUIFHYPALIAS7_GICH_LR2_0
Offset: 0x5f08

AGIC_VCPUIFHYPALIAS7_GICH_LR3_0
Offset: 0x5f0c

Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICH_LR**: List Register

AGIC_VCPUIFVM_GICV_CTLR_0

Offset: 0x6000
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICV_CTLR**: Virtual Machine Control Register

AGIC_VCPUIFVM_GICV_PMR_0

Offset: 0x6004
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICV_PMR**: VM Priority Mask Register

AGIC_VCPUIFVM_GICV_BPR_0

Offset: 0x6008
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICV_BPR**: VM Binary Point Register

AGIC_VCPUIFVM_GICV_IAR_0

Offset: 0x600c
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICV_IAR**: VM Interrupt Acknowledge Register

AGIC_VCPUIFVM_GICV_EOIR_0

Offset: 0x6010
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICV_EOIR**: VM End of Interrupt Register

AGIC_VCPUIFVM_GICV_RPR_0

Offset: 0x6014
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICV_RPR**: VM Running Priority Register

AGIC_VCPUIFVM_GICV_HPPIR_0

Offset: 0x6018
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICV_HPPIR**: VM Highest Priority Pending Interrupt Register

AGIC_VCPUIFVM_GICV_ABPR_0

Offset: 0x601c
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICV_ABPR**: VM Aliased Binary Point Register

AGIC_VCPUIFVM_GICV_AIAR_0

Offset: 0x6020
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICV_AIAR**: VM Aliased Interrupt Acknowledge Register

AGIC_VCPUIFVM_GICV_AEOIR_0

Offset: 0x6024
Read/Write: RW

Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICV_AEOIR**: VM Aliased End of Interrupt Register

AGIC_VCPUIFVM_GICV_AHPPIR_0

Offset: 0x6028
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICV_AHPPIR**: VM Aliased Highest Priority Pending Interrupt Register

AGIC_VCPUIFVM_GICV_APR0_0

Offset: 0x60d0
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICV_APR0**: VM Active Priority Register

AGIC_VCPUIFVM_GICV_IIDR_0

Offset: 0x60fc
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICV_IIDR**: VM CPU Interface Identification Register

AGIC_VCPUIFVM_GICV_DIR_0

Offset: 0x7000
Read/Write: RW
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GICV_DIR**: VM Deactivate Interrupt Register

7.8 Always On Digital Microphone (AODMIC)

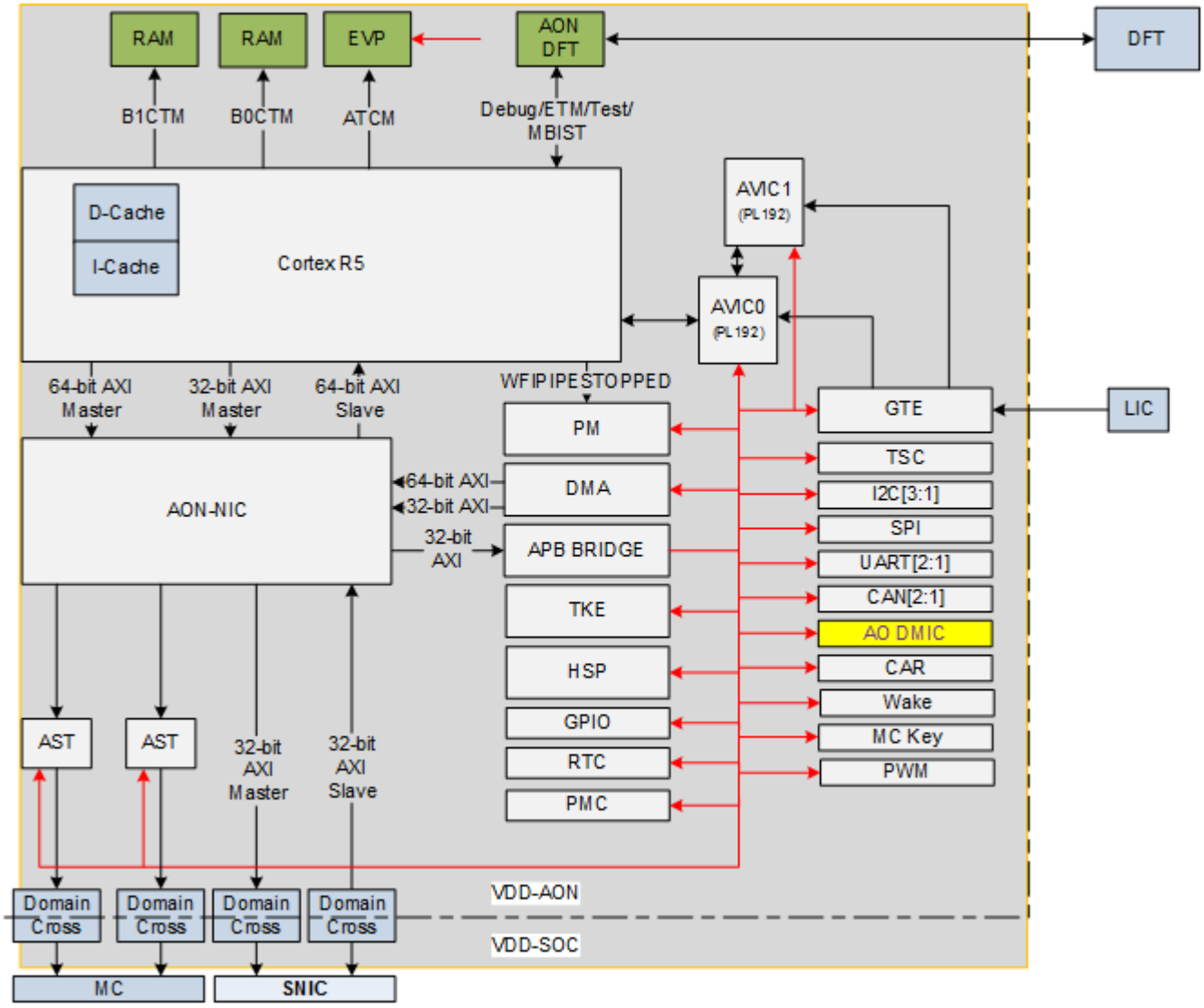
7.8.1 Overview

The Always-On Digital Microphone (AODMIC) resides in the AON cluster and processes input PDM data in low power modes. This allows the Sensor Processing Engine (SPE) to potentially be used for simple audio input processing while the SoC is in SC7.

The AODMIC converts PDM to PCM samples and stores them in the AODMIC FIFO. These samples are DMA transferred by GPC Direct Memory Access (DMA), and the SPE processes them to find if any voice energy is present. If voice energy is present, the SPE performs a key phrase match to determine if the system needs to wake. The accuracy of this match determines if further processing by the Audio Processing Engine (APE) is required before a full-system wake is required.

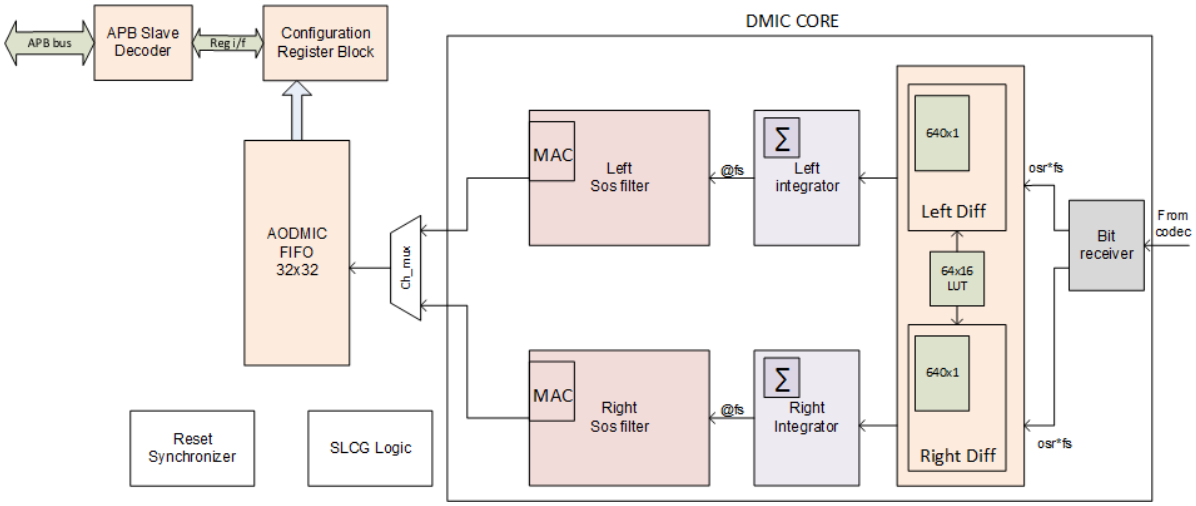
The figure below shows the block diagram of AON cluster and how AODMIC is connected.

Figure 7.50 AODMIC in AON Block Diagram



The diagram below shows the AODMIC module in detail.

Figure 7.51 AODMIC Top Level Block Diagram



7.8.1.1 Features

- Sample rate support: 8 kHz to 48 kHz
- Input PCM bit-width: 16 to 24 bits
- AODMIC supports only 64 OSR
- Filter gains are programmable; coefficients are hardcoded

7.8.2 Functional Description

7.8.2.1 Receive Buffer and Differentiator

The important characteristic of the receiver is the circular buffer used to store the incoming PDM bits. The circular buffer is populated with known data at every reset and power-on. The bit buffer should always be occupied with PDM bits and should be overwritten as and when next data is available.

The differentiator unit must read the bitbuffer at different locations, based on the over-sampling ratio (OSR), and form an index. The index is a pointer to the entries inside the SincNDifferentiation Table. The table is populated with possible combinations of results based on input PDM bit density. The input to the differentiator unit is a single bit data and its output is 16 bits data with the MSB as the sign bit. This operation can be completed in one DMIC clock cycle.

7.8.2.2 Integrator

The integrator unit is an accumulator unit where the integrator state buffer continues to get updated throughout the AODMIC receiver operation. The integrator unit is implemented using adders and the state needs to be reset only during a reset or power-on. The intermediate state buffer is a five deep 64-bit wide buffer to store the intermediate operands. The input to this unit is a 16 bit data from the differentiator unit. Its output is a 48-bit signed integer. With parallel adder implementation of the integrator unit and using 10 adders for the sinc decimation order of 5, this operation can be completed in one DMIC clock cycle.

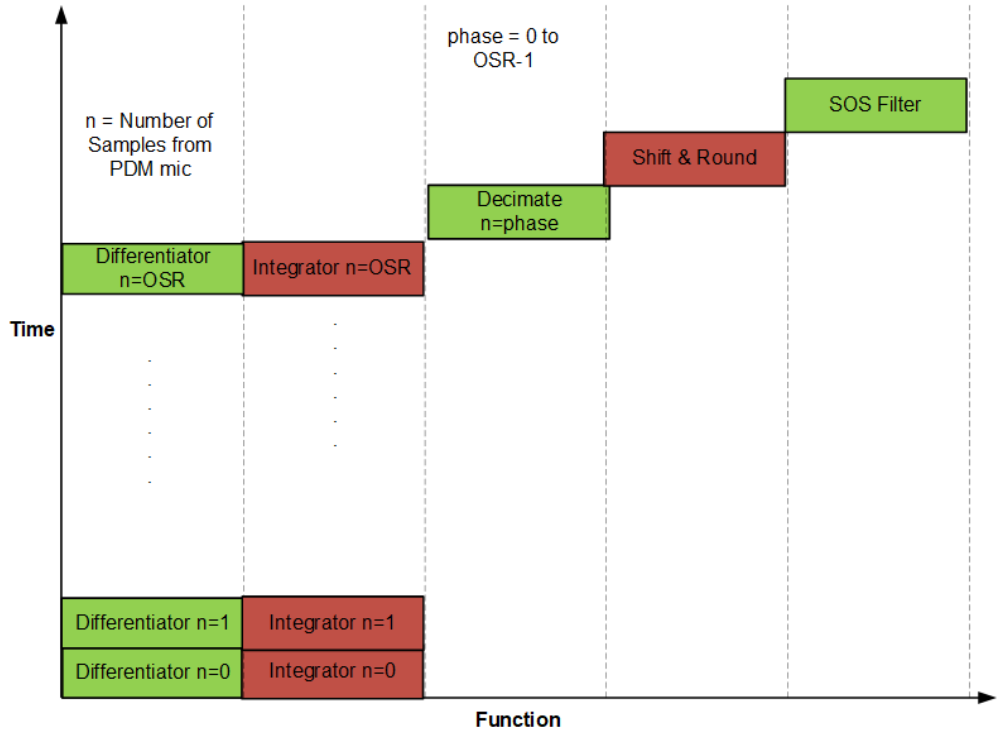
7.8.2.3 Downsampler

This unit selects one sample out of every OSR number of samples of integration output depending on the programming of the phase parameter. The input to this unit is the integrator's output array up to the OSR for each 48-bit signed integer. The output to this unit is one sample which is also 48-bit signed integer.

7.8.2.3.1 Shift and Round

This unit is responsible for performing the shift and round operation on the decimated output produced. The amount of shift depends on the parameters OSR, Sinc decimation order and output width of the output sample required. The output produced from the shift and round unit is 32-bit.

Figure 7.52 Data Flow Diagram



7.8.2.4 Physical Structure

7.8.2.4.1 Clocking

The AODMIC works on two clocks:

1. APB clock with which all registers, APB bus, and AODMIC FIFO reading logic is clocked.
2. DMIC I/O clock with which the entire AODMIC core logic is clocked.

7.8.2.4.2 Data Format

The AODMIC core output is fixed to the 24-bit format. The 24-bit output of AODMIC is present in the least significant bit (LSB) of 32 bits.

The AODMIC FIFO stores the data as is and does not convert the data. Therefore, 23:0 bits is picked by software while capturing. If software is capturing in 32-bit mode, then software should take the 23:0 bits and left shift by 8 bits to form 32 bits. In the APE DMIC this, is handled by CIF (internal hardware module) which does the required conversion. But in AODMIC, the data is given directly without any conversion.

7.8.3 Programming Guidelines

7.8.3.1 Normal Operation (Enabling)

1. Program the CTRL_CHANNEL_SELECT field based on mono/stereo.
2. Program the AODMIC_APB_FIFO_CTRL_THRESHOLD value.
3. Program global enable by writing into AODMIC_ENABLE.

7.8.3.2 Disabling

1. Disable AODMIC_ENABLE by writing 0.
2. Check the AODMIC_APB_FIFO_STATUS_DATA_COUNT; if non-zero. then program DMA to read the data from FIFO and empty it.
3. Check the AODMIC_STATUS_ENABLE_STATUS for 0. If AODMIC_STATUS_ENABLE_STATUS is still '1' (due to pipe line data in internal controller) then repeat step 2 and step 3 *this step).

7.8.3.3 Optional

Filter gains, AODMIC_DCR_FILTER_GAIN_0, AODMIC_LP_FILTER_GAIN_0, AODMIC_CORRECTION_FILTER_GAIN_0 can be programmed for volume control. The coefficients are in the signed format with 23 LSB representing fractional part.

7.8.4 AODMIC Registers

Refer to "Reading Register Tables" in the Introduction chapter for the register table protocol as well as recommendations for accessing registers.

AODMIC_APB_FIFO_CTRL_0
Offset: 0x0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00001000 (0bxxxx,xxxx,xxxx,xxxx,xx01,0000,xxxx,xxxx)

Bit	Reset	Description
13:8	0x10	**THRESHOLD**: When FIFO data count reach this value, AODMIC raises interrupt.

AODMIC_APB_FIFO_CTRL_RD_DATA_0

Offset: 0x4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

AODMIC_APB_FIFO_STATUS_0

Offset: 0x8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000040 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0100,0000)

Bit	Reset	Description
8	FALSE	**RX_REQ**: Indicate if FIFO has data count equal to programmed threshold. **0** = FALSE **1** = TRUE
7	FALSE	**FULL**: **0** = FALSE **1** = TRUE
6	TRUE	**EMPTY**: **0** = FALSE **1** = TRUE
5:0	0x0	**DATA_COUNT**

AODMIC_APB_RX_INT_STATUS_0

Offset: 0xc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,xxxx,xxx0)

Bit	Reset	Description
8	CLEAR	**RX_OVERRUN**: **0** = CLEAR **1** = SET
0	CLEAR	**RX_DONE**: **0** = CLEAR **1** = SET

AODMIC_APB_RX_INT_MASK_0

Offset: 0x10
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000101 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx1,xxxx,xxx1)

Bit	Reset	Description
8	MASK	**RX_OVERRUN**: **0** = UNMASK **1** = MASK
0	MASK	**RX_DONE**: **0** = UNMASK **1** = MASK

AODMIC_APB_RX_INT_SET_0

Offset: 0x14
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,xxxx,xxx0)

Bit	Reset	Description
8	CLEAR	**RX_OVERRUN**: When the Software sets these fields, corresponding interrupts are generated. **0** = CLEAR **1** = SET
0	CLEAR	**RX_DONE**: When the Software sets these fields, corresponding interrupts are generated. **0** = CLEAR **1** = SET

AODMIC_APB_RX_INT_CLEAR_0

Offset: 0x18
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,xxxx,xxx0)

Bit	Reset	Description
8	CLEAR	**RX_OVERRUN**: **0** = CLEAR **1** = SET
0	CLEAR	**RX_DONE**: **0** = CLEAR **1** = SET

AODMIC_ENABLE_0

DMIC Common Registers

Offset: 0x40
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**ENABLE**: **0** = FALSE **1** = TRUE

AODMIC_SOFT_RESET_0

Offset: 0x44
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	FALSE	**SOFT_RESET**: **0** = FALSE **1** = TRUE

AODMIC_CG_0

Offset: 0x48
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	TRUE	**SLCG_EN**: Second level clock gating enable, for first two channels and global/ common logic. **0** = FALSE **1** = TRUE

AODMIC_STATUS_0

Offset: 0x4c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000010 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxx0,xx01,xxx0)

Bit	Reset	Description
31	FALSE	**CNFG_ERR**: **0** = FALSE **1** = TRUE
8	FALSE	**SLCG_CLKEN**: **0** = FALSE **1** = TRUE
5	FALSE	**FIFO_FULL**: **0** = FALSE **1** = TRUE
4	TRUE	**FIFO_EMPTY**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	FALSE	**ENABLE_STATUS**: **0** = FALSE **1** = TRUE

AODMIC_INT_STATUS_0

Offset: 0x50
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,xxxx,xxx0)

Bit	Reset	Description
8	CLEAR	**RX_OVERRUN**: **0** = CLEAR **1** = SET
0	CLEAR	**RX_DONE**: **0** = CLEAR **1** = SET

AODMIC_CTRL_0

Offset: 0x64
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000300 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx11,xxx0,xx00)

Bit	Reset	Description
9:8	STEREO	**CHANNEL_SELECT**: **0** = NONE **1** = LEFT **2** = RIGHT **3** = STEREO
4	LEFT	**LRSEL_POLARITY**: **0** = LEFT **1** = RIGHT
1:0	OSR64	**OSR**: **0** = OSR64 **1** = OSR128 **2** = OSR256

AODMIC_DBG_CTRL_0

Offset: 0x70
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000002 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0010)

Bit	Reset	Description
3	DISABLE	**DCR_ENABLE**: **0** = DISABLE **1** = ENABLE
2	DISABLE	**LP_ENABLE**: **0** = DISABLE **1** = ENABLE
1	ENABLE	**SC_ENABLE**: **0** = DISABLE **1** = ENABLE
0	DISABLE	**BYPASS**: **0** = DISABLE **1** = ENABLE

AODMIC_DCR_FILTER_GAIN_0

All Gains and Coefficients are in Q23 format.

Offset: 0x74
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00800000 (0b0000,0000,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x800000	**GAIN_0**

AODMIC_DCR_BIQUAD_0_COEF_0_0

Offset: 0x78
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00800000 (0b0000,0000,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x800000	**COEF_0**

AODMIC_DCR_BIQUAD_0_COEF_1_0

Offset: 0x7c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xff800000 (0b1111,1111,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0xff800000	**COEF_1**

AODMIC_DCR_BIQUAD_0_COEF_2_0

Offset: 0x80
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**COEF_2**

AODMIC_DCR_BIQUAD_0_COEF_3_0

Offset: 0x84
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xff800347 (0b1111,1111,1000,0000,0000,0011,0100,0111)

Bit	Reset	Description
31:0	0xff800347	**COEF_3**

AODMIC_DCR_BIQUAD_0_COEF_4_0

Offset: 0x88
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**COEF_4**

AODMIC_LP_FILTER_GAIN_0

Offset: 0x8c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x004c255a (0b0000,0000,0100,1100,0010,0101,0101,1010)

Bit	Reset	Description
31:0	0x4c255a	**GAIN_0**

AODMIC_LP_BIQUAD_0_COEF_0_0

Offset: 0x90
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00800000 (0b0000,0000,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x800000	**COEF_0**

AODMIC_LP_BIQUAD_0_COEF_1_0

Offset: 0x94
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00ffa74b (0b0000,0000,1111,1111,1010,0111,0100,1011)

Bit	Reset	Description
31:0	0xffa74b	**COEF_1**

AODMIC_LP_BIQUAD_0_COEF_2_0
Offset: 0x98
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00800000 (0b0000,0000,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x800000	**COEF_2**

AODMIC_LP_BIQUAD_0_COEF_3_0
Offset: 0x9c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x009e382a (0b0000,0000,1001,1110,0011,1000,0010,1010)

Bit	Reset	Description
31:0	0x9e382a	**COEF_3**

AODMIC_LP_BIQUAD_0_COEF_4_0
Offset: 0xa0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00380f38 (0b0000,0000,0011,1000,0000,1111,0011,1000)

Bit	Reset	Description
31:0	0x380f38	**COEF_4**

AODMIC_LP_BIQUAD_1_COEF_0_0

Offset: 0xa4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00800000 (0b0000,0000,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x800000	**COEF_0**

AODMIC_LP_BIQUAD_1_COEF_1_0

Offset: 0xa8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00fe1178 (0b0000,0000,1111,1110,0001,0001,0111,1000)

Bit	Reset	Description
31:0	0xfe1178	**COEF_1**

AODMIC_LP_BIQUAD_1_COEF_2_0

Offset: 0xac
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00800000 (0b0000,0000,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x800000	**COEF_2**

AODMIC_LP_BIQUAD_1_COEF_3_0

Offset: 0xb0
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00e05f02 (0b0000,0000,1110,0000,0101,1111,0000,0010)

Bit	Reset	Description
31:0	0xe05f02	**COEF_3**

AODMIC_LP_BIQUAD_1_COEF_4_0
Offset: 0xb4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x006fc80d (0b0000,0000,0110,1111,1100,1000,0000,1101)

Bit	Reset	Description
31:0	0x6fc80d	**COEF_4**

AODMIC_CORRECTION_FILTER_GAIN_0
Offset: 0xb8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x010628f6 (0b0000,0001,0000,0110,0010,1000,1111,0110)

Bit	Reset	Description
31:0	0x10628f6	**GAIN_0**

AODMIC_CORRECTION_BIQUAD_0_COEF_0_0
Offset: 0xbc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00800000 (0b0000,0000,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x800000	**COEF_0**

AODMIC_CORRECTION_BIQUAD_0_COEF_1_0

Offset: 0xc0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**COEF_1**

AODMIC_CORRECTION_BIQUAD_0_COEF_2_0

Offset: 0xc4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**COEF_2**

AODMIC_CORRECTION_BIQUAD_0_COEF_3_0

Offset: 0xc8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0067ffff (0b0000,0000,0110,0111,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0x67ffff	**COEF_3**

AODMIC_CORRECTION_BIQUAD_0_COEF_4_0

Offset: 0xcc
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**COEF_4**

AODMIC_CORRECTION_BIQUAD_1_COEF_0_0

Offset: 0xd0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00800000 (0b0000,0000,1000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x800000	**COEF_0**

AODMIC_CORRECTION_BIQUAD_1_COEF_1_0

Offset: 0xd4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0048f5c2 (0b0000,0000,0100,1000,1111,0101,1100,0010)

Bit	Reset	Description
31:0	0x48f5c2	**COEF_1**

AODMIC_CORRECTION_BIQUAD_1_COEF_2_0

Offset: 0xd8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**COEF_2**

AODMIC_CORRECTION_BIQUAD_1_COEF_3_0
Offset: 0xdc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00562394 (0b0000,0000,0101,0110,0010,0011,1001,0100)

Bit	Reset	Description
31:0	0x562394	**COEF_3**

AODMIC_CORRECTION_BIQUAD_1_COEF_4_0
Offset: 0xe0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00169446 (0b0000,0000,0001,0110,1001,0100,0100,0110)

Bit	Reset	Description
31:0	0x169446	**COEF_4**

7.9 Pixel Memory Formats

7.9.1 Functional Description

The following outlines the memory tiling formats and the pixel formats that may be produced and consumed by interacting engines in this System-on-Chip (SoC).

7.9.1.1 Tiling Formats

This SoC supports the memory formats described in the following section. MSS clients can support block linear formats and pitch-linear formats. Refer to the description of each device for more details on the modes supported.

- Pitch linear arranges pixels in raster array order, and is useful for clients that interact directly or indirectly with the CPU or that make horizontal (or linear) accesses to memory.

- Block linear is a tiled format optimized for clients that access regions of pixels across more than one line.

Table 7.95 Client Shared Tiling Formats

Format	Status	Used for	Description
Pitch Linear	Supported by some clients	Software-compatibility, clients that linearly access memory	Simple array ordering
Block linear 16Bx2 with TEGRA RAW or XBAR RAW sector ordering	Supported by all clients	GPU color surfaces, clients that access memory in blocks or vertically	Square DRAM page arrangement (blocks) with folded atoms

The tiling format describes the organization of bytes within a surface. The tiling formats convert an (x, y) offset of a sample in a surface to a linear address. To do the translation, they require additional information about the surface.

Parameter	Used by	Description	Values	Minimum Alignment[1]
Layout	All	Surface layout	PITCH, BLOCKLINEAR	Not Applicable
Base Address	Pitch	Starting address of surface, in bytes	40-bit virtual address -or- 40-bit physical address	Not Applicable 64 B (general) 128 B (GPU) 256 B (NVENC, OFA, VIC) 512 B (display)
Pitch	Pitch	Line-to-line stride in bytes	Large enough for at least 4,096 pixels in bytes (some clients may support larger values)	64 B (general) 128 B (GPU, display) 256 B (NVENC, OFA, display cursor) up to 256 B (VIC)
Base Address	Block Linear	Starting address of surface, in bytes	40-bit virtual address -or- 40-bit physical address	512 B (general) 1 KB (display)
Block Width	Block Linear	Width of block in GOBs	ONE_GOB (this is the only setting supported)	Not applicable
Block Height	Block Linear	Height of block in GOBs	[ONE_GOB, THIRTYTWO_GOBS] (SIXTEEN_GOBS is the normal setting, but TWO_GOBS may be needed for display surfaces with rotation disabled.	Not applicable
Block Depth	Block Linear	Depth of block in GOBs	[ONE_GOB, THIRTYTWO_GOBS] (GPU supports values greater than ONE_GOB)	Not applicable
Width	Block Linear	Width of surface in samples	[1, 4096] samples (some clients may support larger values)	One sample

1. These are minimum alignments. Larger alignments are sometimes needed for performance. See the recommendations for software in the Software Guidelines section.

2. The pitch alignment restrictions for VIC depend on its cache line shape: 256 Bx1: 256 B, 128Bx2: 128 B, otherwise: 64 B.

The NVIDIA Ampere Architecture 3D class supported by the GPU has a full list of surface parameter ranges that, in many cases, exceed those in this table. For GPU surfaces that are larger than a SoC client can support, all manipulation of the surface must be done on the GPU. Displayable surfaces are limited to the maximum 8K image width supported by display.

7.9.1.2 DRAM Organization

Although the tiling equations produce byte addresses, the MC interfaces operate in terms of atoms, which are the smallest possible unit of addressable DRAM memory. This SoC uses 32-byte atom with performance benefits when using request size of 64 bytes or larger. This SoC has a multichannel memory system with a channel interleave.

The SoC has a multichannel memory system with a channel interleave of 512 B or 1 KB. The interleave is global memory system configuration. Memory locations within one of these aligned 512 B or 1 KB address blocks are stored in the same DRAM page in the same DRAM channel. In order to achieve frequent utilization of the DRAMs, clients must issue enough requests within a DRAM page to hide precharges and activates for pages in other DRAM banks. This means that a given client must issue ~128 B within a given 512 B or 1 KB page before moving to another. The purpose of most of the tiling formats is to maximize the number of requests in one page to increase DRAM utilization.

In physically contiguous memory (carveout), or with 64 KB pages, channel, and bank swizzling equations ensure that channels and banks are evenly interleaved, so work is distributed over channels and bank conflicts are rare. With 4 KB dynamically allocated pages, a given 4 KB page only contains data from four channels. To prevent ruinous channel camping (worst-case 1/16 DRAM bandwidth), memory allocation must use page coloring. In page coloring, each 4 KB physical page is assigned a “color” based on the channels and banks it contains. For example, a “red” page might have channels 0-3 and bank 0, while a “green” page might have channels 4-7 and bank 5.

The page table mapping should map physical pages of the appropriate color to virtual pages to interleave channels and banks optimally. If page coloring is not feasible, clients should take advantage of the fact that there are no channel conflicts within a single 4 KB virtual memory page. However, be aware that adjacent pages could reference the same four pages, potential-limiting bandwidth to 1/4 of the theoretical bandwidth.

- Certain clients can take advantage of the memory system’s subpartition feature to read or write noncontiguous pairs of 32B subpackets, rather than being restricted to a 64B DRAM atom. See the Sub-Partitions section for details.

7.9.1.3 Block Linear

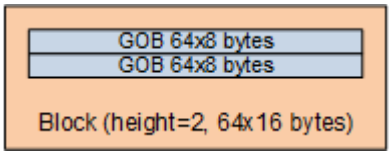
The block linear formats map batches of contiguous addresses into rectangular blocks, and tile these blocks linearly. Blocking serves two purposes:

1. It maps a DRAM page into a rectangular region, rather than into a one-pixel-tall strip, to capture locality in a graphics geometry stream or other 2D-oriented client, such as an MPEG encoder or decoder.
2. It is possible to interleave channels and banks in x and y in a predictable way, minimizing channel camping and bank conflicts over a large region.

Blocks themselves are arranged from GOBs, which are Groups of Bytes. An NVIDIA Ampere Architecture GOB is 512 bytes arranged as 64x8 bytes. GOBs are stacked vertically to form a block. An additional surface parameter called the block height controls the number of GOBs stacked vertically in a block.

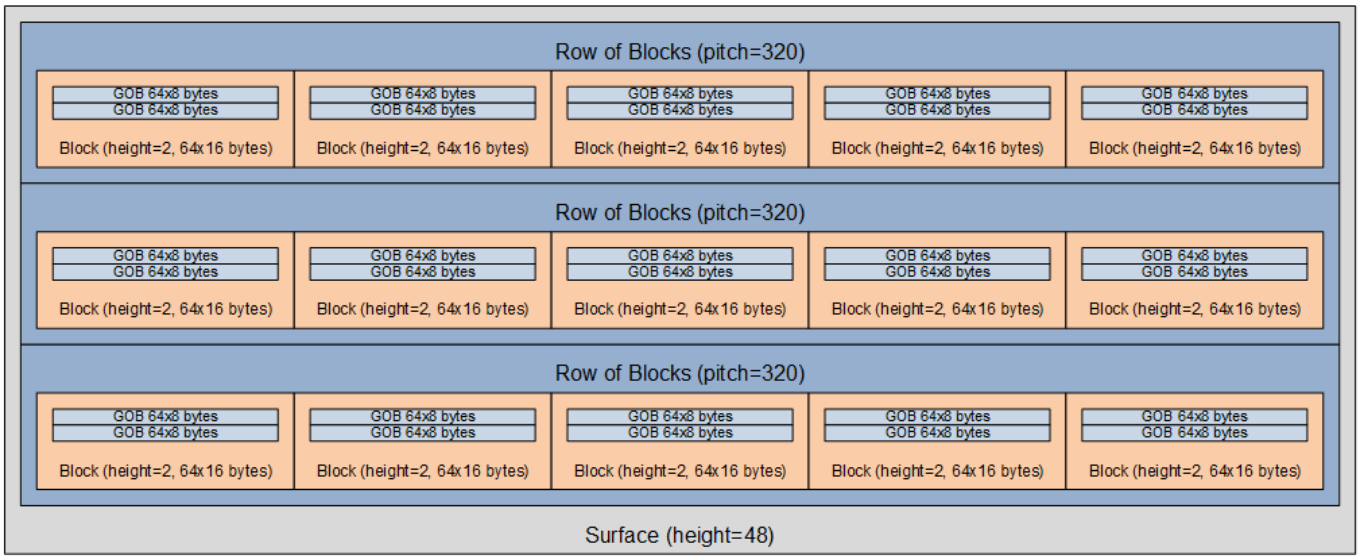
The recommended block height for most buffers is 16 GOBs (128 lines). This supports 8x8 bank interleaving. For buffers, which linear display access is more important than access by GPU, VIC, or other block-oriented engine, block height may be set to one GOB, providing more locality for the display client.

Figure 7.53 Example Block with block_height = 2



Blocks are arranged horizontally into a Row of Blocks (ROBs) to fill out the surface to the width value (or slightly beyond, if the width is not a multiple of the 64-Byte block width). Upper address bit swizzling in the bank swizzling algorithm generally makes it unnecessary to pad the surface width further.

Figure 7.54 Block Linear Surfaces



The tiling equations for block linear start out by computing the GOB address as follows:

GOB_address = base_address +

(y / (8 * block_height)) * 512 * block_height * image_width_in_gobs +

(x * bytes_per_pixel / 64) * 512 * block_height +

(y % (8 * block_height) / 8) * 512

Where:

image_width_in_gobs = ceiling(image_width * bytes_per_pixel / 64)

The individual pixel byte address is then determined from the GOB address based on the sector packing and sector ordering.

> The image width determines Image_width_in_gobs. There is no additional ‘pitch’ parameter, as in legacy tiling.

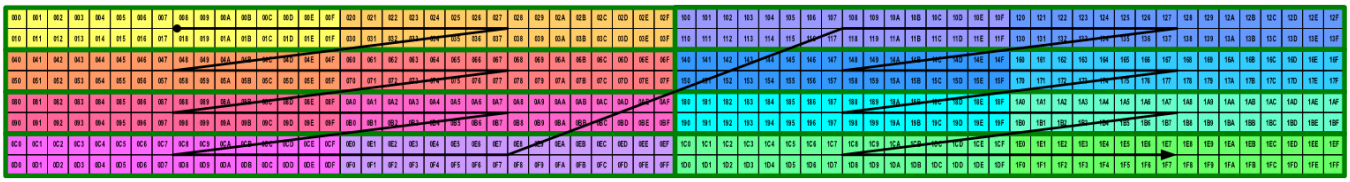
7.9.1.3.1 Sector Packing

Sector packing is the arrangement of bytes within a 32-Byte sector. Sector ordering is the arrangement of sectors within a GOB. All buffers that are shared between the clients use 16 Byte × 2 sector packing: the 32 Byte within a sector are arranged as two 16B lines on top of each other. (The GPU supports KINDs with other packing, but these are private to the GPU.)

The following diagram represents a view of a TEGRA RAW block-linear GOB in pixel space. Each numbered box represents a byte, and each number represents the address offset of that byte from the base address of the GOB. The green outline denotes the set of sectors that make up a 64B memory access.

As an example of how to interpret the following diagram, assuming 4-bytes per pixel, the first eight pixels in row 0 within the GOB are stored in bytes 00-0F and 20-2F. That is, the first, second, third, fourth pixels are stored in bytes 00-03, 04-07, 08-0B, 0C-0F respectively, and fifth, sixth, seventh, eighth pixels are stored in bytes 20-23, 24-27, 28-2B, 2C-2F respectively in a 64 byte contiguous memory access. The ninth pixel in row 0 within the GOB is stored in bytes 100-103 requiring a separate memory access.

Figure 7.55 Block Linear (GPU 16Bx2 kind with TEGRA_RAW Sector Ordering).



The following equations are used for address calculation using SoC and XBAR RAW sector ordering:

TEGRA RAW_Address = GOB_address + ((x*bytes_per_pixel%64)/32)*256 + ((y%8)/2)*64 + ((x*bytes_per_pixel%32)/16)*32 + (y%2)*16 + (x*bytes_per_pixel%16)

XBAR RAW_Address = GOB_address + ((x*bytes_per_pixel%64)/32)*256 + ((y%8)/4)*128 + ((x*bytes_per_pixel%32)/16)*64 + (y%4)*16 + (x*bytes_per_pixel%16)

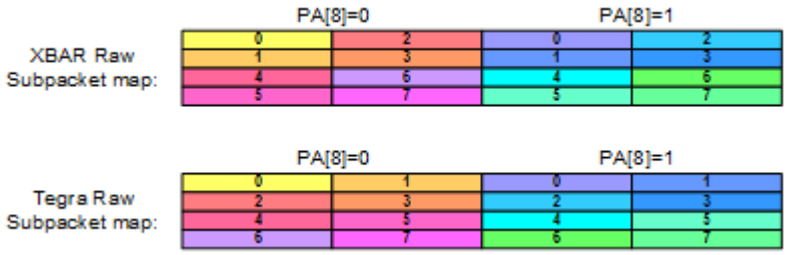
7.9.1.3.2 Sector Ordering

The sector ordering in this format reduces the amount of buffering required for line storage by ISP. It also reduces the amount of instantaneous bandwidth required by ISP because of the reduction in the number of lines that are written to memory.

Support for block linear sector ordering shown in the following figure is as follows:

- Display supports XBAR_RAW sector ordering.
- All other MSS clients supporting block linear format will use TEGRA RAW sector ordering.
- Memory-client address bit 39 will be used to indicate if order swizzling is required (1=Swizzle XBAR RAW <-> TEGRA RAW sector order, 0=data passes through unswizzled).
- GPU upper PTE bit will be used to indicate if order swizzling is required (1=Swizzle XBAR RAW <-> TEGRA RAW sector order, 0=data passes through unswizzled).

Figure 7.56 GOB Subpacket Mapping



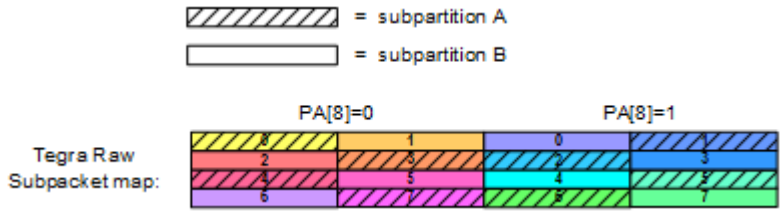
Clients that make more vertically oriented accesses can take advantage of the subpartition feature described in the next section.

7.9.1.4 Sub-Partitions

The 64-bit DRAM channel can be divided into two subpartitions with DQ[31:0] forming subpartition A and DQ[63:32] forming subpartition B.

Each 64-Byte atom is divided into two sectors of 32 Bytes. One sector contains bytes 0 – 31 while the other contains bytes 32 – 63. The sectors are mapped to subpartitions by the MCCIF using PA[8]^PA[6]^PA[5]. This produces a checkerboard mapping of the physical address space to subpartitions, which permits both subpartitions to be used whether the primitive is oriented horizontally or vertically. The checkerboard is mirrored between tiles in a GOB, which is important when the surface is compressed.

Figure 7.57 Address to Sub-Partition Mapping



Clients that use the subpartition feature can access two noncontiguous sectors targeting the same GOB in a single request provided they do not target the same subpartition. A common use for this feature is to change the shape of a 64-Byte request from 32 Byte × 2 (without subpartitions) to 16 Byte × 4 (using subpartitions). This is the preferred fetch footprint for the GPU texture cache and is used by display when scanning out vertically. Clients that do not want to use the subpartition feature set both copies of PA[8:5] to the same value.

7.9.1.5 Hardware Support

Clients implement registers in their module to control the parameters (base address, kind, width or pitch, and block height) required for tiling control of each surface supported by the client. For planar and semiplanar surfaces, in which a single image is composed of multiple buffers or planes, independent parameters should be provided for each buffer or plane.

7.9.1.5.1 Pixel Storage Registers

Clients should make memory fetches that are integral multiples of DRAM atoms to avoid wasting bandwidth. If fetches can extend beyond the active buffer size (in width or height), the buffer must be padded to accommodate the maximum-sized fetch. For pitch-linear surfaces, this may mean padding the pitch and padding the number of lines in the buffer. For block linear surfaces (in rare instances), this may mean adding an additional row of blocks at the bottom of the buffer.

> For block linear surfaces, there is no pitch parameter separate from surface width. To avoid making data fetches outside a buffer's allocated width, units should generally not make requests that are wider than 64B (the block width).

7.9.1.6 Pixel Format Naming

Pixel format naming conventions apply to color surfaces for camera, gaming, and video applications. Color formats listed in this section cover both SoC and dGPU color surfaces.

7.9.1.6.1 Naming Conventions

Pixel format naming conventions continue to use the naming style adopted in previous chips. They are based on the following set of rules:

1. All format names begin T_ (T followed by underscore) to distinguish them from other format name conventions.
2. If the number of bits per pixel is not a power of 2, use an Array of Words name. Example: T_R8_G8_B8.
   a. Write all the components in increasing address order from left to right with an underscore between each component.
   b. The name lists the bits in this order. A set of attributes describes each component:
      i. a Component letter (refer to the Pixel Naming Component Letters table).
      ii. a decimal number (the number of bits in the component–always a multiple of 8)
      iii. an optional Type Suffix (refer to the Pixel Naming Type Suffixes table)
3. Otherwise, if the number of bits per pixel is a power of 2, use a Single Word name. Example: T_A8B8G8R8, T_R5G6B5.

a. Collect all the components in the pixel into a single word using little endian order.
b. Arrange the components from left to right in MSB to LSB order.
c. The name lists the bits in this order. A set of attributes describes each component:
   i. a Component letter (refer to the Pixel Naming Component Letters table).
   ii. a decimal number (the number of bits in the component)
   iii. an optional type suffix (refer to the Pixel Naming Type Suffixes table)

4. Identify the type:
   a. Identify the type of each component, ignoring any S and X components. Applicable types are listed in the Pixel Naming Type Suffixes table.
   b. If all the (non-ignored) types are Unsigned Normalized, then, do not use any type suffix. Example: T_A8B8G8R8.
   c. Else if all the (non-ignored) types use the same type (not Unsigned Normalized), then add to the end of the format name an underscore and a type suffix. This suffix applies to all the components (except S and X components). Example: T_A8B8G8R8_I.
   d. Else there are different types. Add a type suffix (without an underscore) to each component identifier (following the number of bits). Do not add a type identifier to any S or X or Unsigned Normalized components. Do not add an underscore or type identifier to the end of the format name. Example: T_A2B10FG10FR10F.
5. Special cases:
   a. T_E5B9G9R9_F - each RGB component is a mantissa and E is a shared exponent, which is applied to each component. The result is three floating point components (RGB).
   b. Packed YUV formats like T_V8_Y8__U8_Y8 and T_B8_G8__R8_G8, which have a repeated component: This format represents two pixels. The double underscore is the separator between the two pixels. The repeated component (Y/G) is unique to each pixel. Both pixels share the other components (U/V/R/B). Use Array of Words naming for these (see Step 2)).
   c. Bayer formats (RAW10, RAW12, RAW16, RAW20).
   d. Compressed formats (ETC, DXT, LATC, EAC).
   e. Append the “N” delimiter to indicate chroma subsampling (that is, N444, N422, N422R, N420). For example, T_U8V8_N420 indicates a U8V8 plane with 4:2:0 chroma subsampling).
6. Multi-Pixel Packing Formats: If more than one pixel is described by a single word, which is a power-of-two bits, then use a word-oriented name. Example: T_X2Lc10Lb10La10.
   a. Collect all the components in the pixel into a single word using little endian order.
   b. Arrange the components from left to right in MSB to LSB order.
   c. The name lists the bits in this order:
      i. a Component letter (refer to the Pixel Naming Component Letters table).
      ii. a lowercase pixel-ordering identifier “a”, “b”, “c”, and so on, where “a” is the “first” logic pixel in the group, “b” is the second, “c” is the third, and so on. For components that are shared by all pixels in the group, or X components (that are not part of any pixel), the pixel-ordering identifier is omitted. For components that are shared by more than one, but not all, pixels in the group, multiple pixel order-identifying letters are to be used.
      iii. a decimal number (the number of bits in the component)

iv. an optional type suffix (refer to the Pixel Naming Type Suffixes table).

7. Multiple Plane Formats:
   a. Some units may use a different color format name for each plane. In that case, the color format names are as described above (for example, T_Y8, T_U8, T_V8, T_V8U8). Other units may use a single color format name describing all planes in a single name. In this case, the name has one or more triple underscores. The triple underscore separates one plane from the next.
      i. Example: T_Y8___V8U8_N420: Indicates a semiplanar YUV with T_Y8 in the first plane, and T_V8U8 in the second plane, with 4:2:0 chroma subsampling.
      ii. Example: T_Y8___U8___V8_N420: Indicates a fully planar YUV with T_Y8 in the first plane, T_U8 in the second plane, and T_V8 in the third plane, with 4:2:0 chroma subsampling.

MSS clients are required to support YCbCr Rec.601 formats (unsigned 8-bit values). But for brevity, Cb and Cr components are represented with U and V, respectively (for example, AYUV represents a format containing A, Y, Cb, Cr components). This does not imply that “YUV” formats use Y, U, V components (signed 8-bit values with a 128 offset).

Table 7.96 Pixel Naming Component Letters

Component Letter	Description
R	Red
G	Green
B	Blue
A	Alpha
L	Luma (copy to R, G, and B. Copy from R.)
Y	Luminance for YCbCr601 (YUV)
U	Cb for YCbCr601 (YUV)
V	Cr for YCbCr601 (YUV)
N	Delimiter for indicating chroma subsampling (N444, N422, N422R, N420)
P	Palette index
C	Coverage (or other special use)
S	Stencil (always Unsigned Integer)
Z	Depth value
E	Shared exponent. When this component is present, all the other components are mantissas to which this exponent is applied.
X	Unused (ignored on read)

Table 7.97 Pixel Naming Type Suffixes

Type Suffix	Description
<none>	Unsigned Normalized
SN	Signed Normalized
I	Signed Integer
UI	Unsigned Integer
NL	Nonlinear Z Format
F	Float
LN	Unsigned Normalized sRGB colorspace (L=log)
MSAA	Multisample (special case–only used with S8 (T_S8_MSAA)).
N444	No subsampling of chroma (chroma and luma are full resolution).
N422	Subsampled chroma in 422 (half res horizontal, full res vertical)
N422R	Subsampled chroma in 420R (full res horizontal, half res vertical)
N420	Subsampled chroma in 420 (half res horizontal, half res vertical)
TRUE	Uses true YUV (not YCbCr)

7.9.1.6.2 Pixel Format Bit Patterns

The following tables describe the bit patterns associated with the pixel formats supported. The maximum pixel size supported by the SoC's engines is 128 bits.

7.9.1.6.3 Array of Words Pixel Formats

Array of Words type pixel formats lists each component in increasing address order from left to right.

Figure 7.58 T_B8_G8_R8

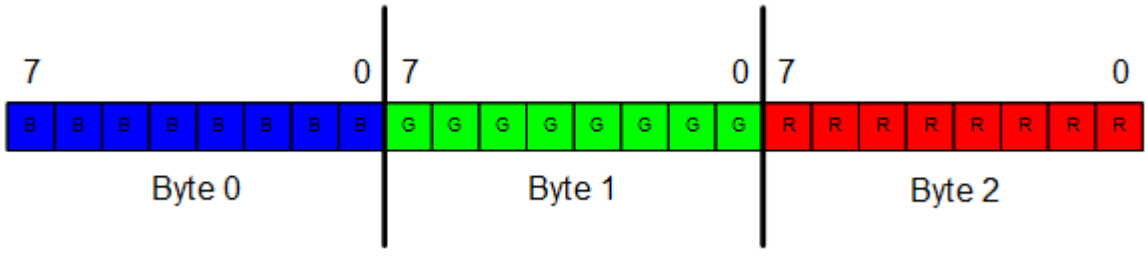


Figure 7.59 T_R16_G16_B16_I

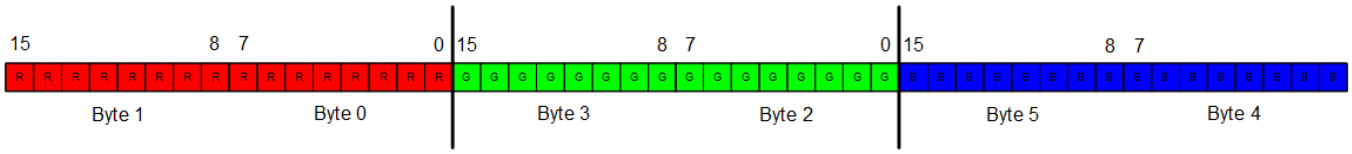


Table 7.98 Array of Words Pixel Formats

Hardware Name	Bit pattern (Left to Right: LS-Byte to MS-Byte, ms-bit to ls-bit per Byte)	Software Name
T_R8_G8_B8	RRRRRRRR GGGGGGGG BBBBBBBB	NvColorFormat_R8_G8_B8
T_B8_G8_R8	BBBBBBBB GGGGGGGG RRRRRRRR	NvColorFormat_B8_G8_R8
T_R8_G8_B8_SN	RRRRRRRR GGGGGGGG BBBBBBBB	NvColorFormat_SNorm_R8_G8_B8
T_R8_G8_B8_UI	RRRRRRRR GGGGGGGG BBBBBBBB	NvColorFormat_Uint_R8_G8_B8
T_R8_G8_B8_I	RRRRRRRR GGGGGGGG BBBBBBBB	NvColorFormat_Int_R8_G8_B8
T_Y8_U8_V8	YYYYYYYY UUUUUUUU VVVVVVVV	NvColorFormat_Y8_U8_V8
T_Y8_V8_U8	YYYYYYYY VVVVVVVV UUUUUUUU	NvColorFormat_Y8_V8_U8
T_U8_Y8_V8	UUUUUUUU YYYYYYYY VVVVVVVV	NvColorFormat_U8_Y8_V8
T_V8_Y8_U8	VVVVVVVV YYYYYYYY UUUUUUUU	NvColorFormat_V8_Y8_U8
T_U8_V8_Y8	UUUUUUUU VVVVVVVV YYYYYYYY	NvColorFormat_U8_V8_Y8
T_V8_U8_Y8	VVVVVVVV UUUUUUUU YYYYYYYY	NvColorFormat_V8_U8_Y8
T_Y8_U8_V8_TRUE	YYYYYYYY UUUUUUUU VVVVVVVV	NvColorFormat_Y8_U8_V8
T_Y8_V8_U8_TRUE	YYYYYYYY VVVVVVVV UUUUUUUU	NvColorFormat_Y8_V8_U8
T_U8_Y8_V8_TRUE	UUUUUUUU YYYYYYYY VVVVVVVV	NvColorFormat_U8_Y8_V8
T_V8_Y8_U8_TRUE	VVVVVVVV YYYYYYYY UUUUUUUU	NvColorFormat_V8_Y8_U8
T_U8_V8_Y8_TRUE	UUUUUUUU VVVVVVVV YYYYYYYY	NvColorFormat_U8_V8_Y8
T_V8_U8_Y8_TRUE	VVVVVVVV UUUUUUUU YYYYYYYY	NvColorFormat_V8_U8_Y8

Hardware Name	Bit pattern (Left to Right: LS-Byte to MS-Byte, ms-bit to ls-bit per Byte)	Software Name
T_R16_G16_B16_I	RRRRRRRRRRRRRRRR GGGGGGGGGGGGGGGG BBBBBBBBBBBBBBBB	NvColorFormat_Int_R16_G16_B16
T_R16_G16_B16_UI	RRRRRRRRRRRRRRRR GGGGGGGGGGGGGGGG BBBBBBBBBBBBBBBB	NvColorFormat_Uint_R16_G16_B16
T_R32_G32_B32	RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB	NvColorFormat_R32_G32_B32
T_R32_G32_B32_I	RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB	NvColorFormat_Int_R32_G32_B32
T_R32_G32_B32_UI	RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB	NvColorFormat_Uint_R32_G32_B32

7.9.1.6.4 Single Word Pixel Formats

Single Word type pixel formats, which use a single word to list all components in a single pixel, list components from left to right. It starts from the component in the most significant bit position and ends with the component in the least significant bit position. Little endian byte order is assumed when storing words into memory bytes.

Figure 7.60 T_A8B8G8R8

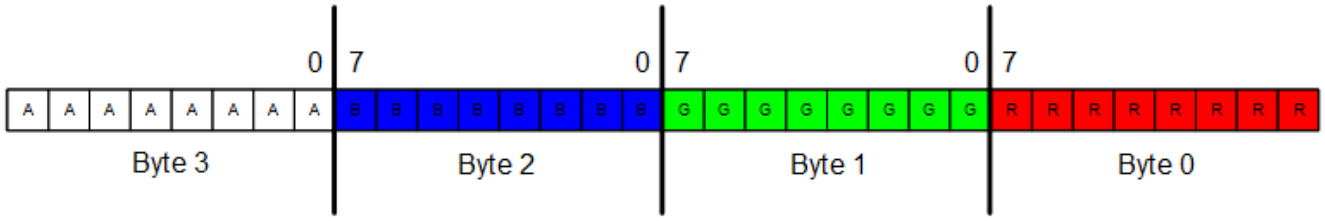


Table 7.99 Single Word Pixel Formats

Hardware Name	Bit Pattern (Left to Right: MS-Byte to LS-Byte, ms-bit to ls-bit per Byte)	Software Name
T_A8	AAAAAAAA	NvColorFormat_A8
T_L8	LLLLLLLL	NvColorFormat_L8
T_L4A4	LLLLAAAA	NvColorFormat_L4A4
T_A4L4	AAAALLLL	NvColorFormat_A4L4

Hardware Name	Bit Pattern (Left to Right: MS-Byte to LS-Byte, ms-bit to ls-bit per Byte)	Software Name
T_G4R4	GGGGRRRR	NvColorFormat_G4R4
T_R8	RRRRRRRR	NvColorFormat_R8
T_R8_SN	RRRRRRRR	NvColorFormat_SNorm_R8
T_R3G3B2	RRRGGGBB	NvColorFormat_R3G3B2
T_A8R8	AAAAAAAARRRRRRRR	NvColorFormat_A8R8
T_A8L8	AAAAAAAALLLLLLLL	NvColorFormat_A8L8
T_L8A8	LLLLLLLLAAAAAAAA	NvColorFormat_L8A8
T_R8G8	RRRRRRRRGGGGGGGG	NvColorFormat_R8G8
T_G8R8	GGGGGGGGRRRRRRRR	NvColorFormat_G8R8
T_G8R8_SN	GGGGGGGGRRRRRRRR	NvColorFormat_SNorm_G8R8
T_R6G5B5	RRRRRRGGGGGBBBBB	NvColorFormat_R6G5B5
T_R5G6B5	RRRRRGGGGGGBBBBB	NvColorFormat_R5G6B5
T_B5G6R5	BBBBBGGGGGGRRRRR	NvColorFormat_B5G6R5
T_B6G5R5	BBBBBBGGGGGRRRRR	NvColorFormat_B6G5R5
T_R5G5B6	RRRRRGGGGGBBBBBB	NvColorFormat_B6G5R5
T_R5G5B5A1	RRRRRGGGGGBBBBBA	NvColorFormat_R5G5B5A1
T_B5G5R5A1	BBBBBGGGGGRRRRRA	NvColorFormat_B5G5R5A1
T_A1B5G5R5	ABBBBBGGGGGRRRRR	NvColorFormat_A1B5G5R5
T_A1R5G5B5	ARRRRRGGGGGBBBBB	NvColorFormat_A1R5G5B5
T_R1G5B5A5	RGGGGGBBBBBAAAAA	NvColorFormat_R1G5B5A5
T_B5G5R1A5	BBBBBGGGGGRAAAAA	NvColorFormat_B5G5R1A5
T_A5B5G5R1	AAAAABBBBBGGGGGR	NvColorFormat_A5G5B5R1
T_A5R1G5B5	AAAAARGGGGGBBBBB	NvColorFormat_A5R1G5B5
T_R5G5B5X1	RRRRRGGGGGBBBBBX	NvColorFormat_R5G5B5X1
T_B5G5R5X1	BBBBBGGGGGRRRRRX	NvColorFormat_B5G5R5X1
T_X1B5G5R5	XBBBBBGGGGGRRRRR	NvColorFormat_X1B5G5R5
T_X1R5G5B5	XRRRRRGGGGGBBBBB	NvColorFormat_X1R5G5B5
T_R4G4B4A4	RRRRGGGGBBBBAAAA	NvColorFormat_R4G4B4A4
T_B4G4R4A4	BBBBGGGGRRRRAAAA	NvColorFormat_B4G4R4A4

Hardware Name	Bit Pattern (Left to Right: MS-Byte to LS-Byte, ms-bit to ls-bit per Byte)	Software Name
T_A4B4G4R4	AAAABBBBGGGGRRRR	NvColorFormat_A4B4G4R4
T_A4R4G4B4	AAAARRRRGGGGBBBB	NvColorFormat_A4R4G4B4
T_G16R16	GGGGGGGGGGGGGGGGRRRRRRRRRRRRRRRR	NvColorFormat_G16R16
T_G16R16_SN	GGGGGGGGGGGGGGGGRRRRRRRRRRRRRRRR	NvColorFormat_SNorm_G16R16
T_A8B8G8R8	AAAAAAAABBBBBBBBGGGGGGGGRRRRRRRR	NvColorFormat_A8B8G8R8
T_A8R8G8B8	AAAAAAAARRRRRRRRGGGGGGGGBBBBBBBB	NvColorFormat_A8R8G8B8
T_B8G8R8A8	BBBBBBBBGGGGGGGGRRRRRRRRAAAAAAAA	NvColorFormat_B8G8R8A8
T_G8B8G8R8	GGGGGGGGBBBBBBBBGGGGGGGGRRRRRRRR	NvColorFormat_G8B8G8R8
T_B8G8R8G8	BBBBBBBBGGGGGGGGRRRRRRRRGGGGGGGG	NvColorFormat_B8G8R8G8
T_R8G8B8A8	RRRRRRRRGGGGGGGGBBBBBBBBAAAAAAAA	NvColorFormat_R8G8B8A8
T_A8B8G8R8_SN	AAAAAAAABBBBBBBBGGGGGGGGRRRRRRRR	NvColorFormat_SNorm_A8B8G8R8
T_X8B8G8R8	XXXXXXXXBBBBBBBBGGGGGGGGRRRRRRRR	NvColorFormat_X8B8G8R8
T_X8R8G8B8	XXXXXXXXRRRRRRRRGGGGGGGGBBBBBBBB	NvColorFormat_X8R8G8B8
T_B8G8R8X8	BBBBBBBBGGGGGGGGRRRRRRRRXXXXXXXX	NvColorFormat_B8G8R8X8
T_R8G8B8X8	RRRRRRRRGGGGGGGGBBBBBBBBXXXXXXXX	NvColorFormat_R8G8B8X8
T_X8B8G8R8_LN	XXXXXXXXBBBBBBBBGGGGGGGGRRRRRRRR	NvColorFormat_X8B8G8R8
T_X8B8G8R8_SN	XXXXXXXXBBBBBBBBGGGGGGGGRRRRRRRR	NvColorFormat_SNorm_X8B8G8R8
T_X8R8G8B8_LN	XXXXXXXXRRRRRRRRGGGGGGGGBBBBBBBB	NvColorFormat_X8R8G8B8
T_A2B10G10R10	AABBBBBBBBBBGGGGGGGGGGRRRRRRRRRR	NvColorFormat_A2B10G10R10
T_A2R10G10B10	AARRRRRRRRRRGGGGGGGGGGBBBBBBBBBB	NvColorFormat_A2R10G10B10
T_B10G10R10A2	BBBBBBBBBBGGGGGGGGGGRRRRRRRRRRAA	NvColorFormat_B10G10R10A2
T_R10G10B10A2	RRRRRRRRRRGGGGGGGGGGBBBBBBBBBBAA	NvColorFormat_R10G10B10A2
T_A2Y10U10V10	AAYYYYYYYYYYUUUUUUUUUUVVVVVVVVVV	NvColorFormat_A2Y10U10V10
T_V10U10Y10A2	VVVVVVVVVVUUUUUUUUUUYYYYYYYYYYAA	NvColorFormat_V10U10Y10A2
T_X16B16G16R16	XXXXXXXXXXXXXXXXBBBBBBBBBBBBBBBBGGGGGGGGGGGGGGGGRRRRRRRRRRRRRRRR	NvColorFormat_X16B16G16R16
T_A16B16G16R16	AAAAAAAAAAAAAAAABBBBBBBBBBBBBBBBGGGGGGGGGGGGGGGGRRRRRRRRRRRRRRRR	NvColorFormat_A16B16G16R16
T_A16	AAAAAAAAAAAAAAAA	NvColorFormat_A16

Hardware Name	Bit Pattern (Left to Right: MS-Byte to LS-Byte, ms-bit to ls-bit per Byte)	Software Name
T_A16_F	AAAAAAAAAAAAAAAA	NvColorFormat_Float_A16
T_L16	LLLLLLLLLLLLLLLL	NvColorFormat_L16
T_L16_F	LLLLLLLLLLLLLLLL	NvColorFormat_Float_L16
T_L16_I	LLLLLLLLLLLLLLLL	NvColorFormat_Int_L16
T_R10	RRRRRRRRRR000000	NvColorFormat_R10
T_R12	RRRRRRRRRRRR0000	NvColorFormat_R12
T_R8_RJ_ISP	00000000RRRRRRRR	Bayer RAW8 right justified
T_R10_RJ_ISP	000000RRRRRRRRRR	Bayer RAW10 right justified
T_R12_RJ_ISP	0000RRRRRRRRRRRR	Bayer RAW12 right justified
T_R14_RJ_ISP	00RRRRRRRRRRRRRR	Bayer RAW14 right justified
T_R20_RJ_ISP	00000000000RRRRRRRRRRRRRRRRRRR	Bayer RAW20 right justified
T_R24_RJ_ISP	00000000RRRRRRRRRRRRRRRRRRRRRRRR	Bayer RAW24 right justified
T_R16	RRRRRRRRRRRRRRRR	NvColorFormat_R16
T_R16_F	RRRRRRRRRRRRRRRR	NvColorFormat_Float_R16
T_R16_I	RRRRRRRRRRRRRRRR	NvColorFormat_Int_R16
T_R16_SN	RRRRRRRRRRRRRRRR	NvColorFormat_SNorm_R16
T_R16_UI	RRRRRRRRRRRRRRRR	NvColorFormat_Uint_R16
T_A16L16_F	AAAAAAAAAAAAAAAALLLLLLLLLLLLLLLL	NvColorFormat_Float_A16L16
T_R24	RRRRRRRRRRRRRRRRRRRRRRRR	NvColorFormat_R32
T_A32_F	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	NvColorFormat_Float_A32
T_L32_F	LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL	NvColorFormat_Float_L32
T_L32_I	LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL	NvColorFormat_Int_L32
T_L32_UI	LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL	NvColorFormat_UInt_L32
T_R32	RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	NvColorFormat_ R32
T_R32_F	RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	NvColorFormat_Float_R32
T_G16R16_F	GGGGGGGGGGGGGGGGRRRRRRRRRRRRRRRR	NvColorFormat_Float_G16R16
T_E5B9G9R9_F	EEEEEBBBBBBBBBGGGGGGGGGRRRRRRRRR	NvColorFormat_Float_E5B9G9R9
T_B10G11R11_F	BBBBBBBBBBGGGGGGGGGGRRRRRRRRRR	NvColorFormat_Float_B10G11R11

Hardware Name	Bit Pattern (Left to Right: MS-Byte to LS-Byte, ms-bit to ls-bit per Byte)	Software Name
T_R11G11B10_F	RRRRRRRRRRRGGGGGGGGGGGBBBBBBBBBB	NvColorFormat_Float_R11G11B10
T_A2B10FG10FR10F	AABBBBBBBBBBGGGGGGGGGGRRRRRRRRRR	NvColorFormat_A2B10FG10FR10F
T_X16B16G16R16_F	XXXXXXXXXXXXXXXXBBBBBBBBBBBBBBBBGGGGGGGGGGGGGGGGRRRRRRRRRRRRRRRR	NvColorFormat_Float_X16B16G16R16
T_A16B16G16R16_F	AAAAAAAAAAAAAAAABBBBBBBBBBBBBBBBGGGGGGGGGGGGGGGGRRRRRRRRRRRRRRRR	NvColorFormat_Float_A16B16G16R16
T_G32R32_F	GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	NvColorFormat_Float_G32R32
T_X32B32G32R32_F	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	NvColorFormat_Float_X32B32G32R32
T_A32B32G32R32_F	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	NvColorFormat_Float_A32B32G32R32
T_S8	SSSSSSSS	NvColorFormat_S8
T_Z16	ZZZZZZZZZZZZZZZZ	NvColorFormat_Z16
T_Z16_NL	ZZZZZZZZZZZZZZZZ	NvColorFormat_Z16
T_G24R8	GGGGGGGGGGGGGGGGGGGGGGGGRRRRRRRR	NvColorFormat_G24R8
T_G8R24	GGGGGGGGRRRRRRRRRRRRRRRRRRRRRR	NvColorFormat_G8R24
T_Z24X8	ZZZZZZZZZZZZZZZZZZZZZZZZXXXXXXXX	NvColorFormat_Z24X8
T_Z24S8	ZZZZZZZZZZZZZZZZZZZZZZZZSSSSSSSS	NvColorFormat_Z24S8
T_X8Z24	XXXXXXXXZZZZZZZZZZZZZZZZZZZZZZZZ	NvColorFormat_X8Z24
T_S8Z24	SSSSSSSSZZZZZZZZZZZZZZZZZZZZZZZZ	NvColorFormat_S8Z24
T_Z32_F	ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ	NvColorFormat_Float_Z32
T_X24S8Z32_F	XXXXXXXXXXXXXXXXXXXXXXXXSSSSSSSSZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ	NvColorFormat_Float_X24S8Z32
T_R8_UI	RRRRRRRR	NvColorFormat_Uint_R8
T_R8_I	RRRRRRRR	NvColorFormat_Int_R8
T_R16_UI	RRRRRRRRRRRRRRRR	NvColorFormat_Uint_R16
T_R16_I	RRRRRRRRRRRRRRRR	NvColorFormat_Int_R16
T_R16_SN	RRRRRRRRRRRRRRRR	NvColorFormat_SNorm_R16
T_R16_ISP	RRRRRRRRRRRRRRRR	RAW ISP Floating Point Format

Hardware Name	Bit Pattern (Left to Right: MS-Byte to LS-Byte, ms-bit to ls-bit per Byte)	Software Name
T_R16_X_ISP20	RRRRRRRRRRRRRRRR	RAW ISP encoded float/20-bit fixed
T_R16_X_ISP24	RRRRRRRRRRRRRRRR	RAW ISP encoded float/24-bit fixed
T_G8R8_UI	GGGGGGGGRRRRRRRR	NvColorFormat_Uint_G8R8
T_G8R8_I	GGGGGGGGRRRRRRRR	NvColorFormat_Int_G8R8
T_R32_UI	RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	NvColorFormat_Uint_R32
T_R32_I	RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	NvColorFormat_Int_R32
T_G16R16	GGGGGGGGGGGGGGGGRRRRRRRRRRRRRRRR	NvColorFormat_G16R16
T_G16R16_F	GGGGGGGGGGGGGGGGRRRRRRRRRRRRRRRR	NvColorFormat_Float_G16R16
T_G16R16_I	GGGGGGGGGGGGGGGGRRRRRRRRRRRRRRRR	NvColorFormat_Int_G16R16
T_G16R16_UI	GGGGGGGGGGGGGGGGRRRRRRRRRRRRRRRR	NvColorFormat_Uint_G16R16
T_G16R16_SN	GGGGGGGGGGGGGGGGRRRRRRRRRRRRRRRR	NvColorFormat_SNorm_G16R16
T_A16R16	AAAAAAAAAAAAAAAARRRRRRRRRRRRRRRR	NvColorFormat_A16R16
T_A16R16_F	AAAAAAAAAAAAAAAARRRRRRRRRRRRRRRR	NvColorFormat_Float_A16R16
T_A2B10G10R10_UI	AABBBBBBBBBBGGGGGGGGGGRRRRRRRRRR	NvColorFormat_Uint_A2B10G10R10
T_A8B8LNG8LNR8LN	AAAAAAAABBBBBBBBGGGGGGGGRRRRRRRR	NvColorFormat_A8B8LNG8LNR8LN
T_A8R8LNG8LNB8LN	AAAAAAAARRRRRRRRGGGGGGGGBBBBBBBB	NvColorFormat_A8R8LNG8LNB8LN
T_A8B8G8R8_I	AAAAAAAABBBBBBBBGGGGGGGGRRRRRRRR	NvColorFormat_Int_A8B8G8R8
T_A8B8G8R8_UI	AAAAAAAABBBBBBBBGGGGGGGGRRRRRRRR	NvColorFormat_Uint_A8B8G8R8
T_G32R32	GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	NvColorFormat_G32R32
T_G32R32_I	GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	NvColorFormat_Int_G32R32
T_G32R32_UI	GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	NvColorFormat_Uint_G32R32
T_A32R32_F	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAARRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	NvColorFormat_Float_A32R32
T_B24G8R32_F	BBBBBBBBBBBBBBBBBBBBBBBBGGGGGGGGRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	NvColorFormat_Float_B24G8R32
T_A16B16G16R16_I	AAAAAAAAAAAAAAAABBBBBBBBBBBBBBBBGGGGGGGGGGGGGGGGRRRRRRRRRRRRRRRR	NvColorFormat_Int_A16B16G16R16

Hardware Name	Bit Pattern (Left to Right: MS-Byte to LS-Byte, ms-bit to ls-bit per Byte)	Software Name
T_A16B16G16R16_UI	AAAAAAAAAAAAAAAABBBBBBBBBBBBBBBBGGGGGGGGGGGGGGGGRRRRRRRRRRRRRRRR	NvColorFormat_Uint_A16B16G16R16
T_A16B16G16R16_SN	AAAAAAAAAAAAAAAABBBBBBBBBBBBBBBBGGGGGGGGGGGGGGGGRRRRRRRRRRRRRRRR	NvColorFormat_SNorm_A16B16G16R16
T_A16B16G16R16_NVB	AAAAAAAAAAAAAAAABBBBBBBBBBBBBBBBGGGGGGGGGGGGGGGGRRRRRRRRRRRRRRRR	NvColorFormat_NVBias_A16B16G16R16
T_X16B16G16R16_I	XXXXXXXXXXXXXXXXBBBBBBBBBBBBBBBBGGGGGGGGGGGGGGGGRRRRRRRRRRRRRRRR	NvColorFormat_Int_X16B16G16R16
T_X16B16G16R16_UI	XXXXXXXXXXXXXXXXBBBBBBBBBBBBBBBBGGGGGGGGGGGGGGGGRRRRRRRRRRRRRRRR	NvColorFormat_Uint_X16B16G16R16
T_A16Y16U16V16	AAAAAAAAAAAAAAAAYYYYYYYYYYYYYYYYUUUUUUUUUUUUUUUUVVVVVVVVVVVVVVVV	NvColorFormat_Uint_A16Y16U16V16
T_V16U16Y16A16	VVVVVVVVVVVVVVVVUUUUUUUUUUUUUUUUYYYYYYYYYYYYYYYYAAAAAAAAAAAAAAAA	NvColorFormat_Uint_V16U16Y16A16
T_V16U16Y16A16_I	VVVVVVVVVVVVVVVVUUUUUUUUUUUUUUUUYYYYYYYYYYYYYYYYAAAAAAAAAAAAAAAA	NvColorFormat_Int_V16U16Y16A16
T_A16Y16U16V16_F	AAAAAAAAAAAAAAAAYYYYYYYYYYYYYYYYUUUUUUUUUUUUUUUUVVVVVVVVVVVVVVVV	NvColorFormat_Float_A16Y16U16V16
T_A16Y16U16V16_I	AAAAAAAAAAAAAAAAYYYYYYYYYYYYYYYYUUUUUUUUUUUUUUUUVVVVVVVVVVVVVVVV	NvColorFormat_Int_A16Y16U16V16
T_A16Y16U16V16_ISP	AAAAAAAAAAAAAAAAYYYYYYYYYYYYYYYYUUUUUUUUUUUUUUUUVVVVVVVVVVVVVVVV	ISP-specific YUV–validation only
T_A32Y32U32V32_F	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV	NvColorFormat_Float_A32Y32U32V32
T_A32B32G32R32	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	NvColorFormat_A32B32G32R32
T_A32B32G32R32_I	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	NvColorFormat_Int_A32B32G32R32
T_A32B32G32R32_UI	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	NvColorFormat_Uint_A32B32G32R32
T_X32B32G32R32_I	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	NvColorFormat_Int_X32B32G32R32
T_X32B32G32R32_UI	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	NvColorFormat_Uint_X32B32G32R32
T_A8Y8U8V8	AAAAAAAAYYYYYYYYUUUUUUUUVVVVVVVV	NvColorFormat_A8Y8U8V8

Hardware Name	Bit Pattern (Left to Right: MS-Byte to LS-Byte, ms-bit to ls-bit per Byte)	Software Name
T_A8Y8V8U8	AAAAAAAAYYYYYYYYVVVVVVVVUUUUUUUU	NvColorFormat_A8Y8V8U8
T_A8U8V8Y8	AAAAAAAAUUUUUUUUVVVVVVVVYYYYYYYY	NvColorFormat_A8U8V8Y8
T_A8V8U8Y8	AAAAAAAAVVVVVVVVUUUUUUUUYYYYYYYY	NvColorFormat_A8V8U8Y8
T_Y8U8V8A8	YYYYYYYYUUUUUUUUVVVVVVVVAAAAAAAA	NvColorFormat_Y8U8V8A8
T_Y8V8U8A8	YYYYYYYYVVVVVVVVUUUUUUUUAAAAAAAA	NvColorFormat_Y8V8U8A8
T_U8V8Y8A8	UUUUUUUUVVVVVVVVYYYYYYYYAAAAAAAA	NvColorFormat_U8V8Y8A8
T_V8U8Y8A8	VVVVVVVVUUUUUUUUYYYYYYYYAAAAAAAA	NvColorFormat_V8U8Y8A8
T_A8Y8U8V8_TRUE	AAAAAAAAYYYYYYYYUUUUUUUUVVVVVVVV	NvColorFormat_A8Y8U8V8
T_A8Y8V8U8_TRUE	AAAAAAAAYYYYYYYYVVVVVVVVUUUUUUUU	NvColorFormat_A8Y8V8U8
T_A8U8V8Y8_TRUE	AAAAAAAAUUUUUUUUVVVVVVVVYYYYYYYY	NvColorFormat_A8U8V8Y8
T_A8V8U8Y8_TRUE	AAAAAAAAVVVVVVVVUUUUUUUUYYYYYYYY	NvColorFormat_A8V8U8Y8
T_Y8U8V8A8_TRUE	YYYYYYYYUUUUUUUUVVVVVVVVAAAAAAAA	NvColorFormat_Y8U8V8A8
T_Y8V8U8A8_TRUE	YYYYYYYYVVVVVVVVUUUUUUUUAAAAAAAA	NvColorFormat_Y8V8U8A8
T_U8V8Y8A8_TRUE	UUUUUUUUVVVVVVVVYYYYYYYYAAAAAAAA	NvColorFormat_U8V8Y8A8
T_V8U8Y8A8_TRUE	VVVVVVVVUUUUUUUUYYYYYYYYAAAAAAAA	NvColorFormat_V8U8Y8A8
T_U8V8	UUUUUUUUVVVVVVVV	NvColorFormat_U8V8
T_V8U8	VVVVVVVVUUUUUUUU	NvColorFormat_U8_V8
T_U8V8_TRUE	UUUUUUUUVVVVVVVV	NvColorFormat_U8V8
T_V8U8_TRUE	VVVVVVVVUUUUUUUU	NvColorFormat_U8_V8
T_Y8	YYYYYYYY	NvColorFormat_Y8
T_U8	UUUUUUUU	NvColorFormat_U8
T_V8	VVVVVVVV	NvColorFormat_V8
T_Y8_TRUE	YYYYYYYY	NvColorFormat_Y8
T_U8_TRUE	UUUUUUUU	NvColorFormat_U8
T_V8_TRUE	VVVVVVVV	NvColorFormat_V8
T_P8	PPPPPPPP	NvColorFormat_I8
T_A4P4	AAAAPPPP	NvColorFormat_A4I4
T_A8P8	AAAAAAAAPPPPPPPP	NvColorFormat_A8I8

7.9.1.6.5 Miscellaneous Pixel Formats

Table 7.100 Miscellaneous Pixel Formats

Hardware Name	Bit pattern (Left to Right: LS-Word to MS-Word, ms-bit to ls-bit per Word)	Software Name
T_S8_MSAA	SSSSSSSS	
T_U8_Y8__V8_Y8	UUUUUUUU YYYYYYYY VVVVVVVV YYYYYYYY	NvColorFormat_U8_Y8__V8_Y8 or NvColorFormat_UYVY
T_V8_Y8__U8_Y8	VVVVVVVV YYYYYYYY UUUUUUUU YYYYYYYY	NvColorFormat_V8_Y8__U8_Y8 or NvColorFormat_VYUY
T_Y8_U8__Y8_V8	YYYYYYYY UUUUUUUU YYYYYYYY VVVVVVVV	NvColorFormat_Y8_U8__Y8_V8 or NvColorFormat_YUYV
T_Y8_V8__Y8_U8	YYYYYYYY VVVVVVVV YYYYYYYY UUUUUUUU	NvColorFormat_Y8_V8__Y8_U8 or NvColorFormat_YVYU
T_U8_Y8__V8_Y8_TRUE	UUUUUUUU YYYYYYYY VVVVVVVV YYYYYYYY	NvColorFormat_U8_Y8__V8_Y8
T_V8_Y8__U8_Y8_TRUE	VVVVVVVV YYYYYYYY UUUUUUUU YYYYYYYY	NvColorFormat_V8_Y8__U8_Y8
T_Y8_U8__Y8_V8_TRUE	YYYYYYYY UUUUUUUU YYYYYYYY VVVVVVVV	NvColorFormat_Y8_U8__Y8_V8
T_Y8_V8__Y8_U8_TRUE	YYYYYYYY VVVVVVVV YYYYYYYY UUUUUUUU	NvColorFormat_Y8_V8__Y8_U8
T_B8_G8__R8_G8	BBBBBBBB GGGGGGGG RRRRRRRR GGGGGGGG	NvColorFormat_B8_G8__R8_G8
T_G8_B8__G8_R8	GGGGGGGG BBBBBBBB GGGGGGGG RRRRRRRR	NvColorFormat_G8_B8__G8_R8
T_R1	R	NvColorFormat_R1
T_P1	P	NvColorFormat_I1
T_P2	PP	NvColorFormat_I2
T_P4	PPPP	NvColorFormat_I4
T_DXT1_RGBA		
T_DXT1C_RGB		
T_DXT3_RGBA		
T_DXT5_RGBA		
T_ETC_RGB		
T_ETC3_RGBA		
T_ETC5_RGBA		

Hardware Name	Bit pattern (Left to Right: LS-Word to MS-Word, ms-bit to ls-bit per Word)	Software Name
T_LATC1		
T_LATC2		
T_ETC2_RGB		
T_ETC2P_RGBA		
T_ETC2A_RGBA		
T_EAC4BPP_R		
T_EAC4BPP_R_SN		
T_EAC4BPP_RG		
T_EAC4BPP_RG_SN		
DPCM_RAW10	RLE compressed format for RAW10 data	Private VI output format
DPCM_RAW12	RLE compressed format for RAW12 data	Private VI output format
DPCM_RAW16	RLE compressed format for RAW16 data	Private VI output format
DPCM_RAW20	RLE compressed format for RAW20 data	Private VI output format

7.9.1.6.6 Multi-Pixel Packing Pixel Formats

Pixel data can come in nonpower of two bit sizes that are difficult to fetch and process when contained in certain memory formats (for example, block linear). For example, raw camera data can come in 6-bit and 10-bit pixel sizes. It can simplify pixel data fetch and processing by packing multiple pixels into a single large word.

Table 7.101 Multiple Plane Pixel Formats

Hardware Name	Bit pattern (Left to Right: MS-Byte to LS-Byte, ms-bit to ls-bit per Byte)
T_X2Lc10Lb10La10	XX $L^2L^2L^2L^2L^2L^2L^2L^2L^2L^2$ $L^1L^1L^1L^1L^1L^1L^1L^1L^1L^1$ $L^0L^0L^0L^0L^0L^0L^0L^0L^0L^0$
T_X2Le6Ld6Lc6Lb6La6	XX $L^4L^4L^4L^4L^4L^4$ $L^3L^3L^3L^3L^3L^3$ $L^2L^2L^2L^2L^2L^2$ $L^1L^1L^1L^1L^1L^1$ $L^0L^0L^0L^0L^0L^0$

7.9.1.6.7 Multiple Plane Pixel Formats

Some color formats, for example, describing the YUV color space, split pixel data across multiple planes. Fully planar YUV formats contain three planes, one each for Y, U, and V components. Semiplanar YUV formats contain two planes, one for Y, and one shared between U and V

components. Some units may use a different color format name for each plane. Other units may use a single color format name describing all planes in a single name as described in the following table.

Table 7.102 8-bit Multiple Plane Pixel Formats

Hardware Name	Bit Pattern (Left to Right: MS-Byte to LS-Byte, ms-bit to ls-bit per Byte)	FOURCC
T_Y8___U8V8_N444 Y8,U8V8 (YUV444 semiplanar)	Plane0: YYYYYYYY Plane1: UUUUUUUUVVVVVVVV	
T_Y8___V8U8_N444 Y8,V8U8 (YUV444 semiplanar)	Plane0: YYYYYYYY Plane1: VVVVVVVVUUUUUUUU	
T_Y8___U8V8_N422 Y8,U8V8 (YUV422 semiplanar)	Plane0: YYYYYYYY Plane1: UUUUUUUUVVVVVVVV	NV61
T_Y8___V8U8_N422 Y8,V8U8 (YUV422 semiplanar)	Plane0: YYYYYYYY Plane1: VVVVVVVVUUUUUUUU	NV16
T_Y8___U8V8_N422R Y8,U8V8 (YUV422R semiplanar)	Plane0: YYYYYYYY Plane1: UUUUUUUUVVVVVVVV	
T_Y8___V8U8_N422R Y8,V8U8 (YUV422R semiplanar)	Plane0: YYYYYYYY Plane1: VVVVVVVVUUUUUUUU	
T_Y8___U8V8_N420 Y8,U8V8 (YUV420 semiplanar)	Plane0: YYYYYYYY Plane1: UUUUUUUUVVVVVVVV	NV21
T_Y8___V8U8_N420 Y8,V8U8 (YUV420 semiplanar)	Plane0: YYYYYYYY Plane1: VVVVVVVVUUUUUUUU	NV12
T_Y8___U8___V8_N444 Y8,U8, V8 (YUV444 full planar)	Plane0: YYYYYYYY Plane1: UUUUUUUU Plane2: VVVVVVVV	YV24
T_Y8___U8___V8_N422 Y8,U8, V8 (YUV422 full planar)	Plane0: YYYYYYYY Plane1: UUUUUUUU Plane2: VVVVVVVV	YV16
T_Y8___U8___V8_N422R Y8,U8, V8 (YUV422R full planar)	Plane0: YYYYYYYY Plane1: UUUUUUUU Plane2: VVVVVVVV	
T_Y8___U8___V8_N420 Y8,U8, V8 (YUV420 full planar)	Plane0: YYYYYYYY Plane1: UUUUUUUU Plane2: VVVVVVVV	YV12
T_Y8___U8V8_N444_TRUE Y8,U8V8 (YUV444 semiplanar)	Plane0: YYYYYYYY Plane1: UUUUUUUUVVVVVVVV	
T_Y8___V8U8_N444_TRUE Y8,V8U8 (YUV444 semiplanar)	Plane0: YYYYYYYY Plane1: VVVVVVVVUUUUUUUU	
T_Y8___U8V8_N422_TRUE Y8,U8V8 (YUV422 semiplanar)	Plane0: YYYYYYYY Plane1: UUUUUUUUVVVVVVVV	

Hardware Name	Bit Pattern (Left to Right: MS-Byte to LS-Byte, ms-bit to ls-bit per Byte)	FOURCC
T_Y8___V8U8_N422_TRUE Y8,V8U8 (YUV422 semiplanar)	Plane0: YYYYYYYY Plane1: VVVVVVVVUUUUUUUU	
T_Y8___U8V8_N422R_TRUE Y8,U8V8 (YUV422R semiplanar)	Plane0: YYYYYYYY Plane1: UUUUUUUUVVVVVVVV	
T_Y8___V8U8_N422R_TRUE Y8,V8U8 (YUV422R semiplanar)	Plane0: YYYYYYYY Plane1: VVVVVVVVUUUUUUUU	
T_Y8___U8V8_N420_TRUE Y8,U8V8 (YUV420 semiplanar)	Plane0: YYYYYYYY Plane1: UUUUUUUUVVVVVVVV	
T_Y8___V8U8_N420_TRUE Y8,V8U8 (YUV420 semiplanar)	Plane0: YYYYYYYY Plane1: VVVVVVVVUUUUUUUU	
T_Y8___U8___V8_N444_TRUE Y8,U8, V8 (YUV444 full planar)	Plane0: YYYYYYYY Plane1: UUUUUUUU Plane2: VVVVVVVV	
T_Y8___U8___V8_N422_TRUE Y8,U8, V8 (YUV422 full planar)	Plane0: YYYYYYYY Plane1: UUUUUUUU Plane2: VVVVVVVV	
T_Y8___U8___V8_N422R_TRUE Y8,U8, V8 (YUV422R full planar)	Plane0: YYYYYYYY Plane1: UUUUUUUU Plane2: VVVVVVVV	
T_Y8___U8___V8_N420_TRUE Y8,U8, V8 (YUV420 full planar)	Plane0: YYYYYYYY Plane1: UUUUUUUU Plane2: VVVVVVVV	

Table 7.103 10-bit Multiple Plane Pixel Formats

Hardware Name	Bit pattern (Left to Right: MS-Byte to LS-Byte, ms-bit to ls-bit per Byte)	FOURCC
T_Y10___U10V10_N444 Y10,U10V10 (YUV444 semiplanar)	Plane0: YYYYYYYYYY000000 Plane1: UUUUUUUUUU000000VVVVVVVVVV000000	
T_Y10___V10U10_N444 Y10,V10U10 (YUV444 semiplanar)	Plane0: YYYYYYYYYY000000 Plane1: VVVVVVVVVV000000UUUUUUUUUU000000	
T_Y10___U10V10_N422 Y10,U10V10 (YUV422 semiplanar)	Plane0: YYYYYYYYYY000000 Plane1: UUUUUUUUUU000000VVVVVVVVVV000000	NV61
T_Y10___V10U10_N422 Y10,V10U10 (YUV422 semiplanar)	Plane0: YYYYYYYYYY000000 Plane1: VVVVVVVVVV000000UUUUUUUUUU000000	NV16
T_Y10___U10V10_N422R Y10,U10V10 (YUV422R semiplanar)	Plane0: YYYYYYYYYY000000 Plane1: UUUUUUUUUU000000VVVVVVVVVV000000	
T_Y10___V10U10_N422R Y10,V10U10 (YUV422R semiplanar)	Plane0: YYYYYYYYYY000000 Plane1: VVVVVVVVVV000000UUUUUUUUUU000000	

Hardware Name	Bit pattern (Left to Right: MS-Byte to LS-Byte, ms-bit to ls-bit per Byte)	FOURCC
T_Y10___U10V10_N420 Y10,U10V10 (YUV420 semiplanar)	Plane0: YYYYYYYYYY000000 Plane1: UUUUUUUUUU000000VVVVVVVVVV000000	NV21
T_Y10___V10U10_N420 Y10,V10U10 (YUV420 semiplanar)	Plane0: YYYYYYYYYY000000 Plane1: VVVVVVVVVV000000UUUUUUUUUU000000	NV12
T_Y10___U10___V10_N444 Y10,U10, V10 (YUV444 full planar)	Plane0: YYYYYYYYYY000000 Plane1: UUUUUUUUUU000000 Plane2: VVVVVVVVVV000000	YV24
T_Y10___U10___V10_N422 Y10,U10, V10 (YUV422 full planar)	Plane0: YYYYYYYYYY000000 Plane1: UUUUUUUUUU000000 Plane2: VVVVVVVVVV000000	YV16
T_Y10___U10___V10_N422R Y10,U10, V10 (YUV422R full planar)	Plane0: YYYYYYYYYY000000 Plane1: UUUUUUUUUU000000 Plane2: VVVVVVVVVV000000	
T_Y10___U10___V10_N420 Y10,U10, V10 (YUV420 full planar)	Plane0: YYYYYYYYYY000000 Plane1: UUUUUUUUUU000000 Plane2: VVVVVVVVVV000000	YV12
T_Y10___U10V10_N444_TRUE Y10,U10V10 (YUV444 semiplanar)	Plane0: YYYYYYYYYY000000 Plane1: UUUUUUUUUU000000VVVVVVVVVV000000	
T_Y10___V10U10_N444_TRUE Y10,V10U10 (YUV444 semiplanar)	Plane0: YYYYYYYYYY000000 Plane1: VVVVVVVVVV000000UUUUUUUUUU000000	
T_Y10___U10V10_N422_TRUE Y10,U10V10 (YUV422 semiplanar)	Plane0: YYYYYYYYYY000000 Plane1: UUUUUUUUUU000000VVVVVVVVVV000000	
T_Y10___V10U10_N422_TRUE Y10,V10U10 (YUV422 semiplanar)	Plane0: YYYYYYYYYY000000 Plane1: VVVVVVVVVV000000UUUUUUUUUU000000	
T_Y10___U10V10_N422R_TRUE Y10,U10V10 (YUV422R semiplanar)	Plane0: YYYYYYYYYY000000 Plane1: UUUUUUUUUU000000VVVVVVVVVV000000	
T_Y10___V10U10_N422R_TRUE Y10,V10U10 (YUV422R semiplanar)	Plane0: YYYYYYYYYY000000 Plane1: VVVVVVVVVV000000UUUUUUUUUU000000	
T_Y10___U10V10_N420_TRUE Y10,U10V10 (YUV420 semiplanar)	Plane0: YYYYYYYYYY000000 Plane1: UUUUUUUUUU000000VVVVVVVVVV000000	
T_Y10___V10U10_N420_TRUE Y10,V10U10 (YUV420 semiplanar)	Plane0: YYYYYYYYYY000000 Plane1: VVVVVVVVV000000VUUUUUUUUUU000000	
T_Y10___U10___V10_N444_TRUE Y10,U10, V10 (YUV444 full planar)	Plane0: YYYYYYYYYY000000 Plane1: UUUUUUUUUU000000 Plane2: VVVVVVVVVV000000	
T_Y10___U10___V10_N422_TRUE Y10,U10, V10 (YUV422 full planar)	Plane0: YYYYYYYYYY000000 Plane1: UUUUUUUUUU000000 Plane2: VVVVVVVVVV000000	

Hardware Name	Bit pattern (Left to Right: MS-Byte to LS-Byte, ms-bit to ls-bit per Byte)	FOURCC
T_Y10___U10___V10_N422R_TRUE Y10,U10, V10 (YUV422R full planar)	Plane0: YYYYYYYYYY000000 Plane1: UUUUUUUUUU000000 Plane2: VVVVVVVVVV000000	
T_Y10___U10___V10_N420_TRUE Y10,U10, V10 (YUV420 full planar)	Plane0: YYYYYYYYYY000000 Plane1: UUUUUUUUUU000000 Plane2: VVVVVVVVVV000000	

Table 7.104 12-bit Multiple Plane Pixel Formats

Hardware Name	Bit pattern (Left to Right: MS-Byte to LS-Byte, ms-bit to ls-bit per Byte)	FOURCC
T_Y12___U12V12_N444 Y12,U12V12 (YUV444 semiplanar)	Plane0: YYYYYYYYYYYY0000 Plane1: UUUUUUUUUUUU0000VVVVVVVVVVVV0000	
T_Y12___V12U12_N444 Y12,V12U12 (YUV444 semiplanar)	Plane0: YYYYYYYYYYYY0000 Plane1: VVVVVVVVVVVV0000UUUUUUUUUUUU0000	
T_Y12___U12V12_N422 Y12,U12V12 (YUV422 semiplanar)	Plane0: YYYYYYYYYYYY0000 Plane1: UUUUUUUUUUUU0000VVVVVVVVVVVV0000	NV61
T_Y12___V12U12_N422 Y12,V12U12 (YUV422 semiplanar)	Plane0: YYYYYYYYYYYY0000 Plane1: VVVVVVVVVVVV0000UUUUUUUUUUUU0000	NV16
T_Y12___U12V12_N422R Y12,U12V12 (YUV422R semiplanar)	Plane0: YYYYYYYYYYYY0000 Plane1: UUUUUUUUUUUU0000VVVVVVVVVVVV0000	
T_Y12___V12U12_N422R Y12,V12U12 (YUV422R semiplanar)	Plane0: YYYYYYYYYYYY0000 Plane1: VVVVVVVVVVVV0000UUUUUUUUUUUU0000	
T_Y12___U12V12_N420 Y12,U12V12 (YUV420 semiplanar)	Plane0: YYYYYYYYYYYY0000 Plane1: UUUUUUUUUUUU0000VVVVVVVVVVVV0000	NV21
T_Y12___V12U12_N420 Y12,V12U12 (YUV420 semiplanar)	Plane0: YYYYYYYYYYYY0000 Plane1: VVVVVVVVVVVV0000UUUUUUUUUUUU0000	NV12
T_Y12___U12___V12_N444 Y12,U12, V12 (YUV444 full planar)	Plane0: YYYYYYYYYYYY0000 Plane1: UUUUUUUUUUUU0000 Plane2: VVVVVVVVVVVV0000	YV24
T_Y12___U12___V12_N422 Y12,U12, V12 (YUV422 full planar)	Plane0: YYYYYYYYYYYY0000 Plane1: UUUUUUUUUUUU0000 Plane2: VVVVVVVVVVVV0000	YV16
T_Y12___U12___V12_N422R Y12,U12, V12 (YUV422R full planar)	Plane0: YYYYYYYYYYYY0000 Plane1: UUUUUUUUUUUU0000 Plane2: VVVVVVVVVVVV0000	
T_Y12___U12___V12_N420 Y12,U12, V12 (YUV420 full planar)	Plane0: YYYYYYYYYYYY0000 Plane1: UUUUUUUUUUUU0000 Plane2: VVVVVVVVVVVV0000	YV12

Hardware Name	Bit pattern (Left to Right: MS-Byte to LS-Byte, ms-bit to ls-bit per Byte)	FOURCC
T_Y12___U12V12_N444_TRUE Y12,U12V12 (YUV444 semiplanar)	Plane0: YYYYYYYYYYYY0000 Plane1: UUUUUUUUUUUU0000VVVVVVVVVVVV0000	
T_Y12___V12U12_N444_TRUE Y12,V12U12 (YUV444 semiplanar)	Plane0: YYYYYYYYYYYY0000 Plane1: VVVVVVVVVVVV0000UUUUUUUUUUUU0000	
T_Y12___U12V12_N422_TRUE Y12,U12V12 (YUV422 semiplanar)	Plane0: YYYYYYYYYYYY0000 Plane1: UUUUUUUUUUUU0000VVVVVVVVVVVV0000	
T_Y12___V12U12_N422_TRUE Y12,V12U12 (YUV422 semiplanar)	Plane0: YYYYYYYYYYYY0000 Plane1: VVVVVVVVVVVV0000UUUUUUUUUUUU0000	
T_Y12___U12V12_N422R_TRUE Y12,U12V12 (YUV422R semiplanar)	Plane0: YYYYYYYYYYYY0000 Plane1: UUUUUUUUUUUU0000VVVVVVVVVVVV0000	
T_Y12___V12U12_N422R_TRUE Y12,V12U12 (YUV422R semiplanar)	Plane0: YYYYYYYYYYYY0000 Plane1: VVVVVVVVVVVV0000UUUUUUUUUUUU0000	
T_Y12___U12V12_N420_TRUE Y12,U12V12 (YUV420 semiplanar)	Plane0: YYYYYYYYYYYY0000 Plane1: UUUUUUUUUUUU0000VVVVVVVVVVVV0000	
T_Y12___V12U12_N420_TRUE Y12,V12U12 (YUV420 semiplanar)	Plane0: YYYYYYYYYYYY0000 Plane1: VVVVVVVVVVVV0000VUUUUUUUUUUUU0000	
T_Y12___U12___V12_N444_TRUE Y12,U12, V12 (YUV444 full planar)	Plane0: YYYYYYYYYYYY0000 Plane1: UUUUUUUUUUUU0000 Plane2: VVVVVVVVVVVV0000	
T_Y12___U12___V12_N422_TRUE Y12,U12, V12 (YUV422 full planar)	Plane0: YYYYYYYYYYYY0000 Plane1: UUUUUUUUUUUU0000 Plane2: VVVVVVVVVVVV0000	
T_Y12___U12___V12_N422R_TRUE Y12,U12, V12 (YUV422R full planar)	Plane0: YYYYYYYYYYYY0000 Plane1: UUUUUUUUUUUU0000 Plane2: VVVVVVVVVVVV0000	
T_Y12___U12___V12_N420_TRUE Y12,U12, V12 (YUV420 full planar)	Plane0: YYYYYYYYYYYY0000 Plane1: UUUUUUUUUUUU0000 Plane2: VVVVVVVVVVVV0000	

Table 7.105 16-bit Multiple Plane Pixel Formats

Hardware Name	Bit Pattern (Left to Right: MS-Byte to LS-Byte, ms-bit to ls-bit per Byte)	FOURCC
T_Y16___U16V16_N444 Y16,U16V16 (YUV444 semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUUUUUUUUUUVVVVVVVVVVVVVVVV	
T_Y16___V16U16_N444 Y16,V16U16 (YUV444 semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: VVVVVVVVVVVVVVVVUUUUUUUUUUUUUUUU	
T_Y16___U16V16_N422 Y16,U16V16 (YUV422 semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUUUUUUUUUUVVVVVVVVVVVVVVVV	NV61

Hardware Name	Bit Pattern (Left to Right: MS-Byte to LS-Byte, ms-bit to ls-bit per Byte)	FOURCC
T_Y16___V16U16_N422 Y16,V16U16 (YUV422 semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: VVVVVVVVVVVVVVVVUUUUUUUUUUUUUUUU	NV16
T_Y16___U16V16_N422R Y16,U16V16 (YUV422R semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUUUUUUUUUUVVVVVVVVVVVVVVVV	
T_Y16___V16U16_N422R Y16,V16U16 (YUV422R semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: VVVVVVVVVVVVVVVVUUUUUUUUUUUUUUUU	
T_Y16___U16V16_N420 Y16,U16V16 (YUV420 semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUUUUUUUUUUVVVVVVVVVVVVVVVV	NV21
T_Y16___V16U16_N420 Y16,V16U16 (YUV420 semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: VVVVVVVVVVVVVVVVUUUUUUUUUUUUUUUU	NV12
T_Y16___U16___V16_N444 Y16,U16, V16 (YUV444 full planar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUUUUUUUUUU Plane2: VVVVVVVVVVVVVVVV	YV24
T_Y16___U16___V16_N422 Y16,U16, V16 (YUV422 full planar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUUUUUUUUUU Plane2: VVVVVVVVVVVVVVVV	YV16
T_Y16___U16___V16_N422R Y16,U16, V16 (YUV422R full planar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUUUUUUUUUU Plane2: VVVVVVVVVVVVVVVV	
T_Y16___U16___V16_N420 Y16,U16, V16 (YUV420 full planar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUUUUUUUUUU Plane2: VVVVVVVVVVVVVVVV	YV12
T_Y16___U16V16_N444_TRUE Y16,U16V16 (YUV444 semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUUUUUUUUUUVVVVVVVVVVVVVVVV	
T_Y16___V16U16_N444_TRUE Y16,V16U16 (YUV444 semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: VVVVVVVVVVVVVVVVUUUUUUUUUUUUUUUU	
T_Y16___U16V16_N422_TRUE Y16,U16V16 (YUV422 semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUUUUUUUUUUVVVVVVVVVVVVVVVV	
T_Y16___V16U16_N422_TRUE Y16,V16U16 (YUV422 semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: VVVVVVVVVVVVVVVVUUUUUUUUUUUUUUUU	
T_Y16___U16V16_N422R_TRUE Y16,U16V16 (YUV422R semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUUUUUUUUUUVVVVVVVVVVVVVVVV	
T_Y16___V16U16_N422R_TRUE Y16,V16U16 (YUV422R semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: VVVVVVVVVVVVVVVVUUUUUUUUUUUUUUUU	
T_Y16___U16V16_N420_TRUE Y16,U16V16 (YUV420 semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUUUUUUUUUUVVVVVVVVVVVVVVVV	
T_Y16___V16U16_N420_TRUE Y16,V16U16 (YUV420 semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: VVVVVVVVVVVVVVVVUUUUUUUUUUUUUUUU	

Hardware Name	Bit Pattern (Left to Right: MS-Byte to LS-Byte, ms-bit to ls-bit per Byte)	FOURCC
T_Y16___U16___V16_N444_TRUE Y16,U16, V16 (YUV444 full planar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUUUUUUUUUU Plane2: VVVVVVVVVVVVVVVV	
T_Y16___U16___V16_N422_TRUE Y16,U16, V16 (YUV422 full planar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUUUUUUUUUU Plane2: VVVVVVVVVVVVVVVV	
T_Y16___U16___V16_N422R_TRUE Y16,U16, V16 (YUV422R full planar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUUUUUUUUUU Plane2: VVVVVVVVVVVVVVVV	
T_Y16___U16___V16_N420_TRUE Y16,U16, V16 (YUV420 full planar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUUUUUUUUUU Plane2: VVVVVVVVVVVVVVVV	

Table 7.106 16-bit Y/8-bit UV Multiple Plane Pixel Formats

Hardware Name	Bit Pattern (Left to Right: MS-Byte to LS-Byte, ms-bit to ls-bit per Byte)	FOURCC
T_Y16___U8V8_N444 Y16,U8V8 (YUV444 semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUUVVVVVVVV	
T_Y16___V8U8_N444 Y16,V8U8 (YUV444 semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: VVVVVVVVUUUUUUUU	
T_Y16___U8V8_N422 Y16,U8V8 (YUV422 semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUUVVVVVVVV	NV61
T_Y16___V8U8_N422 Y16,V8U8 (YUV422 semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: VVVVVVVVUUUUUUUU	NV16
T_Y16___U8V8_N422R Y16,U8V8 (YUV422R semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUUVVVVVVVV	
T_Y16___V8U8_N422R Y16,V8U8 (YUV422R semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: VVVVVVVVUUUUUUUU	
T_Y16___U8V8_N420 Y16,U8V8 (YUV420 semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUUVVVVVVVV	NV21
T_Y16___V8U8_N420 Y16,V8U8 (YUV420 semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUUVVVVVVVV	NV12
T_Y16___U8___V8_N444 Y16,U8, V8 (YUV444 full planar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUU Plane2: VVVVVVVV	YV24
T_Y16___U8___V8_N422 Y16,U8, V8 (YUV422 full planar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUU Plane2: VVVVVVVV	YV16

Hardware Name	Bit Pattern (Left to Right: MS-Byte to LS-Byte, ms-bit to ls-bit per Byte)	FOURCC
T_Y16___U8___V8_N422R Y16,U8, V8 (YUV422R full planar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUU Plane2: VVVVVVVV	
T_Y16___U8___V8_N420 Y16,U8, V8 (YUV420 full planar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUU Plane2: VVVVVVVV	YV16
T_Y16___U8V8_N444_TRUE Y16,U16V16 (YUV444 semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUUVVVVVVVV	
T_Y16___V8U8_N444_TRUE Y16,V8U8 (YUV444 semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: VVVVVVVVUUUUUUUU	
T_Y16___U8V8_N422_TRUE Y16,U8V8 (YUV422 semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUUVVVVVVVV	
T_Y16___V8U8_N422_TRUE Y16,V8U8 (YUV422 semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: VVVVVVVVUUUUUUUU	
T_Y16___U8V8_N422R_TRUE Y16,U8V8 (YUV422R semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUUVVVVVVVV	
T_Y16___V8U8_N422R_TRUE Y16,V8U8 (YUV422R semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: VVVVVVVVUUUUUUUU	
T_Y16___U8V8_N420_TRUE Y16,U8V8 (YUV420 semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUUVVVVVVVV	
T_Y16___V8U8_N420_TRUE Y16,V8U8 (YUV420 semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: VVVVVVVVUUUUUUUU	
T_Y16___U8___V8_N444_TRUE Y16,U8, V8 (YUV444 full planar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUU Plane2: VVVVVVVV	
T_Y16___U8___V8_N422_TRUE Y16,U8, V8 (YUV422 full planar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUU Plane2: VVVVVVVV	
T_Y16___U8___V8_N422R_TRUE Y16,U8, V8 (YUV422R full planar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUU Plane2: VVVVVVVV	
T_Y16___U8___V8_N420_TRUE Y16,U8, V8 (YUV420 full planar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUU Plane2: VVVVVVVV	
T_Y16I___U8V8_N420 Y16,U8V8 (YUV420 semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUUVVVVVVVV	ISP-only Y-int
T_Y16F___U8V8_N420 Y16,U8V8 (YUV420 semiplanar)	Plane0: YYYYYYYYYYYYYYYY Plane1: UUUUUUUUVVVVVVVV	Y-float

7.9.1.6.8 Bayer Pixel Formats

Bayer pixel data can come from camera sensor equipment in various sizes, e.g., 6 bits, 10 bits, 14 bits, 16 bits, etc. Bayer (also referred to as RAW) data is stored using the following pixel formats:

- T_R8, T_R16, T_R16_I, T_R16_F, T_R24, T_R32, T_R32_F, T_R16_X_ISP20, T_R16_X_ISP24

7.9.1.6.9 Supported Pixel Formats

This table summarizes the pixel formats supported by each of the pixel-capable bus masters. A '1' indicates support for that format by the device identified at the head of the corresponding column.

Note: T_* convention - MSB to LSB if no "_" notation*

Table 7.107 Supported Pixel Formats

Supported Pixel Format (T_* convention - MSB to LSB if no "_" notation)*	NVDisplay	NVDEC	NVENC	NVOFA	NVJPG	NVIDIA Ampere Architecture Tegra Tex	NVIDIA Ampere Architecture Tegra CROP	VIC	VI	ISP	PVA	DLA
Single Plane Surface Formats												
T_A8	-	-	-	-	-	1	1	1	-	-	-	-
T_A4	-	-	-	-	-	-	-	-	-	-	-	-
T_A2	-	-	-	-	-	-	-	-	-	-	-	-
T_A1	-	-	-	-	-	-	-	-	-	-	-	-
T_L8/T_Y8/YUV100	-	-	-	-	Decoder:1 Encoder: -	-	1	-	1	-	1	-
T_L16	-	-	-	-	-	-	-	1	-	1	-	-
T_L16_I	-	-	-	-	-	-	-	-	-	1	-	-
T_L16_F	-	-	-	-	-	-	-	-	-	1	-	-
T_L32_I	-	-	-	-	-	-	-	-	-	1	-	-
T_L32_F	-	-	-	-	-	-	-	-	-	1	-	-
T_A4L4	-	-	-	-	-	-	-	1	-	-	-	-
T_L4A4	-	-	-	-	-	-	-	1	-	-	-	-

Supported Pixel Format (T_* convention - MSB to LSB if no "_" notation)*	NVDisplay	NVDEC	NVENC	NVOFA	NVJPG	NVIDIA Ampere Architecture Tegra Tex	NVIDIA Ampere Architecture Tegra CROP	VIC	VI	ISP	PVA	DLA
T_R4												1
T_R4_I												1
T_R8	-	-	-	1	-	1	1	1	1	-	1	1
T_R10	-	-	-	1	-	-	-	-	-	-	-	1
T_R12	-	-	-	1	-	-	-	-	-	-	-	1
T_R8_I	-	-	-	-	-	-	-	-	-	-	1	1
T_R16	-	-	-	1	-	1	1	1	1	1	1	1
T_R16_I	-	-	-	1	-	-	-	-	1	1	1	1
T_R16_F	-	-	-	-	-	-	-	-	1	1	1	1
T_R8_RJ_ISP									1			
T_R10_RJ_ISP									1			
T_R12_RJ_ISP									1			
T_R14_RJ_ISP									1			
T_R20_RJ_ISP									1			
T_R24_RJ_ISP									1			
T_R16_X_ISP20	-	-	-	-	-	-	-	-	1	1	-	-
T_R16_X_ISP24	-	-	-	-	-	-	-	-	1	1	-	-
T_R24	-	-	-	-	-	-	-	-	1	1	-	-
T_R32	-	-	-	-	-	-	-	-	1	1	1	-
T_R32_I	-	-	-	-	-	-	-	-	-	1	1	-
T_R32_F	-	-	-	-	-	-	-	-	1	1	1	-
T_G16R16	-	-	-	-	-	1	1	-	-	-	1	-
T_G16R16_I	-	-	-	1	-	1	1	-	-	-	1	-
T_A16B16G16R16	1	-	-	-	-	1	1	1	-	-	1	1

Supported Pixel Format (T_* convention - MSB to LSB if no "_" notation)*	NVDisplay	NVDEC	NVENC	NVOFA	NVJPG	NVIDIA Ampere Architecture Tegra Tex	NVIDIA Ampere Architecture Tegra CROP	VIC	VI	ISP	PVA	DLA
T_A16B16G16R16_I	-	-	-	-	-	-	-	-	-	-	1	-
T_A16B16G16R16_NVB	1	-	-	-	-	-	-	-	-	-	-	-
T_X16B16G16R16	-	-	-	-	-	-	-	-	-	1	1	1
T_X16B16G16R16_I	-	-	-	-	-	-	-	-	-	1	1	-
T_A16B16G16R16_F	1	-	-	-	-	1	1	-	-	1	1	1
T_A16Y16U16V16	-	-	-	-	-	-	-	1	-	1	1	1
T_A16Y16U16V16_I	-	-	-	-	-	-	-	-	-	1	1	-
T_V16U16Y16A16	-	-	-	-	-	-	-	-	-	1	1	1
T_V16U16Y16A16_I	-	-	-	-	-	-	-	-	-	1	1	-
T_A16Y16U16V16_F	-	-	-	-	-	-	-	-	-	1	1	1
T_A32B32G32R32_F	-	-	-	-	-	-	-	-	-	1	-	-
T_A32Y32U32V32_F	-	-	-	-	-	-	-	-	-	1	-	-
T_X2Lc10Lb10La10	-	-	-	-	-	-	-	-	-	-	-	-
T_A8L8	-	-	-	-	-	-	-	1	-	-	-	-
T_L8A8	-	-	-	-	-	-	-	1	-	-	-	-
T_R8G8	-	-	-	-	-	1	-	1	-	-	-	-
T_G8R8	-	-	-	-	-	1	1	1	-	-	-	-
T_G8R8_I	-	-	-	-	-	-	-	-	-	-	-	-
T_G8R24	-	-	-	-	-	1	-	-	-	-	-	-
T_G24R8	-	-	-	-	-	1	-	-	-	-	-	-
T_G4R4	-	-	-	-	-	1	-	-	-	-	-	-
T_G8B8G8R8	-	-	-	-	-	1	-	-	-	-	-	-
T_B8G8R8G8	-	-	-	-	-	1	-	-	-	-	-	-
T_B5G6R5	-	-	-	-	-	1	-	1	1	-	-	-

Supported Pixel Format (T_* convention - MSB to LSB if no "_" notation)*	NVDisplay	NVDEC	NVENC	NVOFA	NVJPG	NVIDIA Ampere Architecture Tegra Tex	NVIDIA Ampere Architecture Tegra CROP	VIC	VI	ISP	PVA	DLA
T_R5G6B5	1	-	-	-	-	1	1	1	1	-	-	-
T_B6G5R5	-	-	-	-	-	1	-	1	-	-	-	-
T_R5G5B6	-	-	-	-	-	1	-	1	-	-	-	-
T_A1B5G5R5	-	-	-	-	-	1	-	1	1	-	-	-
T_A1R5G5B5	1	-	-	-	-	1	1	1	1	-	-	-
T_B5G5R5A1	-	-	-	-	-	1	-	1	1	-	-	-
T_R5G5B5A1	1	-	-	-	-	1	-	1	1	-	-	-
T_A5B5G5R1	-	-	-	-	-	1	-	1	-	-	-	-
T_A5R1G5B5	-	-	-	-	-	1	-	1	-	-	-	-
T_B5G5R1A5	-	-	-	-	-	1	-	1	-	-	-	-
T_R1G5B5A5	-	-	-	-	-	1	-	1	-	-	-	-
T_X1B5G5R5	-	-	-	-	-	1	-	1	-	-	-	-
T_X1R5G5B5	-	-	-	-	-	1	1	1	-	-	-	-
T_B5G5R5X1	-	-	-	-	-	1	-	1	-	-	-	-
T_R5G5B5X1	-	-	-	-	-	1	-	1	-	-	-	-
T_A4B4G4R4	-	-	-	-	-	1	-	1	1	-	-	-
T_A4R4G4B4	-	-	-	-	-	1	-	1	1	-	-	-
T_B4G4R4A4	-	-	-	-	-	1	-	1	1	-	-	-
T_R4G4B4A4	1	-	-	-	-	1	-	1	1	-	-	-
T_R8G8B8	-	-	-	-	-	-	-	-	-	-	-	-
T_B8G8R8	-	-	-	-	-	-	-	-	-	-	-	-
T_A8B8G8R8	1	-	-	-	Decoder:1 Encoder: -	1	1	1	1	1	1	1

Supported Pixel Format (T_* convention - MSB to LSB if no "_" notation)*	NVDisplay	NVDEC	NVENC	NVOFA	NVJPG	NVIDIA Ampere Architecture Tegra Tex	NVIDIA Ampere Architecture Tegra CROP	VIC	VI	ISP	PVA	DLA
T_A8R8G8B8	1	-	-	-	Decoder:1 Encoder: -	1	1	1	1	1	1	1
T_B8G8R8A8	-	-	-	-	Decoder:1 Encoder: -	1	-	1	1	1	1	1
T_R8G8B8A8	-	-	-	-	Decoder:1 Encoder: -	1	-	1	1	1	1	1
T_X8B8G8R8	1	-	-	-	Decoder:1 Encoder: -	1	1	1	-	-	1	1
T_X8R8G8B8	1	-	-	-	Decoder:1 Encoder: -	1	1	1	-	-	1	1
T_B8G8R8X8	-	-	-	-	Decoder:1 Encoder: -	1	-	1	-	-	1	1
T_R8G8B8X8	-	-	-	-	Decoder:1 Encoder: -	1	-	1	-	-	1	1
T_A2B10G10R10	1	-	-	-	-	1	1	1	1	1	1	-
T_A2R10G10B10	1	-	-	-	-	1	1	1	1	1	1	-
T_B10G10R10A2	-	-	-	-	-	1	-	1	1	1	1	-
T_R10G10B10A2	-	-	-	-	-	1	-	1	1	1	1	-
T_A2Y10U10V10	-	-	-	-	-	-	-	-	-	1	-	-
T_V10U10Y10A2	-	-	-	-	-	-	-	-	-	1	-	-
T_A4P4	-	-	-	-	-	-	-	1	-	-	-	-
T_A8P8	-	-	-	-	-	-	-	1	-	-	-	-

Supported Pixel Format (T_* convention - MSB to LSB if no "_" notation)*	NVDisplay	NVDEC	NVENC	NVOFA	NVJPG	NVIDIA Ampere Architecture Tegra Tex	NVIDIA Ampere Architecture Tegra CROP	VIC	VI	ISP	PVA	DLA
T_P8	-	-	-	-	-	-	-	1	-	-	-	-
T_P4	-	-	-	-	-	-	-	-	-	-	-	-
T_P2	-	-	-	-	-	-	-	-	-	-	-	-
T_P1	-	-	-	-	-	-	-	-	-	-	-	-
T_U8V8	-	-	-	-	-	-	-	1	-	-	1	-
T_V8U8	-	-	-	-	-	-	-	1	-	-	1	-
T_A8Y8U8V8 (AYUV444 packed)	-	-	-	-	-	-	-	1	-	1	1	1
T_V8U8Y8A8 (AYUV444 packed)	-	-	-	-	-	-	-	1	-	1	1	1
T_V8_U8_Y8 (YUV444 packed)	-	-	-	-	-	-	-	-	-	-	-	-
T_U8_V8_Y8 (YUV444 packed)	-	-	-	-	-	-	-	-	-	-	-	-
T_V8_Y8_U8 (YUV444 packed)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y8_V8_U8 (YUV444 packed)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y8_U8_V8 (YUV444 packed)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y8_U8__Y8_V8 (YUV422 packed)	1	-	-	-	Decoder:1 Encoder: -	-	-	1	1	1	-	-
T_Y8_V8__Y8_U8 (YUV422 packed)	-	-	-	-	-	-	-	1	1	1	-	-
T_U8_Y8__V8_Y8 (YUV422 packed)	1	-	-	-	-	-	-	1	1	1	-	-
T_V8_Y8__U8_Y8 (YUV422 packed)	-	-	-	-	-	-	-	1	1	1	-	-
T_A8Y8U8V8_TRUE (AYUV444 packed)	-	-	-	-	-	-	-	-	-	-	-	-
T_V8U8Y8A8_TRUE (AYUV444 packed)	-	-	-	-	-	-	-	-	-	-	-	-
T_V8_U8_Y8_TRUE (YUV444 packed)	-	-	-	-	-	-	-	-	-	-	-	-
T_U8_V8_Y8_TRUE (YUV444 packed)	-	-	-	-	-	-	-	-	-	-	-	-
T_V8_Y8_U8_TRUE (YUV444 packed)	-	-	-	-	-	-	-	-	-	-	-	-

Supported Pixel Format (T_* convention - MSB to LSB if no "_" notation)*	NVDisplay	NVDEC	NVENC	NVOFA	NVJPG	NVIDIA Ampere Architecture Tegra Tex	NVIDIA Ampere Architecture Tegra CROP	VIC	VI	ISP	PVA	DLA
T_V8_Y8_U8_TRUE (YUV444 packed)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y8_V8_U8_TRUE (YUV444 packed)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y8_U8_V8_TRUE (YUV444 packed)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y8_U8__Y8_V8_TRUE (YUV422 packed)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y8_V8__Y8_U8_TRUE (YUV422 packed)	-	-	-	-	-	-	-	-	-	-	-	-
T_U8_Y8__V8_Y8_TRUE (YUV422 packed)	-	-	-	-	-	-	-	-	-	-	-	-
T_V8_Y8__U8_Y8_TRUE (YUV422 packed)	-	-	-	-	-	-	-	-	-	-	-	-
8-bit Multi-Plane Surface Formats												
T_Y8___U8V8_N444 (YUV444 semiplanar)	-	-	-	1	-	-	-	1	-	1	1	1
T_Y8___V8U8_N444 (YUV444 semiplanar)	1	1	1	1	-	-	-	1	-	1	1	1
T_Y8___U8V8_N422 (YUV422 semiplanar)	-	-	-	1	-	-	-	1	1	-	-	-
T_Y8___V8U8_N422 (YUV422 semiplanar)	1	-	-	1	-	-	-	1	1	-	-	-
T_Y8___U8V8_N422R (YUV422R semiplanar)	-	-	-	1	-	-	-	1	-	-	-	-
T_Y8___V8U8_N422R (YUV422R semiplanar)	-	-	-	1	-	-	-	1	-	-	-	-
T_Y8___U8V8_N420 (YUV420 semiplanar)	-	-	-	1	-	-	-	1	1	1	1	-
T_Y8___V8U8_N420 (YUV420 semiplanar)	1	1	1	1	Decoder:1 Encoder: -	-	-	1	1	1	1	-
T_Y8___U8___V8_N444 (YUV444 planar)	1	-	-	1	Decoder:1 Encoder: -	-	-	1	-	-	-	-

Supported Pixel Format (T_* convention - MSB to LSB if no "_" notation)*	NVDisplay	NVDEC	NVENC	NVOFA	NVJPG	NVIDIA Ampere Architecture Tegra Tex	NVIDIA Ampere Architecture Tegra CROP	VIC	VI	ISP	PVA	DLA
T_Y8___U8___V8_N422 (YUV422 planar)	-	-	-	1	Decoder:1 Encoder: -	-	-	1	1	-	-	-
T_Y8___U8___V8_N422R (YUV422R planar)	-	-	-	1	Decoder:1 Encoder: -	-	-	1	-	-	-	-
T_Y8___U8___V8_N420 (YUV420 planar)	1	-	-	1	Decoder:1 Encoder: -	-	-	1	1	-	-	-
T_Y8___U8V8_N444_TRUE (YUV444 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y8___V8U8_N444_TRUE (YUV444 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y8___U8V8_N422_TRUE (YUV422 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y8___V8U8_N422_TRUE (YUV422 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y8___U8V8_N422R_TRUE (YUV422R semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y8___V8U8_N422R_TRUE (YUV422R semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y8___U8V8_N420_TRUE (YUV420 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y8___V8U8_N420_TRUE (YUV420 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y8___U8___V8_N444_TRUE (YUV444 planar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y8___U8___V8_N422_TRUE (YUV422 planar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y8___U8___V8_N422R_TRUE (YUV422R planar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y8___U8___V8_N420_TRUE (YUV420 planar)	-	-	-	-	-	-	-	-	-	-	-	-

Supported Pixel Format (T_* convention - MSB to LSB if no "_" notation)*	NVDisplay	NVDEC	NVENC	NVOFA	NVJPG	NVIDIA Ampere Architecture Tegra Tex	NVIDIA Ampere Architecture Tegra CROP	VIC	VI	ISP	PVA	DLA
10-bit Multi-Plane Surface Formats		-	-	-								
T_Y10___U10V10_N444 (YUV444 semiplanar)	-	-	-	1	-	-	-	1	-	-	-	1
T_Y10___V10U10_N444 (YUV444 semiplanar)	1	1	1	1	-	-	-	1	-	-	-	1
T_Y10___U10V10_N422 (YUV422 semiplanar)	-	-	-	1	-	-	-	1	1	-	-	-
T_Y10___V10U10_N422 (YUV422 semiplanar)	1	-	-	1	-	-	-	1	1	-	-	-
T_Y10___U10V10_N422R (YUV422R semiplanar)	-	-	-	1	-	-	-	1	-	-	-	-
T_Y10___V10U10_N422R (YUV422R semiplanar)	-	-	-	1	-	-	-	1	-	-	-	-
T_Y10___U10V10_N420 (YUV420 semiplanar)	-	-	-	1	-	-	-	1	1	-	-	-
T_Y10___V10U10_N420 (YUV420 semiplanar)	1	1	1	1	-	-	-	1	1	-	-	-
T_Y10___U10___V10_N444 (YUV444 planar)	-	-	-	1	-	-	-	1	-	-	-	-
T_Y10___U10___V10_N422 (YUV422 planar)	-	-	-	1	-	-	-	1	1	-	-	-
T_Y10___U10___V10_N422R (YUV422R planar)	-	-	-	1	-	-	-	1	-	-	-	-
T_Y10___U10___V10_N420 (YUV420 planar)	-	-	-	1	-	-	-	1	1	-	-	-
T_Y10___U10V10_N444_TRUE (YUV444 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y10___V10U10_N444_TRUE (YUV444 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y10___U10V10_N422_TRUE (YUV422 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y10___V10U10_N422_TRUE (YUV422 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-

Supported Pixel Format (T_* convention - MSB to LSB if no "_" notation)*	NVDisplay	NVDEC	NVENC	NVOFA	NVJPG	NVIDIA Ampere Architecture Tegra Tex	NVIDIA Ampere Architecture Tegra CROP	VIC	VI	ISP	PVA	DLA
T_Y10___U10V10_N422R_TRUE (YUV422R semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y10___V10U10_N422R_TRUE (YUV422R semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y10___U10V10_N420_TRUE (YUV420 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y10___V10U10_N420_TRUE (YUV420 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y10___U10___V10_N444_TRUE (YUV444 planar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y10___U10___V10_N422_TRUE (YUV422 planar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y10___U10___V10_N422R_TRUE (YUV422R planar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y10___U10___V10_N420_TRUE (YUV420 planar)	-	-	-	-	-	-	-	-	-	-	-	-
12-bit Multi-Plane Surface Formats		-	-	-								
T_Y12___U12V12_N444 (YUV444 semiplanar)	-	-	-	1	-	-	-	1	-	-	-	1
T_Y12___V12U12_N444 (YUV444 semiplanar)	1	1	-	1	-	-	-	1	-	-	-	1
T_Y12___U12V12_N422 (YUV422 semiplanar)	-	-	-	1	-	-	-	1	-	-	-	-
T_Y12___V12U12_N422 (YUV422 semiplanar)	1	-	-	1	-	-	-	1	-	-	-	-
T_Y12___U12V12_N422R (YUV422R semiplanar)	-	-	-	1	-	-	-	1	-	-	-	-
T_Y12___V12U12_N422R (YUV422R semiplanar)	-	-	-	1	-	-	-	1	-	-	-	-
T_Y12___U12V12_N420 (YUV420 semiplanar)	-	-	-	1	-	-	-	1	-	-	-	-
T_Y12___V12U12_N420 (YUV420 semiplanar)	1	1	1	1	-	-	-	1	-	-	-	-

Supported Pixel Format (T_* convention - MSB to LSB if no "_" notation)*	NVDisplay	NVDEC	NVENC	NVOFA	NVJPG	NVIDIA Ampere Architecture Tegra Tex	NVIDIA Ampere Architecture Tegra CROP	VIC	VI	ISP	PVA	DLA
T_Y12___U12___V12_N444 (YUV444 planar)	-	-	-	1	-	-	-	1	-	-	-	-
T_Y12___U12___V12_N422 (YUV422 planar)	-	-	-	1	-	-	-	1	-	-	-	-
T_Y12___U12___V12_N422R (YUV422R planar)	-	-	-	1	-	-	-	1	-	-	-	-
T_Y12___U12___V12_N420 (YUV420 planar)	-	-	-	1	-	-	-	1	-	-	-	-
T_Y12___U12V12_N444_TRUE (YUV444 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y12___U12V12_N422_TRUE (YUV422 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y12___U12V12_N422R_TRUE (YUV422R semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y12___U12V12_N420_TRUE (YUV420 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y12___U12___V12_N444_TRUE (YUV444 planar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y12___U12___V12_N422_TRUE (YUV422 planar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y12___U12___V12_N422R_TRUE (YUV422R planar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y12___U12___V12_N420_TRUE (YUV420 planar)	-	-	-	-	-	-	-	-	-	-	-	-
16-bit Multi-Plane Surface Formats		-	-	-								
T_Y16___U16V16_N444 (YUV444 semiplanar)	-	-	-	1	-	-	-	-	-	1	1	1
T_Y16___V16U16_N444 (YUV444 semiplanar)	-	-	-	1	-	-	-	1	-	1	1	1
T_Y16___U16V16_N422 (YUV422 semiplanar)	-	-	-	1	-	-	-	-	-	-	-	-
T_Y16___V16U16_N422 (YUV422 semiplanar)	-	-	-	1	-	-	-	1	-	-	-	-

Supported Pixel Format (T_* convention - MSB to LSB if no "_" notation)*	NVDisplay	NVDEC	NVENC	NVOFA	NVJPG	NVIDIA Ampere Architecture Tegra Tex	NVIDIA Ampere Architecture Tegra CROP	VIC	VI	ISP	PVA	DLA
T_Y16___U16V16_N422R (YUV422R semiplanar)	-	-	-	1	-	-	-	-	-	-	-	-
T_Y16___V16U16_N422R (YUV422R semiplanar)	-	-	-	1	-	-	-	-	-	-	-	-
T_Y16___U16V16_N420 (YUV420 semiplanar)	-	-	-	1	-	-	-	-	-	1	1	-
T_Y16___V16U16_N420 (YUV420 semiplanar)	-	-	-	1	-	-	-	1	-	1	1	-
T_Y16___U16___V16_N444 (YUV444 planar)	-	-	-	1	-	-	-	-	-	-	-	-
T_Y16___U16___V16_N422 (YUV422 planar)	-	-	-	1	-	-	-	-	-	-	-	-
T_Y16___U16___V16_N422R (YUV422R planar)	-	-	-	1	-	-	-	-	-	-	-	-
T_Y16___U16___V16_N420 (YUV420 planar)	-	-	-	1	-	-	-	-	-	-	-	-
T_Y16___U16V16_N444_TRUE (YUV444 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16___V16U16_N444_TRUE (YUV444 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16___U16V16_N422_TRUE (YUV422 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16___V16U16_N422_TRUE (YUV422 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16___U16V16_N422R_TRUE (YUV422R semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16___V16U16_N422R_TRUE (YUV422R semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16___U16V16_N420_TRUE (YUV420 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16___V16U16_N420_TRUE (YUV420 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16___U16___V16_N444_TRUE (YUV444 planar)	-	-	-	-	-	-	-	-	-	-	-	-

Supported Pixel Format (T_* convention - MSB to LSB if no "_" notation)*	NVDisplay	NVDEC	NVENC	NVOFA	NVJPG	NVIDIA Ampere Architecture Tegra Tex	NVIDIA Ampere Architecture Tegra CROP	VIC	VI	ISP	PVA	DLA
T_Y16___U16___V16_N422_TRUE (YUV422 planar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16___U16___V16_N422R_TRUE (YUV422R planar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16___U16___V16_N420_TRUE (YUV420 planar)	-	-	-	-	-	-	-	-	-	-	-	-
16-bit Y/8-bit UV Multi-Plane Surface Formats		-	-	-								
T_Y16___U8V8_N444 (YUV444 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16___V8U8_N444 (YUV444 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16___U8V8_N422 (YUV422 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16___V8U8_N422 (YUV422 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16___U8V8_N422R (YUV422R semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16___V8U8_N422R (YUV422R semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16___U8V8_N420 (YUV420 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16___V8U8_N420 (YUV420 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16I___U8V8_N420 (YUV420 semiplanar)	-	-	-	-	-	-	-	-	-	1	-	-
T_Y16I___V8U8_N420 (YUV420 semiplanar)	-	-	-	-	-	-	-	-	-	1	-	-
T_Y16F___U8V8_N420 (YUV420 semiplanar)	-	-	-	-	-	-	-	-	-	1	-	-
T_Y16F___V8U8_N420 (YUV420 semiplanar)	-	-	-	-	-	-	-	-	-	1	-	-
T_Y16___U8___V8_N422R (YUV422R planar)	-	-	-	-	-	-	-	-	-	-	-	-

Supported Pixel Format (T_* convention - MSB to LSB if no "_" notation)*	NVDisplay	NVDEC	NVENC	NVOFA	NVJPG	NVIDIA Ampere Architecture Tegra Tex	NVIDIA Ampere Architecture Tegra CROP	VIC	VI	ISP	PVA	DLA
T_Y16___U8___V8_N420 (YUV420 planar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16___U8V8_N444_TRUE (YUV444 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16___V8U8_N444_TRUE (YUV444 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16___U8V8_N422_TRUE (YUV422 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16___V8U8_N422_TRUE (YUV422 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16___U8V8_N422R_TRUE (YUV422R semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16___V8U8_N422R_TRUE (YUV422R semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16___U8V8_N420_TRUE (YUV420 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16___V8U8_N420_TRUE (YUV420 semiplanar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16___U8___V8_N444_TRUE (YUV444 planar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16___U8___V8_N422_TRUE (YUV422 planar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16___U8___V8_N422R_TRUE (YUV422R planar)	-	-	-	-	-	-	-	-	-	-	-	-
T_Y16___U8___V8_N420_TRUE (YUV420 planar)	-	-	-	-	-	-	-	-	-	-	-	-

7.9.2 Programming Guidelines

7.9.2.1 Software Guidelines

Software should allocate surfaces respecting the minimum alignments specified in the Tiling Formats section. This section gives additional alignment restrictions for certain clients and advice on allocating surfaces for best performance. Certain clients, such as NVENC, NVDEC, Display, and VIC use 32-bit registers (left-shifted 8) to specify 40-bit base addresses of a buffer. Base addresses for these clients must be aligned to a minimum of 256 B. Certain clients, such as NVENC and VIC have a fixed cache line size and read all data within a cache line, even if it is beyond the nominal height and width of the surface. For such clients, surface width and height need to be padded to at least the next multiple of the cache line size so the unit does not read beyond the end of the buffer.

7.9.2.1.1 Surface Type

- Block linear format generally provides best performance and should be chosen by default.
- Pitch format may be preferable if producer and consumer clients perform linear accesses and both clients support pitch.
- Private formats are used by a few units, such as VI, ISP, NVENC, and NVDEC. For private formats, the client-required allocation and padding rules must be followed.

7.9.2.1.2 Block Linear

- Surfaces to be accessed by the ga10b texture unit may require a smaller block height under certain conditions.
- Hardware allocates a block linear surface to be an integral number of blocks wide, so its allocated width is width*Bpp rounded up to the next block boundary (64B), which is sufficient padding in almost all circumstances.
- Surfaces should be an integral number of blocks tall. If a client such as NVENC or VIC has a cache line that is taller than the block height, additional row(s) of blocks may be required for padding.

Note 1: For buffers that are to be accessed using the gv11b texture unit, the BlockHeight parameter must match that used by the texture unit. In particular, note that if a texture surface is smaller than the specified size in blocks, texture uses a shrunken BlockHeight instead of the nominal parameter specified in the Texture Header.

Note 2: If VIC cacheline shapes of 128x2 or 256x1 are used (generally not recommended for block linear), width must be padded up to a multiple of the cache line width. If this is not possible, a narrower cache line shape must be used.

7.9.2.1.3 Pitch

- For clients, such as NVENC and VIC, additional lines of padding may be required beyond the nominal height of the surface. For example, when VIC is reading a source buffer with an internal cache line shape of 32x8, the source buffer must be a multiple of eight lines tall.

7.9.2.2 Block Height Limitations

NVENC and OFA support only a subset of the block heights. However, in general all hardware blocks support full range of block heights (that is, 2/4/8/16/32). Refer to the following table of programmable block height support.

Block	Surface	Block Heights Supported
ISP	Any	All (optimal performance BlockHeight=2)
NVDEC	Any	All except 1 (optimal performance based on Coding Tree Block size)
NVENC	Input/Reference	All (optimal performance based on Coding Tree Block size)
	Recon	BlockHeight = 2 only
OFA	Input/Reference	All except 1 (optimal performance BlockHeight=2)
	Output Flow/Disparity	All except 1
VIC	Input	All (optimal performance BlockHeight=16)
	Output	All (optimal performance BH=32 normal/BH=2 rotated)
Display	Horizontal	All (optimal performance BlockHeight=2)
	Vertical	All (optimal performance BlockHeight=16)
PVA	Any	All (optimal performance BlockHeight=2)
NVDLA	Input	All (optimal performance BlockHeight=2)

Beyond limitations of multimedia blocks performance with respect to block height, the SMMU has possibly a larger performance penalty for block heights that are outside of the SMMU block height sweet spot.

A summary of block height recommendations:

- BlockHeight = 16 GOBs (128 lines) is the recommended setting for most clients that use block linear. It provides decent accesses per activate for clients that traverse a buffer horizontally, vertically, or in tiles. DRAM pages are 64B × 64 lines. TLB cache lines are roughly square in screen space. This should be the default setting and is the only performant block height for rotated display.

- BlockHeight = 1 GOB (eight lines) is optimal for clients that make horizontal accesses, doubling access per activate, but at the expense of clients that make vertical accesses. This is the optimal setting for nonrotated display, but other block heights are okay for nonrotated display.
- BlockHeight = 2 GOBs (16 lines) is the hard-coded value for NVENC video engines RECON surface. It does not provide the access per activate benefit of BlockHeight = 1 and TLB cache lines have a poor aspect ratio. Buffers used by this client must have block height = 2.

8. System Components

8.1 Clock and Reset Controller (CAR)

8.1.1 Overview

The Clock and Reset Controller (CAR) block contains all the logic needed to control most of the clocks and resets to the System-on-Chip (SoC). The CAR block provides the registers to program the PLLs and controls most of the clock source programming and clock dividers.

> For Orin, clocks and resets are controlled by NVIDIA provided firmware running on BPMP. The hardware interface to the clock and reset controls is not documented, and the software documentation should be consulted for details of how these functions are controlled.

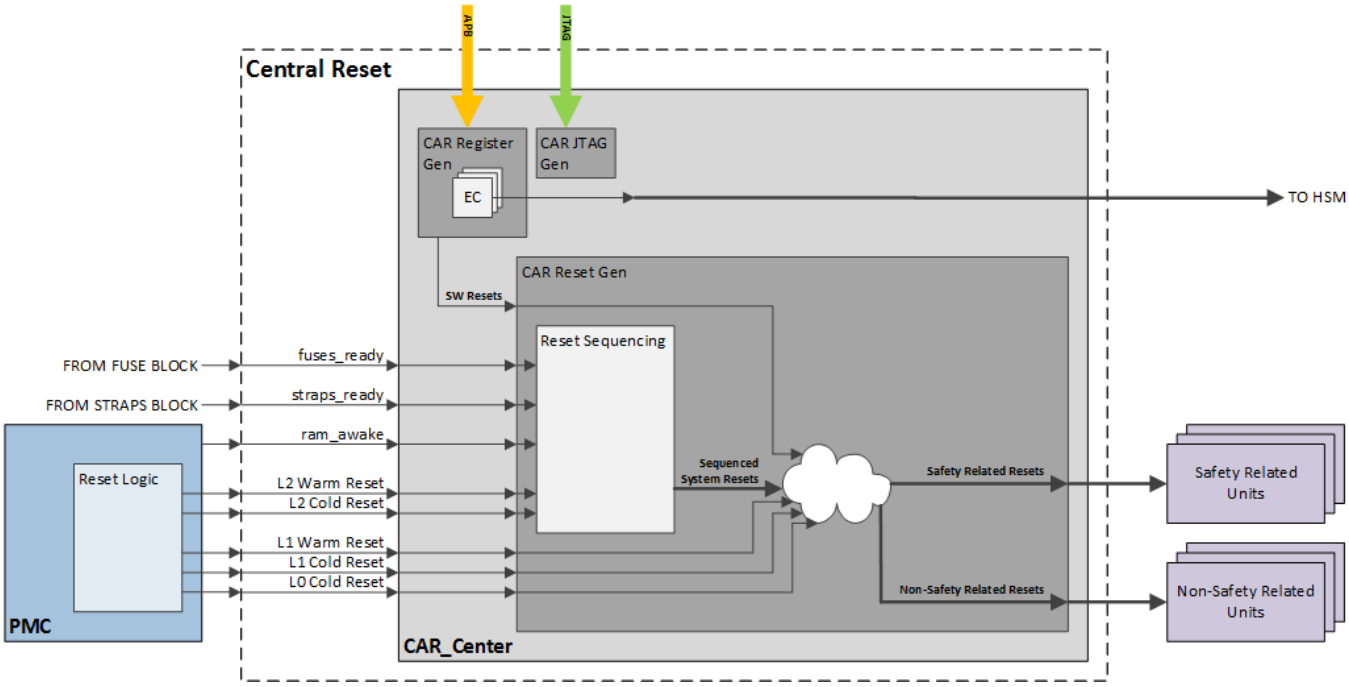
8.1.2 Reset Controller

This section describes the reset architecture for the System-on-Chip (SoC), including the reset-associated logic and the system-level reset domains. Asserting any of these domains results in a reset of the majority of VDD_CORE, CV, CCPLEX, and GPU rail logic and, potentially, some or all of the VDD_RTC logic.

This reset architecture spans multiple modules, such as PMC, CAR, Timers, and BPMP. It also includes the requirements of the downstream units. This chapter also includes details of unit-level reset generation from these system-level resets and unit-level reset sources, such as software resets and resets for various power-management states.

The high-level block diagram for the reset generation logic is shown in the figure below. The reset generation logic is contained both in PMC and CAR and is a subset of those units.

Figure 8.1 High Level Reset Diagram



The reset logic has one main input from the external PMIC and one main output back to the external PMIC.

SYS_RESET_N is the reset input to the SoC from the external PMIC. This reset is asserted when all of the pre-boot core logic rails from the PMIC are enabled and stable and de-asserted when the pre-boot I/O and PLL rails from the PMIC, the 32-kHz clock, and reference clocks are enabled and stable. This is the only reset source that asserts all resets from the PMC and CAR. Some register state can persist through the other internal reset sources. SYS_RESET_N is distributed to the PMC for further system reset generation.

Also available on this interface are the watch-dog reset WDT_RESET_OUT* pads. These are sideband reset request outputs for PMICs that support such a feature. These pads are pin-muxed and are only used one at a time. By asserting one of these active low outputs (or sending an I^2C SFT_RST request for PMICs that do not support this sideband input), it results in the assertion of the SYS_RESET_N to the Orin.

8.1.3 Clock Controller

The clocking logic makes use of the following to provide the various clocks in the system, including the frequency of operation, clock quality, spread, and clock gating.

- Sources capable of generating frequencies independently or based on a reference clock – External oscillators, Phase-Locked Loops (PLLs), and Noise-Aware Frequency Locked Loops (NAFLLs).
- Clock manipulation elements – switches, dividers, gates, skippers, trimmers, shapers, etc.

The controls for these elements are present in the clock controller module. These controls may need to be sequenced in an appropriate manner to ensure that there are no glitches or other clock artifacts in the system.

Key features include the following:

- Clock generation for all functional blocks.
- Register parity for clock configuration registers.
- LUT RAM parity for certain NAFLL LUTs.
- Clock Monitoring for certain clocks.

8.2 Interrupt Controllers

8.2.1 Overview

This chapter discusses the Interrupt Controllers, with focus on the mapping of all Interrupts and the architecture for routing and handling of these Interrupts.

Orin supports a large number of Hardware interrupts, routed between sources and processors via interrupt controllers. From the perspective of Interrupts, devices like the GPU, Memory Controller, Video Encode/Decode Engines, functional blocks/clusters, and various I/O modules are the sources of Interrupts; and processors are the sinks of Interrupts. From the Interrupt source, Interrupts go to the Interrupt controllers, receive prioritization, then get routed to the appropriate target processor according to the software-defined configuration.

Multiple types of Interrupt controllers are used: four of these interrupt controllers are Arm® IP related (GIC-600AE, GIC 400, Cortex-R52 internal GIC, and AVIC), while one is designed by NVIDIA (LIC).

Masking of the Interrupts are always done at the source of Interrupts whereas removal of Interrupts is performed at end of the appropriate Interrupt Service Routine (ISR) running on the target processor. This chapter does not discuss the details related to the masking and removal of Interrupts.

Note: For masking (enabling and disabling) of Interrupts, refer to the individual Arm architecture documents.

8.2.1.1 List of References

This chapter makes the implicit use of the following document available from Arm® based on the assumption that readers are familiar with the Arm Architecture and have access to the documents for reference. Refer to the Arm website to download the document.

- *Arm® CoreLink™ GIC-600AE Generic Interrupt Controller Technical Reference Manual*
  https://developer.arm.com/documentation/101206/0002
- *Arm CoreLink GIC-400 Generic Interrupt controller Technical Reference Manual (version r0p1)*
  http://infocenter.arm.com/help/help/index.jsp?topic=/com.arm.doc.ddi0471b
- *Arm PrimeCell Vectored Interrupt controller (PL192) Technical Reference Manual*
  http://infocenter.arm.com/help/topic/com.arm.doc.ddi0273a/DDI0273.pdf
- *Arm Architecture Reference Manual Armv8*
  http://d eveloper.arm.com/docs/ddi0487/latest/arm-architecture-reference-manual-armv8-for-armv8-a-architecture-profile

Note: Arm refers to the PL192 as VIC, but it is always termed as AVIC in this document to avoid confusion with the Video Image Compositor (VIC) block.

8.2.1.2 Glossary

Note that different names may end up in the same acronym when the entire TRM is put in perspective. The acronyms listed here are within the context of the Interrupt controllers chapter.

Term	Definition
AO	Always On
APB	Advanced Peripheral Bus
AVIC	Arm Vectored Interrupt Controller an Arm IP block used with Arm Cortex Rx processors.
BL	Boot Loader
BPMP	Boot and Power Management Processor
BR	Boot ROM
CAR	Clock and Reset
FIQ	Fast Interrupt Request, signaled using the active low nFIQ
FSI	Functional Safety Island
IPI	Inter Process Interrupt
IRQ	Interrupt Request, signaled using the active low nIRQ
ISR	Interrupt Service Routine or Interrupt Status Register
LIC	Legacy Interrupt Controller

Term	Definition
PPI	Private Peripheral Interrupt
R5	Cortex-R5 Processor
SGI	Software Generated Interrupt
SPI	Shared Peripheral Interrupt
vGIC	Virtualization capable GIC
VFIQ	Virtual FIQ, coded active low on nVFIQ
VIRQ	Virtual IRQ, coded active low on nIRQ
VREI	Virtual REI, coded active low on nVREI
WFE	Wait For Event
WFI	Wait For Interrupt

8.2.1.3 Relevant Chapters in the TRM

- Address Map
- Always-On (AON) Cluster and SPE
- Audio Processing Engine (APE)
- Boot and Power Management Processor (BPMP)
- Clock Controller and Reset (CAR)
- CPU Complex (CCPLEX)
- Real-time Camera Engine (RCE)
- Safety Cluster Engine (SCE)
- Display Control Engine (DCE)
- Functional Safety Island (FSI)

8.2.1.4 Specifics in Reading This Chapter

There are eight UART units in the SoC. These eight units are dubbed with either an alphabetical suffix (A, B, C, D, E, F, G, H) or a numerical suffix (1, 2, 3, 4, 5, 6, 7, 8) throughout its design specification and documentation. For consistency and the sake of eliminating unnecessary confusion, this chapter uses only the alphabetical suffix for these UART units, that is, UARTA, UARTB, UARTC, UARTD, UARTE, UARTF, UARTG, UARTH (which may be referred to as UART1, UART2, UART3, UART4, UART5, UART6, UART7, and UART8 in other specification and documentation including other parts of this TRM).

8.2.2 Functional Description

8.2.2.1 Interrupt Handling Mechanism

There are different types of Interrupt controllers which receive Interrupts from devices (that is, hardware Interrupts) or processors (that is, software Interrupts including IPI). These different Interrupts are arbitrated then sent to appropriate target processor(s).

From the perspective of Interrupts, there are different types of processors:

- The different CPU cores in the CCPLEX, receiving their interrupt from a GIC, an instance of the GIC-600AE.
- The Cortex-A9 core in the APE, receiving its interrupt from a GIC, an instance of the GIC-400, the GIC part of the SCU logic remains in bypass mode all the time and is not further discussed.
- The Cortex-R5 core in each of RCE, SPE, BPMP, DCE, SCE, and FSI CHSM receives its interrupt from an AVIC, made of two chained instances of a PL192 IP block.
- The Cortex-R52 cores in FSI SFTY_CPU receives its interrupts from a GIC internal to the Cortex-R52 processor.

There is distinction between shared and local Interrupts as well. Local interrupts are only connected to the local interrupt cluster. These local interrupts have strong affinity with their associated processor and are not directly visible outside of the processor cluster.

Shared interrupts are of different types:

- Peripherals outside of the processor clusters except for PCIe MSI and the TOP TKE WDT are always considered shared interrupts and connected to the LIC.
- PCIe MSI interrupts are sent to the CCPLEX GIC.
- A peripheral local to a processor cluster may broadcast an interrupt to both the local interrupt controller and to the LIC. This happens when the peripheral is preferentially managed by the local processor but could be managed by a different processor in other use cases.
- The APE uses a GIC-400 with extra channels to expose some locally generated interrupts as shared interrupts, possibly aggregated.

CCPLEX vGIC supports 960 SPI interrupts. There are a total of 554 interrupt wires from LIC. Eight of the SPI interrupt wires are CCPLEX local interrupts, taking up eight of the ten local IDs. The other two local IDs are reserved.

MSI interrupts can be allocated to any unused SPI. Interrupt sources are allocated to one or more interrupt signals based on their requirements and documented in the IP Metadata. At chip assembly, the full list of interrupts per interrupt controller is assembled and combined with an

interrupt mapping file per interrupt controller, providing the interrupt ID. Specialized scripts expand the definition into derived files, especially header files.

All Interrupt controllers support a statically configurable number of Interrupts in steps of 32. The attributes of the interrupt controllers are shown in the table below.

Table 8.1 Interrupt Controllers and Their Characteristics

Location	Type	IP	Timestamping	LIC Channels
TOP	LIC	-	Yes	12
CCPLEX	vGIC	GIC-600AE	SPI[1]	0
AON	AVIC	PL-192	Yes	2
BPMP	AVIC	PL-192	Yes	1
SCE	AVIC	PL-192	Yes	2
RCE	AVIC	PL-192	Yes	2
APE	vGIC	GIC-400	No	1
DCE	AVIC	PL-192	Yes	2
FSI	GIC	R52	No	1
FSI	AVIC	PL-192	No	1

1. Timestamping is supported using the LIC timestamp engine. SGIs, PPIs, Local CCPLEX SPIs, and PCIe MSIs are not timestamped.

The LIC channels send aggregated signals to the different clusters, except the CCPLEX, to provide global visibility of top-level interrupts to every processor. The CCPLEX is an exception as it gets all these Interrupts in non-aggregated fashion. The channels of LIC are connected as shown in the table below.

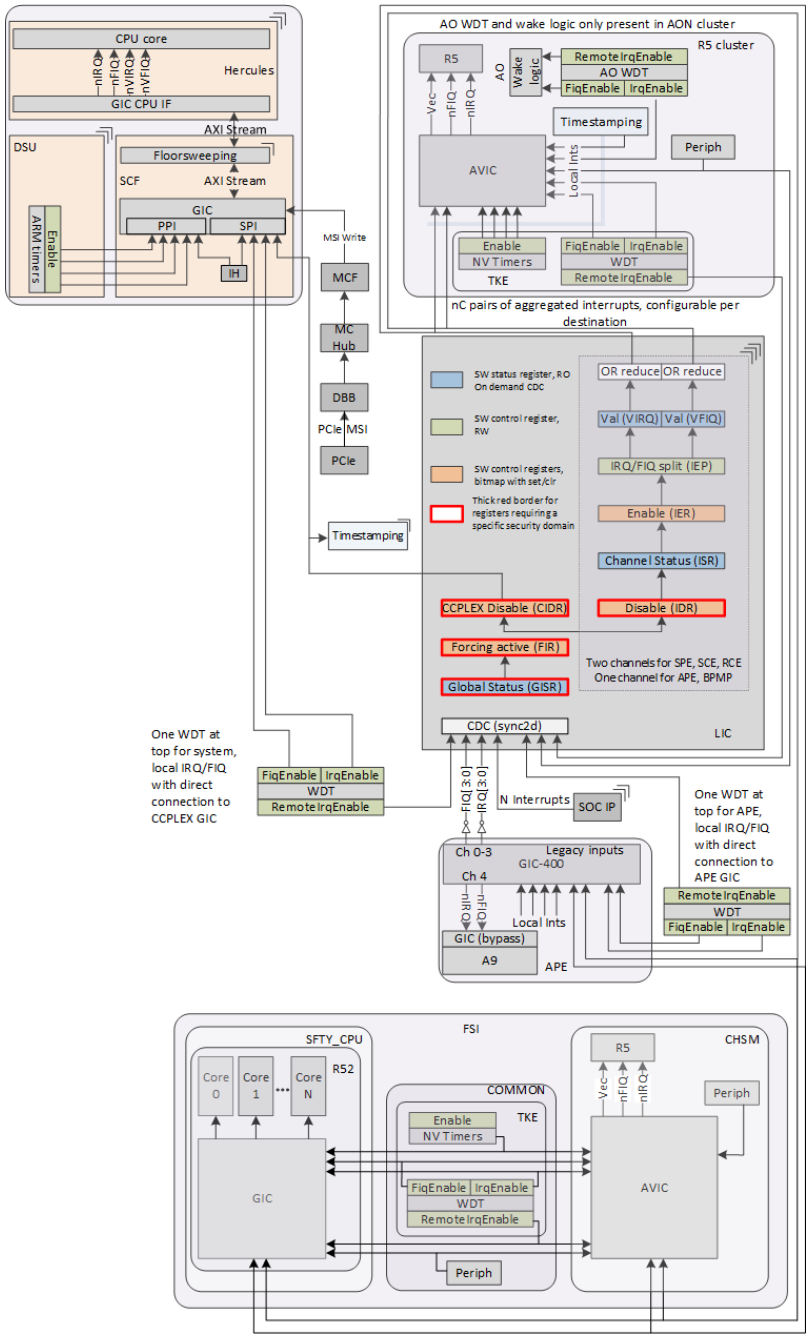
Table 8.2 LIC Channel Mapping

Channel Name	Channel Number	IRQ/FIQ Connected to
BPMP	0	AVIC in BPMP
APE	1	GIC in APE
AON[1:0]	[3:2]	AVIC in AON
SCE[1:0]	[5:4]	AVIC in SCE
RCE[1:0]	[7:6]	AVIC in RCE
DCE[1:0]	[9:8]	AVIC in DCE
FSI[0]	10	GIC in FSI SFTY_CPU

Channel Name	Channel Number	IRQ/FIQ Connected to
FSI[1]	11	AVIC in FSI CHSM

The figure below shows where the different Interrupt controllers are and how Interrupts are routed.

Figure 8.2 Interrupt Routing (Top-Level View)



8.2.2.2 Interrupt Controllers

The Interrupt Controller receives Interrupts from a large number of sources. The Interrupt sources can be assigned a target processor, a type (IRQ versus FIQ), priority levels, and so on, by configuring Interrupt controller registers. The Interrupt controller arbitrates among different incoming

Interrupts and informs the target processor of a pending Interrupt by asserting one or more of the signals described above.

8.2.2.2.1 LIC

The LIC is a simple Interrupt controller performing the following functions:

- Routing all received (352) Interrupt signals through Force Interrupt Register (FIR) for potential forced Interrupts and CCPLEX Interrupt Disable Register (CIDR) for potential disabled Interrupts then to Interrupt signals #0 through #543 of the vGIC in CCPLEX
- Routing shared Interrupts to Cortex-R5 clusters, in aggregated form (that is, via channels)
- Timestamping of shared Interrupts
- Security features

Note: The LIC does not service any of its received Interrupts, as there is no processor inside or associated with the LIC.

The LIC can be extended in steps of 32 Interrupts, each group of 32 is called a slice. The input Interrupt signals are combined with software set/clear bits (per Interrupt). This allows software to be able to set/clear individual Interrupts (provided corresponding Interrupt is not asserted by hardware). The signals are further aggregated using a channel structure, with two output signals per channel using masking and classification stages, followed by a reduce operation.

8.2.2.2.2 vGIC in APE (AGIC)

APE contains one instance of GICv3 (often referred to as vGIC for Virtual Generic Interrupt controller), called AGIC (APE GIC) to support more channels than the built-in GIC in the Cortex-A9 MP. The AGIC is an SMP Interrupt controller with five channels (0 through 4). Specifically, Channel 4 is routed to the APE's Cortex-A9 Interrupt controller (GIC) and Channel 0 through 3 are routed to the LIC.

Note that the Cortex-A9 core does not support virtualization.

8.2.2.2.3 vGIC in CCPLEX

The Interrupt controller used for the CCPLEX cluster is a GICv3 which supports virtualization. The vGIC is an SMP Interrupt controller with eight channels (one per CPU core in the CCPLEX) and receiving Interrupts targeted to (any) one or more of the CPUs in the CCPLEX.

8.2.2.2.4 AVIC in Cortex-R5 Processor Complexes

The AVIC i s an Interrupt controller that supports vectorized and prioritized operation for IRQs, that is, hardware selects the pending Interrupt of the highest priority, then sends an Interrupt request and its associated entry address to the Cortex-R5 CPU core.

8.2.2.2.5 GIC in APE

The Interrupt controller in the Cortex-A9 multiprocessor logic in APE is an instance of the GICv1 (Generic Interrupt controller). The AGIC in APE needs an external vGIC to allow for more channels. See section vGIC in APE (AGIC). This GIC is used in a reduced functionality mode to handle the one channel of legacy IRQ/FIQ from AGIC and timer Interrupts inside the Cortex-A9.

8.2.2.2.6 GIC Interrupt Classes

A GIC (either version 1 or version 2) can support a large number of Interrupts, with each Interrupt identified by its unique ID. A GIC supports different classes of Interrupts. The rest of the discussion only presents the GICv3 (vGIC) operations.

8.2.2.2.7 Software Generated Interrupts

Software Generated Interrupts (SGIs) are software Interrupts which are generated by writing to the Software Generated Interrupt register (GICD_SGIR). Each CPU interface can generate a maximum of 16 SGIs, with ID0-15 for each target processor. The SGIs are also referred to as Inter Processor Interrupts (IPIs).

8.2.2.2.8 Private Peripheral Interrupts

Private Peripheral Interrupts (PPIs) are Interrupts generated by a peripheral that is specific to a single processor.

There are seven PPIs for each CPU processor:

- Virtual Maintenance Interrupt (ID25)
- Hypervisor Timer Interrupt (ID26)
- Virtual Timer Interrupt (ID27)
- Legacy nFIQ (ID28)
- Secure Physical Timer Interrupt (ID29)
- Non-secure Physical Timer Interrupt (ID30)
- Legacy nIRQ (ID31)

Note that the (Virtual) Maintenance Interrupt is not associated with a peripheral, but normally software generated and used as a way to signal an Interrupt targeting explicitly the Hypervisor.

8.2.2.2.9 Shared Peripheral Interrupts

Shared Peripheral Interrupts (SPIs) are external hardware Interrupts generated by asserting signals on GIC or vGIC input pins (called IRQs). From the hardware perspective, these are peripheral Interrupts. All shared Interrupts are first routed to the LIC. SPI IDs start at ID32. The SPIs can be

configured to be edge-triggered or level-sensitive (active-high), the vast majority of Interrupts are level, with some minor exceptions.

8.2.2.3 Interrupt Routing

Every hardware Interrupt is driven by a hardware module within the SoC. Some of those modules may generate a hardware Interrupt in response to an external event (for example, a GPIO assertion, a MSI received at a PCIe root port, and so on). However, these external causes are outside the scope of this TRM chapter.

All of these hardware Interrupts are routed through side-band active-high level-sensitive signals either to LIC (shared Interrupts) or to a local Interrupt controller (that is an Interrupt controller inside a processor cluster). The general routing of Interrupt is illustrated in the figure above.

8.2.2.3.1 Interrupt Handling Targeting a Powered Off Processor

When an Interrupt controller asserts an Interrupt to a processor which is in a less-than-fully-powered state, wake logic associated with the processor is responsible for transitioning the CPU back to a state in which the CPU can handle the Interrupt. This is particularly relevant for processors within the CCPLEX. The exact details of the wake logic are specific to the relevant processor complex.

8.2.2.3.2 LIC Functional Description

The LIC handles a configurable number of Interrupts , using 32-bit slices. The LIC synchronizes the incoming hardware Interrupt requests and combines (bitwise OR) them with the FIR register. The FIR allows software to force the assertion of a particular Interrupt, using set/clear register. The combined software/hardware Interrupt signals are forwarded to the CCPLEX GIC and to the LIC channels.

The rest of the logic is duplicated per channel, with each channel configured via Interrupt Disable (IDR), Interrupt Enable (IER) and Interrupt Class Registers (IEP_CLASS). When a 1 is set in the proper bit position in the IER/IDR register of a channel, that particular source is enabled/disabled for that channel. The class is set in the proper bit position of IEP_CLASS for the corresponding source to be routed as IRQ or FIQ.

The Interrupt status register (ISR) allows the processor to view the state of the pending hardware Interrupt requests regardless of the bit enables programmed in IER. The forced Interrupt status register (FIR) allows the software to selectively force set or clear specific Interrupts. The read-only VIRQ/VFIQ allows the processor to determine the source of the Interrupt request(s) causing the processor to enter the Interrupt service routine.

The individual nIRQ/nFIQ signals generated by all slices are combined (logical AND) in the combiner to generate final nIRQ/nFIQ for the given channel.

The LIC contains two security-related mechanisms:

1. the standard security model for registers, with each channel forming a security register group, and shared registers forming also a security register group.
2. disable bitmaps per channel, in a specific security group, to control the visibility of certain signals across security domains.

8.2.2.3.3 LIC Registers Description

Each channel is split into slices of 32 bits; the set of slices form a wide bitmap, with as many bits as there are Interrupt signals (similar to the way the GIC uses multiple registers to construct wide configuration or status registers). There are also a set of shared registers, that is, outside of any channel. The set of bitmap registers are:

- Shared registers:
  - Forcing active (FIR), with associated Set/Clr access registers.
  - Interrupt status register (G ISR), showing the input to LIC (before FIR and CIDR).
  - CCPLEX Interrupt Disable register (CIDR), allowing to block specific Interrupt in CCPLEX path.
- Per channel registers, for channel <c>:
  - Disable register (IDR<c>), with associated Set/Clr access registers, in a separate security register group.
  - Enable register (IER<c>), with associated Set/Clr access registers.
  - Class register (IEP_CLASS<c>)
  - Valid Interrupt Status Register (VIRQ<c>) and Valid FIQ Status Register (VFIQ<c>) indicate the currently active Interrupts that are valid on the respective pins (nIRQ or nFIQ) for that channel, that is after the enable and class register for that channel.
    - VIRQ<c> = (~IEP_CLASS<c>) & IER<c> & !IDR<c> & (G ISR | FIR)
    - VFIQ<c> = (IEP_CLASS<c>) & IER<c> & !IDR<c> & (G ISR | FIR)

8.2.2.3.4 Interrupt Timestamping

Shared Interrupts (as connected between LIC and CCPLEX GIC) are fed into two generic timestamping engines (two to allow for differentiated security domains).

8.2.2.4 Interrupt Mapping

This section presents the mapping of the Interrupts from system devices to the Interrupt ID's in the different Interrupt controllers.

8.2.2.4.1 Interrupt Mapping by LIC

The following table shows the mapping of the Interrupts to the LIC, which serves only as an aggregator offering no service of any Interrupts.

Table 8.3 LIC Interrupt Mapping

Source Module	Interrupt Number	Interrupt Name	Interrupt Description
TOP TKE	0	TOP_TKE_SHARED_0	Shared timer Interrupt from top TKE (1 of 16 total)
TOP TKE	1	TOP_TKE_SHARED_1	Shared timer Interrupt from top TKE (1 of 16 total)
TOP TKE	2	TOP_TKE_SHARED_2	Shared timer Interrupt from top TKE (1 of 16 total)
TOP TKE	3	TOP_TKE_SHARED_3	Shared timer Interrupt from top TKE (1 of 16 total)
TOP TKE	4	TOP_TKE_SHARED_4	Shared timer Interrupt from top TKE (1 of 16 total)
TOP TKE	5	TOP_TKE_SHARED_5	Shared timer Interrupt from top TKE (1 of 16 total)
TOP TKE	6	TOP_TKE_SHARED_6	Shared timer Interrupt from top TKE (1 of 16 total)
TOP TKE	7	TOP_TKE_SHARED_7	Shared timer Interrupt from top TKE (1 of 16 total)
TOP TKE	8	TOP_TKE_SHARED_8	Shared timer Interrupt from top TKE (1 of 16 total)
TOP TKE	9	TOP_TKE_SHARED_9	Shared timer Interrupt from top TKE (1 of 16 total)
AON - RTC	10	RTC	RTC Interrupt
LIC - GTE	11	LIC_GTE_0	Interrupt reflecting occupancy of FIFO in a Generic Timestamp Engine; here a GTE coupled with LIC
LIC - GTE	12	LIC_GTE_1	Interrupt reflecting occupancy of FIFO in a Generic Timestamp Engine; here a GTE coupled with LIC
AON - GTE	13	AON_GTE	Interrupt reflecting occupancy of FIFO in a Generic Timestamp Engine; here a GTE coupled with the AON AVIC
BPMP - TKE	14	BPMP_WDT_REMOTE	Remote Interrupt associated with a given WDT; associated cluster probably down if risen
AON - TKE	15	SPE_WDT_REMOTE	Remote Interrupt associated with a given WDT; associated cluster probably down if risen
SCE - TKE	16	SCE_WDT_REMOTE	Remote Interrupt associated with a given WDT; associated cluster probably down if risen
TOP TKE	17	TOP_WDT_REMOTE	Remote Interrupt associated with a given WDT; associated cluster probably down if risen
AON - RTC	18	AOWDT_REMOTE	Remote Interrupt associated with a given WDT; this one is the AO WDT with no direct HW association with a specific core (some affinity with SPE in AON)

Source Module	Interrupt Number	Interrupt Name	Interrupt Description
RCE - TKE	19	RCE_WDT_REMOTE	Remote Interrupt associated with a given WDT; associated cluster probably down if risen
APE	20	APE_WDT_REMOTE	Remote Interrupt associated with a given WDT; associated cluster probably down if risen
XUSB - HOST	21	USB3_HOST_VF0	XUSB Interrupt
XUSB - HOST	22	USB3_HOST_VF1	XUSB Interrupt
XUSB - HOST	23	USB3_HOST_VF2	XUSB Interrupt
XUSB - HOST	24	USB3_HOST_VF3	XUSB Interrupt
I2C1	25	I2C1	I2C Interrupt
I2C2 (in AON)	26	I2C2	I2C Interrupt; From AON
I2C3	27	I2C3	I2C Interrupt
I2C4	28	I2C4	I2C Interrupt
I2C5	29	I2C5	I2C Interrupt
I2C6	30	I2C6	I2C Interrupt
I2C7	31	I2C7	I2C Interrupt
I2C8 (in AON)	32	I2C8	I2C Interrupt; From AON
I2C9	33	I2C9	I2C Interrupt
RESERVED	34	RESERVED	NA
QSPI0	35	QSPI0	QSPI Interrupt
SPI1	36	SPI1	SPI Interrupt
SPI2 (in AON)	37	SPI2	SPI Interrupt; From AON
SPI3	38	SPI3	SPI Interrupt
QSPI1	39	QSPI1	QSPI Interrupt
CAN1	40	CAN1_0	CAN Interrupt; in pair with CAN1_1
CAN1	41	CAN1_1	CAN Interrupt; in pair with CAN1_0
CAN2	42	CAN2_0	CAN Interrupt; in pair with CAN2_1

Source Module	Interrupt Number	Interrupt Name	Interrupt Description
CAN2	43	CAN2_1	CAN Interrupt; in pair with CAN2_0
UFSHC	44	UFSHC	UFS Host Controller Interrupt
PCIE	45	PEX0_C1_SYS_0	PCIe Interrupt
PCIE	46	PEX0_C1_SYS_1	PCIe Interrupt
PCIE	47	PEX0_C2_SYS_0	PCIe Interrupt
PCIE	48	PEX0_C2_SYS_1	PCIe Interrupt
PCIE	49	PEX0_C3_SYS_0	PCIe Interrupt
PCIE	50	PEX0_C3_SYS_1	PCIe Interrupt
PCIE	51	PEX0_C4_SYS_0	PCIe Interrupt
PCIE	52	PEX0_C4_SYS_1	PCIe Interrupt
PCIE	53	PEX1_C5_SYS_0	PCIe Interrupt
PCIE	54	PEX1_C5_SYS_1	PCIe Interrupt
RESERVED	55	RESERVED	NA
AON - GPIO	56	AON_GPIO_0	GPIO Interrupt from AON GPIO
AON - GPIO	57	AON_GPIO_1	GPIO Interrupt from AON GPIO
AON - GPIO	58	AON_GPIO_2	GPIO Interrupt from AON GPIO
AON - GPIO	59	AON_GPIO_3	GPIO Interrupt from AON GPIO
HDA	60	AZA	Interrupt from AZA controller
HDA	61	HDACODEC	Interrupt from had codec
SDMMC1	62	SDMMC1	SDMMC Interrupt; in pair with SDMMC1_SYS
SPI4	63	SPI4	SPI Interrupt
SPI5	64	SPI5	SPI Interrupt
SDMMC4	65	SDMMC4	SDMMC Interrupt; in pair with SDMMC4_SYS
SDMMC1	66	SDMMC1_SYS	SDMMC Interrupt; in pair with SDMMC1
GPU	67	GPU_HOST2SOC_0	GPU MSI-X vectors (1 of 4 total)
GPU	68	GPU_HOST2SOC_1	GPU MSI-X vectors (1 of 4 total)

Source Module	Interrupt Number	Interrupt Name	Interrupt Description
SDMMC4	69	SDMMC4_SYS	SDMMC Interrupt; in pair with SDMMC4
GPU	70	GPU_HOST2SOC_2	GPU MSI-X vectors (1 of 4 total)
GPU	71	GPU_HOST2SOC_3	GPU MSI-X vectors (1 of 4 total)
PCIE	72	PEX0_C0_SYS_0	PCIe Interrupt
PCIE	73	PEX0_C0_SYS_1	PCIe Interrupt
TSC	74	TSC_IRQ	TSC Interrupt
Central DMA	75	CENTRAL_DMA_CH0	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	76	CENTRAL_DMA_CH1	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	77	CENTRAL_DMA_CH2	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	78	CENTRAL_DMA_CH3	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	79	CENTRAL_DMA_CH4	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	80	CENTRAL_DMA_CH5	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	81	CENTRAL_DMA_CH6	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	82	CENTRAL_DMA_CH7	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	83	CENTRAL_DMA_CH8	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	84	CENTRAL_DMA_CH9	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	85	CENTRAL_DMA_CH10	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	86	CENTRAL_DMA_CH11	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	87	CENTRAL_DMA_CH12	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	88	CENTRAL_DMA_CH13	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	89	CENTRAL_DMA_CH14	Central DMA channel Interrupt; (1 of 32 total)

Source Module	Interrupt Number	Interrupt Name	Interrupt Description
Central DMA	90	CENTRAL_DMA_CH15	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	91	CENTRAL_DMA_CH16	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	92	CENTRAL_DMA_CH17	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	93	CENTRAL_DMA_CH18	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	94	CENTRAL_DMA_CH19	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	95	CENTRAL_DMA_CH20	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	96	CENTRAL_DMA_CH21	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	97	CENTRAL_DMA_CH22	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	98	CENTRAL_DMA_CH23	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	99	CENTRAL_DMA_CH24	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	100	CENTRAL_DMA_CH25	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	101	CENTRAL_DMA_CH26	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	102	CENTRAL_DMA_CH27	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	103	CENTRAL_DMA_CH28	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	104	CENTRAL_DMA_CH29	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	105	CENTRAL_DMA_CH30	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	106	CENTRAL_DMA_CH31	Central DMA channel Interrupt; (1 of 32 total)
Central DMA	107	CENTRAL_DMA_COMMON	Central DMA common Interrupt
PIPE2UPHY	108	PEX2_L4_P2U_SW	PIPE2UPHY Interrupt

Source Module	Interrupt Number	Interrupt Name	Interrupt Description
PIPE2UPHY	109	PEX2_L5_P2U_SW	PIPE2UPHY Interrupt
PIPE2UPHY	110	PEX2_L6_P2U_SW	PIPE2UPHY Interrupt
PIPE2UPHY	111	PEX2_L7_P2U_SW	PIPE2UPHY Interrupt
UARTA	112	UARTA	UART Interrupt
UARTB	113	UARTB	UART Interrupt
UARTC (in AON)	114	UARTC	UART Interrupt; from AON
UARTD	115	UARTD	UART Interrupt
UARTE	116	UARTE	UART Interrupt
UARTF	117	UARTF	UART Interrupt
RESERVED	118	RESERVED	NA
NVCSI	119	NVCSI	NVCSI Interrupt
TOP HSP0	120	TOP_HSP0_SHARED_0	TOP HSP0 shared Interrupt
TOP HSP0	121	TOP_HSP0_SHARED_1	TOP HSP0 shared Interrupt
TOP HSP0	122	TOP_HSP0_SHARED_2	TOP HSP0 shared Interrupt
TOP HSP0	123	TOP_HSP0_SHARED_3	TOP HSP0 shared Interrupt
TOP HSP0	124	TOP_HSP0_SHARED_4	TOP HSP0 shared Interrupt
TOP HSP0	125	TOP_HSP0_SHARED_5	TOP HSP0 shared Interrupt
TOP HSP0	126	TOP_HSP0_SHARED_6	TOP HSP0 shared Interrupt
TOP HSP0	127	TOP_HSP0_SHARED_7	TOP HSP0 shared Interrupt
TOP HSP1	128	TOP_HSP1_SHARED_0	TOP HSP1 shared Interrupt
TOP HSP1	129	TOP_HSP1_SHARED_1	TOP HSP1 shared Interrupt
TOP HSP1	130	TOP_HSP1_SHARED_2	TOP HSP1 shared Interrupt
TOP HSP1	131	TOP_HSP1_SHARED_3	TOP HSP1 shared Interrupt
TOP HSP1	132	TOP_HSP1_SHARED_4	TOP HSP1 shared Interrupt
AON - HSP	133	AON_MBOX_OUT_0	AON MBOX Interrupt
AON - HSP	134	AON_MBOX_OUT_1	AON MBOX Interrupt

Source Module	Interrupt Number	Interrupt Name	Interrupt Description
AON - HSP	135	AON_MBOX_OUT_2	AON MBOX Interrupt
AON - HSP	136	AON_MBOX_OUT_3	AON MBOX Interrupt
BPMP_HSP	137	BPMP_HSP_SHARED_1	BPMP HSP shared Interrupt
BPMP_HSP	138	BPMP_HSP_SHARED_2	BPMP HSP shared Interrupt
BPMP_HSP	139	BPMP_HSP_SHARED_3	BPMP HSP shared Interrupt
BPMP_HSP	140	BPMP_HSP_SHARED_4	BPMP HSP shared Interrupt
SCE - HSP	141	SCE_MBOX_OUT_0	SCE MBOX Interrupt
SCE - HSP	142	SCE_MBOX_OUT_1	SCE MBOX Interrupt
SCE - HSP	143	SCE_MBOX_OUT_2	SCE MBOX Interrupt
SCE - HSP	144	SCE_MBOX_OUT_3	SCE MBOX Interrupt
APE	145	APE_IRQ0	APE Interrupt; in pair with APE_FIQ0
APE	146	APE_IRQ1	APE Interrupt; in pair with APE_FIQ1
APE	147	APE_IRQ2	APE Interrupt; in pair with APE_FIQ2
APE	148	APE_IRQ3	APE Interrupt; in pair with APE_FIQ3
APE	149	APE_FIQ0	APE Interrupt; in pair with APE_IRQ0
APE	150	APE_FIQ1	APE Interrupt; in pair with APE_IRQ1
APE	151	APE_FIQ2	APE Interrupt; in pair with APE_IRQ2
APE	152	APE_FIQ3	APE Interrupt; in pair with APE_IRQ3
VI2	153	VI2_THI	VI2_THI Interrupt
VI2	154	VI2_VMID_0	VI2 VMID Interrupt
VI2	155	VI2_VMID_1	VI2 VMID Interrupt
VI2	156	VI2_VMID_2	VI2 VMID Interrupt
VI2	157	VI2_VMID_3	VI2 VMID Interrupt
VI2	158	VI2_VMID_4	VI2 VMID Interrupt
VI2	159	VI2_VMID_5	VI2 VMID Interrupt
VI2	160	VI2_VMID_6	VI2 VMID Interrupt

Source Module	Interrupt Number	Interrupt Name	Interrupt Description
VI2	161	VI2_VMID_7	VI2 VMID Interrupt
VI2	162	VI2_VMID_8	VI2 VMID Interrupt
XUSB - HOST	163	USB3_HOST_INT	XUSB Interrupt
XUSB - HOST	164	USB3_HOST_SMI	XUSB Interrupt
XUSB - HOST	165	USB3_HOST_PME	XUSB Interrupt
XUSB - DEV	166	USB3_DEV_INT	XUSB Interrupt
XUSB - PADCTL	167	XUSB_PADCTL	XUSB Interrupt; specifically targeting wakeup
XUSB - DEV	168	USB3_DEV_SMI	XUSB Interrupt
XUSB - DEV	169	USB3_DEV_PME	XUSB Interrupt
SMMU0	170	SMMU_COMBINED_NS	Arm SMMU combined non-secure Interrupt
SMMU0	171	SMMU_COMBINED_S	Arm SMMU combined secure Interrupt
AON Cluster	172	SECURE_AON_FABRIC	Secure AON fabric Interrupt
SCE Cluster	173	SECURE_SCE_FABRIC	Secure SCE fabric Interrupt
BPMP Cluster	174	SECURE_BPMP_FABRIC	Secure BPMP fabric Interrupt
RCE Cluster	175	SECURE_RCE_FABRIC	Secure RCE fabric Interrupt
TOP HSP0	176	DOORBELL_CCPLEX_NOT_SECURE	Doorbell for CCPLEX
TOP HSP0	177	DOORBELL_CCPLEX_SECURE	Doorbell for CCPLEX
RESERVED	178	RESERVED	NA
RESERVED	179	RESERVED	NA
RESERVED	180	RESERVED	NA
RESERVED	181	RESERVED	NA
RCE - HSP	182	RCE_MBOX_OUT_0	RCE MBOX Interrupt

Source Module	Interrupt Number	Interrupt Name	Interrupt Description
RCE - HSP	183	RCE_MBOX_OUT_1	RCE MBOX Interrupt
RCE - HSP	184	RCE_MBOX_OUT_2	RCE MBOX Interrupt
RCE - HSP	185	RCE_MBOX_OUT_3	RCE MBOX Interrupt
EQOS	186	EQOS_VM_0	EQOS Interrupt
EQOS	187	EQOS_VM_1	EQOS Interrupt
EQOS	188	EQOS_VM_2	EQOS Interrupt
EQOS	189	EQOS_VM_3	EQOS Interrupt
EQOS	190	EQOS_MACSEC_NON_SECURE	EQOS MACSEC Interrupt
EQOS	191	EQOS_MACSEC_SECURE	EQOS MACSEC Interrupt
TACH0	192	TACH0	TACH Interrupt
TACH1	193	TACH1	TACH Interrupt
EQOS	194	EQOS_COMMON	EQOS Interrupt
I2S7	195	I2S7	I2S Interrupt
I2S8	196	I2S8	I2S Interrupt
NVJPG1	197	NVJPG1	NVJPG1 General Interrupt
NVJPG	198	NVJPG	NVJPG General Interrupt
NVDEC	199	NVDEC	NVDEC General Interrupt
NVENC	200	NVENC	NVENC Interrupt
VI	201	VI_THI	VI_THI Interrupt
RESERVED	202	RESERVED	NA
PIPE2UPHY	203	PEX2_L0_P2U_SW	PIPE2UPHY Interrupt
ISP	204	ISP_1	ISP Interrupt
ISP	205	ISP	ISP Interrupt
VIC	206	VIC	VIC General Interrupt
UARTH	207	UARTH	UART Interrupt
BPMP - CVC	208	CVC	Central Voltage Control coming from BPMP

Source Module	Interrupt Number	Interrupt Name	Interrupt Description
AON - PMC	209	PMIC_EXT	External Power Management Chip Interrupt
BPMP - ACTMON	210	ACTMON	This is the Central ACTMON Interrupt. ACTMON Interrupts are now inside each R5 cluster; only one from BPMP exposed to LIC and called simply ACTMON
AON - PMC	211	PMC2LIC_INTR	PMC Interrupt - pmc2lic_intr
AON - WAKE	212	AON_WAKE_0	Wake Interrupt from AON
AON - WAKE	213	AON_WAKE_1	Wake Interrupt from AON
AON - WAKE	214	AON_WAKE_2	Wake Interrupt from AON
OFA	215	OFA0	OFA0 General Interrupt
AON - PM	216	AON_PM	AON Power Management Interrupt
DFD	217	DFD_0	DFD Interrupt
BPMP - SoC Therm	218	THERMAL	soc_therm Interrupt; for THERMAL event; from BPMP
BPMP - SoC Therm	219	EDP	soc_therm Interrupt; for EDP event; from BPMP
PIPE2UPHY	220	PEX2_L1_P2U_SW	PIPE2UPHY Interrupt
PIPE2UPHY	221	PEX2_L2_P2U_SW	PIPE2UPHY Interrupt
PIPE2UPHY	222	PEX2_L3_P2U_SW	PIPE2UPHY Interrupt
MCB	223	MC	MC General Interrupt
EMCB	224	EMC	EMC General Interrupt
RESERVED	225	RESERVED	NA
AON - CAR	226	AON_CAR	Interrupt from CAR logic inside AON
EMCB	227	EMC_CORR_ECC_ERR	Correctable ECC error
TSEC	228	TSEC	TSEC General Interrupt
RESERVED	229	RESERVED	NA
RESERVED	230	RESERVED	NA
CBB	231	CBB_SECURE	CBB Secure Interrupt

Source Module	Interrupt Number	Interrupt Name	Interrupt Description
SMMU1	232	SMMU1_COMBINED_NS	Arm SMMU combined non-secure Interrupt
SMMU1	233	SMMU1_COMBINED_S	Arm SMMU combined secure Interrupt
PVA0	234	PVA0	PVA0 Interrupt
RESERVED	235	RESERVED	NA
NVDLA0	236	NVDLA0	NVDLA0 Interrupt
NVDLA1	237	NVDLA1	NVDLA1 Interrupt
SMMU3	238	SMMU3_COMBINED_NS	Arm SMMU combined non-secure Interrupt
SMMU3	239	SMMU3_COMBINED_S	Arm SMMU combined secure Interrupt
SMMU2	240	SMMU2_COMBINED_NS	Arm SMMU combined non-secure Interrupt
SMMU2	241	SMMU2_COMBINED_S	Arm SMMU combined secure Interrupt
SMMU4	242	SMMU4_COMBINED_NS	Arm SMMU combined non-secure Interrupt
SMMU4	243	SMMU4_COMBINED_S	Arm SMMU combined secure Interrupt
TOP HSP1	244	TOP_HSP1_SHARED_5	TOP HSP1 shared Interrupt
TOP HSP1	245	TOP_HSP1_SHARED_6	TOP HSP1 shared Interrupt
TOP HSP1	246	TOP_HSP1_SHARED_7	TOP HSP1 shared Interrupt
VI	247	VI_VMID_0	VI VMID Interrupt
VI	248	VI_VMID_1	VI VMID Interrupt
VI	249	VI_VMID_2	VI VMID Interrupt
VI	250	VI_VMID_3	VI VMID Interrupt
VI	251	VI_VMID_4	VI VMID Interrupt
VI	252	VI_VMID_5	VI VMID Interrupt
VI	253	VI_VMID_6	VI VMID Interrupt
VI	254	VI_VMID_7	VI VMID Interrupt
VI	255	VI_VMID_8	VI VMID Interrupt
TOP TKE	256	TOP_TKE_SHARED_10	Shared timer Interrupt from top TKE (1 of 16 total)

Source Module	Interrupt Number	Interrupt Name	Interrupt Description
TOP TKE	257	TOP_TKE_SHARED_11	Shared timer Interrupt from top TKE (1 of 16 total)
TOP TKE	258	TOP_TKE_SHARED_12	Shared timer Interrupt from top TKE (1 of 16 total)
TOP TKE	259	TOP_TKE_SHARED_13	Shared timer Interrupt from top TKE (1 of 16 total)
TOP TKE	260	TOP_TKE_SHARED_14	Shared timer Interrupt from top TKE (1 of 16 total)
TOP TKE	261	TOP_TKE_SHARED_15	Shared timer Interrupt from top TKE (1 of 16 total)
RESERVED	262	RESERVED	NA
HOST Controller	263	HOST1X_GEN_CPU	HOST Controller General Interrupt
HOST Controller	264	HOST1X_SYNCPT_CPU	HOST Controller syncpt Interrupt
TOP HSP2	265	TOP_HSP2_SHARED_0	TOP HSP2 shared Interrupt
TOP HSP2	266	TOP_HSP2_SHARED_1	TOP HSP2 shared Interrupt
TOP HSP2	267	TOP_HSP2_SHARED_2	TOP HSP2 shared Interrupt
TOP HSP2	268	TOP_HSP2_SHARED_3	TOP HSP2 shared Interrupt
TOP HSP2	269	TOP_HSP2_SHARED_4	TOP HSP2 shared Interrupt
TOP HSP2	270	TOP_HSP2_SHARED_5	TOP HSP2 shared Interrupt
TOP HSP2	271	TOP_HSP2_SHARED_6	TOP HSP2 shared Interrupt
TOP HSP2	272	TOP_HSP2_SHARED_7	TOP HSP2 shared Interrupt
HOST Controller	273	HOST1X_TZ_GEN_CPU	HOST Controller secure Interrupt
HOST Controller	274	HOST1X_TZ_SYNCPT_CPU	HOST Controller secure Interrupt
HOST Controller	275	HOST1X_GEN_BPMP	HOST1X general Interrupt for BPMP
HOST Controller	276	HOST1X_SYNCPT_CAMERAPROC_0	HOST Controller syncpt for camera processing
HOST Controller	277	HOST1X_SYNCPT_CAMERAPROC_1	HOST Controller syncpt for camera processing
RESERVED	278	RESERVED	NA
AON - CEC	279	CEC	CEC Interrupt
AON - PMC	280	VFMON_INTR	AO Voltage/Frequency Monitor Interrupt
SE0	281	SE_APB	SE APB programming Interrupt

Source Module	Interrupt Number	Interrupt Name	Interrupt Description
SE0	282	SE_HOST1X	SE Interrupt from Host Controller programming interface and only for Nvidia engines
RESERVED	283	RESERVED	NA
SE0	284	SE_ELPRNG	SE Interrupt for Elliptic RNG module (RNG1 in SE)
UARTI	285	UARTI	UART Interrupt
UARTJ	286	UARTJ	UART Interrupt
RESERVED	287	RESERVED	NA
GPIO - CTL0	288	GPIO0_0	GPIO Interrupt
GPIO - CTL0	289	GPIO0_1	GPIO Interrupt
GPIO - CTL0	290	GPIO0_2	GPIO Interrupt
GPIO - CTL0	291	GPIO0_3	GPIO Interrupt
GPIO - CTL0	292	GPIO0_4	GPIO Interrupt
GPIO - CTL0	293	GPIO0_5	GPIO Interrupt
GPIO - CTL0	294	GPIO0_6	GPIO Interrupt
GPIO - CTL0	295	GPIO0_7	GPIO Interrupt
GPIO - CTL1	296	GPIO1_0	GPIO Interrupt
GPIO - CTL1	297	GPIO1_1	GPIO Interrupt
GPIO - CTL1	298	GPIO1_2	GPIO Interrupt
GPIO - CTL1	299	GPIO1_3	GPIO Interrupt
GPIO - CTL1	300	GPIO1_4	GPIO Interrupt
GPIO - CTL1	301	GPIO1_5	GPIO Interrupt
GPIO - CTL1	302	GPIO1_6	GPIO Interrupt

Source Module	Interrupt Number	Interrupt Name	Interrupt Description
GPIO - CTL1	303	GPIO1_7	GPIO Interrupt
GPIO - CTL2	304	GPIO2_0	GPIO Interrupt
GPIO - CTL2	305	GPIO2_1	GPIO Interrupt
GPIO - CTL2	306	GPIO2_2	GPIO Interrupt
GPIO - CTL2	307	GPIO2_3	GPIO Interrupt
GPIO - CTL2	308	GPIO2_4	GPIO Interrupt
GPIO - CTL2	309	GPIO2_5	GPIO Interrupt
GPIO - CTL2	310	GPIO2_6	GPIO Interrupt
GPIO - CTL2	311	GPIO2_7	GPIO Interrupt
GPIO - CTL3	312	GPIO3_0	GPIO Interrupt
GPIO - CTL3	313	GPIO3_1	GPIO Interrupt
GPIO - CTL3	314	GPIO3_2	GPIO Interrupt
GPIO - CTL3	315	GPIO3_3	GPIO Interrupt
GPIO - CTL3	316	GPIO3_4	GPIO Interrupt
GPIO - CTL3	317	GPIO3_5	GPIO Interrupt
GPIO - CTL3	318	GPIO3_6	GPIO Interrupt
GPIO - CTL3	319	GPIO3_7	GPIO Interrupt
GPIO - CTL4	320	GPIO4_0	GPIO Interrupt
GPIO - CTL4	321	GPIO4_1	GPIO Interrupt

Source Module	Interrupt Number	Interrupt Name	Interrupt Description
GPIO - CTL4	322	GPIO4_2	GPIO Interrupt
GPIO - CTL4	323	GPIO4_3	GPIO Interrupt
GPIO - CTL4	324	GPIO4_4	GPIO Interrupt
GPIO - CTL4	325	GPIO4_5	GPIO Interrupt
GPIO - CTL4	326	GPIO4_6	GPIO Interrupt
GPIO - CTL4	327	GPIO4_7	GPIO Interrupt
GPIO - CTL5	328	GPIO5_0	GPIO Interrupt
GPIO - CTL5	329	GPIO5_1	GPIO Interrupt
GPIO - CTL5	330	GPIO5_2	GPIO Interrupt
GPIO - CTL5	331	GPIO5_3	GPIO Interrupt
GPIO - CTL5	332	GPIO5_4	GPIO Interrupt
GPIO - CTL5	333	GPIO5_5	GPIO Interrupt
GPIO - CTL5	334	GPIO5_6	GPIO Interrupt
GPIO - CTL5	335	GPIO5_7	GPIO Interrupt
PIPE2UPHY	336	PEX0_L0_P2U_SW	PIPE2UPHY Interrupt
PIPE2UPHY	337	PEX0_L1_P2U_SW	PIPE2UPHY Interrupt
PIPE2UPHY	338	PEX0_L2_P2U_SW	PIPE2UPHY Interrupt
PIPE2UPHY	339	PEX0_L3_P2U_SW	PIPE2UPHY Interrupt
PIPE2UPHY	340	PEX0_L4_P2U_SW	PIPE2UPHY Interrupt

Source Module	Interrupt Number	Interrupt Name	Interrupt Description
PIPE2UPHY	341	PEX0_L5_P2U_SW	PIPE2UPHY Interrupt
PIPE2UPHY	342	PEX0_L6_P2U_SW	PIPE2UPHY Interrupt
PIPE2UPHY	343	PEX0_L7_P2U_SW	PIPE2UPHY Interrupt
PIPE2UPHY	344	PEX1_L0_P2U_SW	PIPE2UPHY Interrupt
PIPE2UPHY	345	PEX1_L1_P2U_SW	PIPE2UPHY Interrupt
PIPE2UPHY	346	PEX1_L2_P2U_SW	PIPE2UPHY Interrupt
PIPE2UPHY	347	PEX1_L3_P2U_SW	PIPE2UPHY Interrupt
PIPE2UPHY	348	PEX1_L4_P2U_SW	PIPE2UPHY Interrupt
PIPE2UPHY	349	PEX1_L5_P2U_SW	PIPE2UPHY Interrupt
PIPE2UPHY	350	PEX1_L6_P2U_SW	PIPE2UPHY Interrupt
PIPE2UPHY	351	PEX1_L7_P2U_SW	PIPE2UPHY Interrupt
PCIE	352	PEX1_C6_SYS_0	PCIe Interrupt
PCIE	353	PEX1_C6_SYS_1	PCIe Interrupt
PCIE	354	PEX2_C7_SYS_0	PCIe Interrupt
PCIE	355	PEX2_C7_SYS_1	PCIe Interrupt
PCIE	356	PEX2_C8_SYS_0	PCIe Interrupt
PCIE	357	PEX2_C8_SYS_1	PCIe Interrupt
PCIE	358	PEX2_C9_SYS_0	PCIe Interrupt
PCIE	359	PEX2_C9_SYS_1	PCIe Interrupt
PCIE	360	PEX2_C10_SYS_0	PCIe Interrupt
PCIE	361	PEX2_C10_SYS_1	PCIe Interrupt
PSC	362	PSC_MB0	PSC MBOX Interrupt
PSC	363	PSC_MB1	PSC MBOX Interrupt

Source Module	Interrupt Number	Interrupt Name	Interrupt Description
PSC	364	PSC_MB2	PSC MBOX Interrupt
PSC	365	PSC_MB3	PSC MBOX Interrupt
PSC	366	PSC_MB4	PSC MBOX Interrupt
PSC	367	PSC_MB5	PSC MBOX Interrupt
PSC	368	PSC_MB6	PSC MBOX Interrupt
PSC	369	PSC_MB7	PSC MBOX Interrupt
PSC	370	PSC_MB8	PSC MBOX Interrupt
PSC	371	PSC_MB9	PSC MBOX Interrupt
PSC	372	PSC_MB10	PSC MBOX Interrupt
PSC	373	PSC_MB11	PSC MBOX Interrupt
RESERVED	374	RESERVED	NA
RESERVED	375	RESERVED	NA
DCE - TKE	376	DCE_WDT_REMOTE	Remote Interrupt associated with a given WDT; associated cluster probably down if risen
DCE - HSP	377	DCE_MBOX_OUT_0	DCE MBOX Interrupt
DCE - HSP	378	DCE_MBOX_OUT_1	DCE MBOX Interrupt
DCE - HSP	379	DCE_MBOX_OUT_2	DCE MBOX Interrupt
DCE - HSP	380	DCE_MBOX_OUT_3	DCE MBOX Interrupt
DCE Cluster	381	SECURE_DCE_FABRIC	Secure DCE fabric Interrupt
RESERVED	382	RESERVED	NA
RESERVED	383	RESERVED	NA
MGBE0	384	MGBE0_COMMON	MGBE Interrupt
MGBE0	385	MGBE0_VM_0	MGBE VM Interrupt
MGBE0	386	MGBE0_VM_1	MGBE VM Interrupt
MGBE0	387	MGBE0_VM_2	MGBE VM Interrupt
MGBE0	388	MGBE0_VM_3	MGBE VM Interrupt
MGBE0	389	MGBE0_VM_4	MGBE VM Interrupt
MGBE0 - MACSEC	390	MGBE0_MACSEC_NON_SECURE	MGBE MACSEC Interrupt

Source Module	Interrupt Number	Interrupt Name	Interrupt Description
MGBE0 - MACSEC	391	MGBE0_MACSEC_SECURE	MGBE MACSEC Interrupt
MGBE1	392	MGBE1_COMMON	MGBE Interrupt
MGBE1	393	MGBE1_VM_0	MGBE VM Interrupt
MGBE1	394	MGBE1_VM_1	MGBE VM Interrupt
MGBE1	395	MGBE1_VM_2	MGBE VM Interrupt
MGBE1	396	MGBE1_VM_3	MGBE VM Interrupt
MGBE1	397	MGBE1_VM_4	MGBE VM Interrupt
MGBE1 - MACSEC	398	MGBE1_MACSEC_NON_SECURE	MGBE MACSEC Interrupt
MGBE1 - MACSEC	399	MGBE1_MACSEC_SECURE	MGBE MACSEC Interrupt
MGBE2	400	MGBE2_COMMON	MGBE Interrupt
MGBE2	401	MGBE2_VM_0	MGBE VM Interrupt
MGBE2	402	MGBE2_VM_1	MGBE VM Interrupt
MGBE2	403	MGBE2_VM_2	MGBE VM Interrupt
MGBE2	404	MGBE2_VM_3	MGBE VM Interrupt
MGBE2	405	MGBE2_VM_4	MGBE VM Interrupt
MGBE2 - MACSEC	406	MGBE2_MACSEC_NON_SECURE	MGBE MACSEC Interrupt
MGBE2 - MACSEC	407	MGBE2_MACSEC_SECURE	MGBE MACSEC Interrupt
MGBE3	408	MGBE3_COMMON	MGBE Interrupt
MGBE3	409	MGBE3_VM_0	MGBE VM Interrupt
MGBE3	410	MGBE3_VM_1	MGBE VM Interrupt
MGBE3	411	MGBE3_VM_2	MGBE VM Interrupt
MGBE3	412	MGBE3_VM_3	MGBE VM Interrupt
MGBE3	413	MGBE3_VM_4	MGBE VM Interrupt
MGBE3 - MACSEC	414	MGBE3_MACSEC_NON_SECURE	MGBE MACSEC Interrupt
MGBE3 - MACSEC	415	MGBE3_MACSEC_SECURE	MGBE MACSEC Interrupt

Source Module	Interrupt Number	Interrupt Name	Interrupt Description
NVDISPLAY	416	DISPLAY_2HOST	Display Interrupt
NVDISPLAY	417	DISPLAY_2GSP	Display Interrupt
NVDISPLAY	418	DISPLAY_2PMU	Display Interrupt
DPAUX	419	DPAUX0	DPAUX Interrupt
RESERVED	420	RESERVED	NA
RESERVED	421	RESERVED	NA
RESERVED	422	RESERVED	NA
SE1	423	SE1_APB	SE1 APB programming Interrupt
SE1	424	SE1_HOST1X	SE1 Interrupt from Host Controller programming interface and only for Nvidia engines
SE1	425	SE1_ELPPKA	SE1 Interrupt for Elliptic PKA module (PKA1 in SE)
SE1	426	SE1_ELPRNG	SE1 Interrupt for Elliptic RNG module (RNG1 in SE)
RESERVED	427	RESERVED	NA
PCIE	428	PEX2_C8_INTA	PCIe Interrupt
PCIE	429	PEX2_C8_INTB	PCIe Interrupt
PCIE	430	PEX2_C8_INTC	PCIe Interrupt
PCIE	431	PEX2_C8_INTD	PCIe Interrupt
PVA0	432	PVA0_VM_0	PVA0 VM Interrupt
PVA0	433	PVA0_VM_1	PVA0 VM Interrupt
PVA0	434	PVA0_VM_2	PVA0 VM Interrupt
PVA0	435	PVA0_VM_3	PVA0 VM Interrupt
PVA0	436	PVA0_VM_4	PVA0 VM Interrupt
PVA0	437	PVA0_VM_5	PVA0 VM Interrupt
PVA0	438	PVA0_VM_6	PVA0 VM Interrupt
PVA0	439	PVA0_VM_7	PVA0 VM Interrupt
PCIE	440	PEX2_C9_INTA	PCIE Interrupt
PCIE	441	PEX2_C9_INTB	PCIe Interrupt

Source Module	Interrupt Number	Interrupt Name	Interrupt Description
PCIE	442	PEX2_C9_INTC	PCIe Interrupt
PCIE	443	PEX2_C9_INTD	PCIe Interrupt
PCIE	444	PEX2_C10_INTA	PCIe Interrupt
PCIE	445	PEX2_C10_INTB	PCIe Interrupt
PCIE	446	PEX2_C10_INTC	PCIe Interrupt
PCIE	447	PEX2_C10_INTD	PCIe Interrupt
HOST Controller	448	HOST1X2VM1_0_SYNCPT	HOST Controller VM Interrupt
HOST Controller	449	HOST1X2VM1_1_SYNCPT	HOST Controller VM Interrupt
HOST Controller	450	HOST1X2VM1_2_SYNCPT	HOST Controller VM Interrupt
HOST Controller	451	HOST1X2VM1_3_SYNCPT	HOST Controller VM Interrupt
HOST Controller	452	HOST1X2VM1_4_SYNCPT	HOST Controller VM Interrupt
HOST Controller	453	HOST1X2VM1_5_SYNCPT	HOST Controller VM Interrupt
HOST Controller	454	HOST1X2VM1_6_SYNCPT	HOST Controller VM Interrupt
HOST Controller	455	HOST1X2VM1_7_SYNCPT	HOST Controller VM Interrupt
HOST Controller	456	HOST1X2VM2_0_SYNCPT	HOST Controller VM Interrupt
HOST Controller	457	HOST1X2VM2_1_SYNCPT	HOST Controller VM Interrupt
HOST Controller	458	HOST1X2VM2_2_SYNCPT	HOST Controller VM Interrupt
HOST Controller	459	HOST1X2VM2_3_SYNCPT	HOST Controller VM Interrupt
HOST Controller	460	HOST1X2VM2_4_SYNCPT	HOST Controller VM Interrupt
HOST Controller	461	HOST1X2VM2_5_SYNCPT	HOST Controller VM Interrupt
HOST Controller	462	HOST1X2VM2_6_SYNCPT	HOST Controller VM Interrupt

Source Module	Interrupt Number	Interrupt Name	Interrupt Description
HOST Controller	463	HOST1X2VM2_7_SYNCPT	HOST Controller VM Interrupt
HOST Controller	464	HOST1X2VM3_0_SYNCPT	HOST Controller VM Interrupt
HOST Controller	465	HOST1X2VM3_1_SYNCPT	HOST Controller VM Interrupt
HOST Controller	466	HOST1X2VM3_2_SYNCPT	HOST Controller VM Interrupt
HOST Controller	467	HOST1X2VM3_3_SYNCPT	HOST Controller VM Interrupt
HOST Controller	468	HOST1X2VM3_4_SYNCPT	HOST Controller VM Interrupt
HOST Controller	469	HOST1X2VM3_5_SYNCPT	HOST Controller VM Interrupt
HOST Controller	470	HOST1X2VM3_6_SYNCPT	HOST Controller VM Interrupt
HOST Controller	471	HOST1X2VM3_7_SYNCPT	HOST Controller VM Interrupt
HOST Controller	472	HOST1X2VM4_0_SYNCPT	HOST Controller VM Interrupt
HOST Controller	473	HOST1X2VM4_1_SYNCPT	HOST Controller VM Interrupt
HOST Controller	474	HOST1X2VM4_2_SYNCPT	HOST Controller VM Interrupt
HOST Controller	475	HOST1X2VM4_3_SYNCPT	HOST Controller VM Interrupt
HOST Controller	476	HOST1X2VM4_4_SYNCPT	HOST Controller VM Interrupt
HOST Controller	477	HOST1X2VM4_5_SYNCPT	HOST Controller VM Interrupt
HOST Controller	478	HOST1X2VM4_6_SYNCPT	HOST Controller VM Interrupt
HOST Controller	479	HOST1X2VM4_7_SYNCPT	HOST Controller VM Interrupt
HOST Controller	480	HOST1X2VM5_0_SYNCPT	HOST Controller VM Interrupt
HOST Controller	481	HOST1X2VM5_1_SYNCPT	HOST Controller VM Interrupt

Source Module	Interrupt Number	Interrupt Name	Interrupt Description
HOST Controller	482	HOST1X2VM5_2_SYNCPT	HOST Controller VM Interrupt
HOST Controller	483	HOST1X2VM5_3_SYNCPT	HOST Controller VM Interrupt
HOST Controller	484	HOST1X2VM5_4_SYNCPT	HOST Controller VM Interrupt
HOST Controller	485	HOST1X2VM5_5_SYNCPT	HOST Controller VM Interrupt
HOST Controller	486	HOST1X2VM5_6_SYNCPT	HOST Controller VM Interrupt
HOST Controller	487	HOST1X2VM5_7_SYNCPT	HOST Controller VM Interrupt
HOST Controller	488	HOST1X2VM6_0_SYNCPT	HOST Controller VM Interrupt
HOST Controller	489	HOST1X2VM6_1_SYNCPT	HOST Controller VM Interrupt
HOST Controller	490	HOST1X2VM6_2_SYNCPT	HOST Controller VM Interrupt
HOST Controller	491	HOST1X2VM6_3_SYNCPT	HOST Controller VM Interrupt
HOST Controller	492	HOST1X2VM6_4_SYNCPT	HOST Controller VM Interrupt
HOST Controller	493	HOST1X2VM6_5_SYNCPT	HOST Controller VM Interrupt
HOST Controller	494	HOST1X2VM6_6_SYNCPT	HOST Controller VM Interrupt
HOST Controller	495	HOST1X2VM6_7_SYNCPT	HOST Controller VM Interrupt
HOST Controller	496	HOST1X2VM7_0_SYNCPT	HOST Controller VM Interrupt
HOST Controller	497	HOST1X2VM7_1_SYNCPT	HOST Controller VM Interrupt
HOST Controller	498	HOST1X2VM7_2_SYNCPT	HOST Controller VM Interrupt
HOST Controller	499	HOST1X2VM7_3_SYNCPT	HOST Controller VM Interrupt
HOST Controller	500	HOST1X2VM7_4_SYNCPT	HOST Controller VM Interrupt

Source Module	Interrupt Number	Interrupt Name	Interrupt Description
HOST Controller	501	HOST1X2VM7_5_SYNCPT	HOST Controller VM Interrupt
HOST Controller	502	HOST1X2VM7_6_SYNCPT	HOST Controller VM Interrupt
HOST Controller	503	HOST1X2VM7_7_SYNCPT	HOST Controller VM Interrupt
HOST Controller	504	HOST1X2VM8_0_SYNCPT	HOST Controller VM Interrupt
HOST Controller	505	HOST1X2VM8_1_SYNCPT	HOST Controller VM Interrupt
HOST Controller	506	HOST1X2VM8_2_SYNCPT	HOST Controller VM Interrupt
HOST Controller	507	HOST1X2VM8_3_SYNCPT	HOST Controller VM Interrupt
HOST Controller	508	HOST1X2VM8_4_SYNCPT	HOST Controller VM Interrupt
HOST Controller	509	HOST1X2VM8_5_SYNCPT	HOST Controller VM Interrupt
HOST Controller	510	HOST1X2VM8_6_SYNCPT	HOST Controller VM Interrupt
HOST Controller	511	HOST1X2VM8_7_SYNCPT	HOST Controller VM Interrupt
PCIE	512	PEX0_C0_INTA	PCIe Interrupt
PCIE	513	PEX0_C0_INTB	PCIe Interrupt
PCIE	514	PEX0_C0_INTC	PCIe Interrupt
PCIE	515	PEX0_C0_INTD	PCIe Interrupt
PCIE	516	PEX0_C1_INTA	PCIe Interrupt
PCIE	517	PEX0_C1_INTB	PCIe Interrupt
PCIE	518	PEX0_C1_INTC	PCIe Interrupt
PCIE	519	PEX0_C1_INTD	PCIe Interrupt
PCIE	520	PEX0_C2_INTA	PCIe Interrupt
PCIE	521	PEX0_C2_INTB	PCIe Interrupt
PCIE	522	PEX0_C2_INTC	PCIe Interrupt
PCIE	523	PEX0_C2_INTD	PCIe Interrupt

Source Module	Interrupt Number	Interrupt Name	Interrupt Description
PCIE	524	PEX0_C3_INTA	PCIe Interrupt
PCIE	525	PEX0_C3_INTB	PCIe Interrupt
PCIE	526	PEX0_C3_INTC	PCIe Interrupt
PCIE	527	PEX0_C3_INTD	PCIe Interrupt
PCIE	528	PEX0_C4_INTA	PCIe Interrupt
PCIE	529	PEX0_C4_INTB	PCIe Interrupt
PCIE	530	PEX0_C4_INTC	PCIe Interrupt
PCIE	531	PEX0_C4_INTD	PCIe Interrupt
PCIE	532	PEX1_C5_INTA	PCIe Interrupt
PCIE	533	PEX1_C5_INTB	PCIe Interrupt
PCIE	534	PEX1_C5_INTC	PCIe Interrupt
PCIE	535	PEX1_C5_INTD	PCIe Interrupt
PCIE	536	PEX1_C6_INTA	PCIe Interrupt
PCIE	537	PEX1_C6_INTB	PCIe Interrupt
PCIE	538	PEX1_C6_INTC	PCIe Interrupt
PCIE	539	PEX1_C6_INTD	PCIe Interrupt
PCIE	540	PEX2_C7_INTA	PCIe Interrupt
PCIE	541	PEX2_C7_INTB	PCIe Interrupt
PCIE	542	PEX2_C7_INTC	PCIe Interrupt
PCIE	543	PEX2_C7_INTD	PCIe Interrupt

8.2.2.4.2 Interrupt Handling by CCPLEX via GIC-600AE

The 544 Interrupt signals that are routed to the LIC (as listed in the table above) are sent from the LIC to the GIC-600AE in CCPLEX and joined by another 12 Interrupt signals to the GIC-600AE. The additional Interrupt signals are listed in the table below.

Table 8.4 GIC-600AE Interrupts Mapping

Source Module	Interrupt Number	Interrupt Name	Interrupt Description
(See table above)	543:0	(See table above)	(See table above)

Source Module	Interrupt Number	Interrupt Name	Interrupt Description
TKE	544	WDT_FIQ	Connected to WDT at SoC level
TKE	545	WDT_IRQ	Connected to WDT at SoC level
CCPLEX	546	CCPLEX_SPI_546	GIC PMU Interrupt
CCPLEX	547	CCPLEX_SPI_547	DSU PMU Interrupt
CCPLEX	548	CCPLEX_SPI_548	DSU PMU Interrupt
CCPLEX	549	CCPLEX_SPI_549	DSU PMU Interrupt
RESERVED	550	RESERVED	NA
CCPLEX	551	CCPLEX_SPI_551	SCF PMU Interrupt
CCPLEX	552	CCPLEX_SPI_552	Online IST Interrupt
CCPLEX	553	CCPLEX_SPI_553	Online IST Interrupt
CCPLEX	554	CCPLEX_SPI_554	Online IST Interrupt
RESERVED	555	RESERVED	NA

8.2.2.5 Interrupt Handling by Different Cortex-R5 Clusters

8.2.2.5.1 Arm Vectored Interrupt Controller Chain (AVIC Chain)

The Cortex-R5 clusters employ two Arm Vectored Interrupt controllers, AVIC0 and AVIC1, in daisy chain, referred to as AVIC Chain, to handle two types of Interrupts:

1. Local Interrupts internal to the Cortex-R5 cluster
2. External Interrupts from the Legacy Interrupt controller (LIC) and other SoC functional modules

The AVIC Chain is located off the Cortex-R5 cluster NoC whose firewall protects accesses to AVIC configuration space.

The Cortex-R5 interfaces to AVIC0 in AVIC Chain and performs the following functions in hardware to minimize the software overheads of servicing Interrupt.

- Determine which Interrupt source is requesting service.
- Determine where the Interrupt Service Routine (ISR) for the Interrupt source is located.
- Disable the Interrupt source, before re-enabling processor Interrupts to permit another Interrupt to be taken.

The two types of Interrupts are connected to the AVIC Chain as follows:

- Local Interrupts to AVIC0 or AVIC1, where they get vectored (identified with the corresponding ISR's starting address) then serviced by Cortex-R5 with low Interrupt latency.
- One or two pairs of external Interrupts, LIC_IRQ and LIC_FIQ (from LIC to AVIC0), that are the aggregation from a subset of global Interrupts. Services to these aggregated Interrupts are expected to have high latency.

Note: There is one pair of IRQ/FIQ from LIC to BPMP, while two pairs of IRQ/FIQ are sent from LIC to AON, RCE, and SCE respectively. (See the Interrupt Routing (Top-Level View) diagram above.)

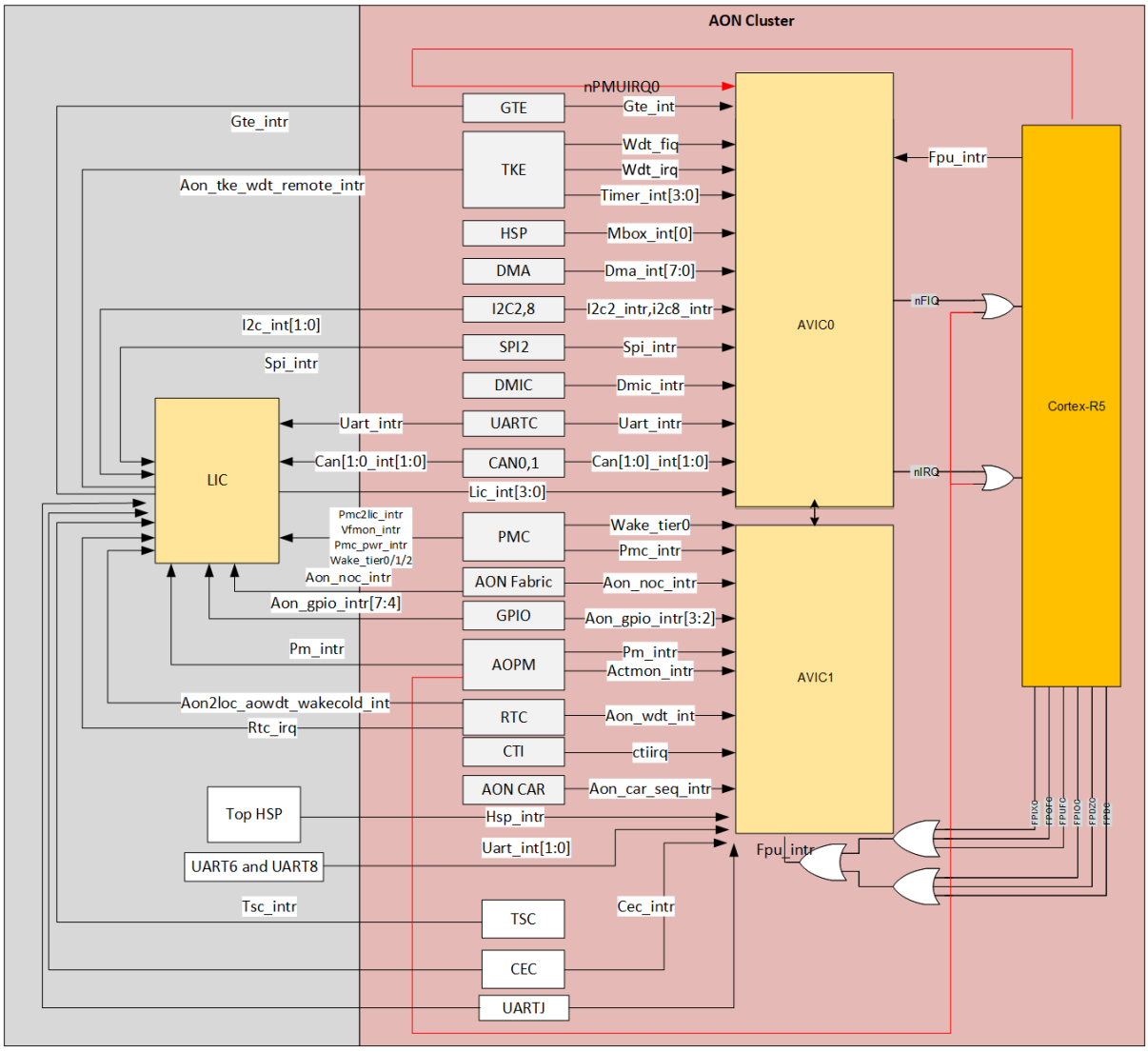
The SoC comprises six Cortex-R5 clusters.

- AON (SPE) cluster
- BPMP cluster
- RCE cluster
- SCE cluster
- FSI Cluster
- DCE Cluster

8.2.2.5.2 AON Cluster Interrupt Controller

The AON Cluster Interrupt controller with its AVIC Chain is shown in the diagram below.

Figure 8.3 AON Cluster Interrupt Structure



All the AON-related Interrupts are listed in the table below.

Table 8.5 AON Interrupts Connection

Source	Interrupt Name	Software Name	AVIC Connection	LIC Connection
AON - TKE	aon_tke_wdt_fiq_	(AON_INTERRUPT_WDTFIQ)	0 - 0	none
AON - TKE	aon_tke_wdt_irq	(AON_INTERRUPT_WDTIRQ)	0 - 1	none
AON - TKE	aon_tke_tmr_irq[0]	(AON_INTERRUPT_TIMER0)	0 - 2	none
AON - TKE	aon_tke_tmr_irq[1]	(AON_INTERRUPT_TIMER1)	0 - 3	none

Source	Interrupt Name	Software Name	AVIC Connection	LIC Connection
AON - TKE	aon_tke_tmr_irq[2]	(AON_INTERRUPT_TIMER2)	0 - 4	none
AON - TKE	aon_tke_tmr_irq[3]	(AON_INTERRUPT_TIMER3)	0 - 5	none
AON - HSP Mbox	aon_hsp_shrd_intr[0]	(AON_INTERRUPT_MBOX)	0 - 6	none
AON - GTE	aon_gte_interrupt	(AON_INTERRUPT_GTE)	0 - 7	none
AON - R5	aon_int_npmuirq	(AON_INTERRUPT_PMU)	0 - 8	none
AON - DMA	dma_intr[0]	(AON_INTERRUPT_DMA0)	0 - 9	none
AON - DMA	dma_intr[1]	(AON_INTERRUPT_DMA1)	0 - 10	none
AON - DMA	dma_intr[2]	(AON_INTERRUPT_DMA2)	0 - 11	none
AON - DMA	dma_intr[3]	(AON_INTERRUPT_DMA3)	0 - 12	none
AON - DMA	dma_intr[4]	(AON_INTERRUPT_DMA4)	0 - 13	none
AON - DMA	dma_intr[5]	(AON_INTERRUPT_DMA5)	0 - 14	none
AON - DMA	dma_intr[6]	(AON_INTERRUPT_DMA6)	0 - 15	none
AON - DMA	dma_intr[7]	(AON_INTERRUPT_DMA7)	0 - 16	none
AON - I2C2	aon_i2c2_rupt (AVIC) / aon2lic_i2c2int (LIC)	(AON_INTERRUPT_I2C2 (AVIC) / I2C2 (LIC))	0 - 18	26
AON - I2C8	aon_i2c3_rupt (AVIC) / aon2lic_i2c3int (LIC)	(AON_INTERRUPT_I2C3 (AVIC) / I2C8 (LIC))	0 - 19	32
AON - SPI2	aon_spi_intr (AVIC) / aon2lic_spiint (LIC)	(AON_INTERRUPT_SPI (AVIC) / SPI2 (LIC))	0 - 20	37
AON - DMIC	aon_dmic_intr	(AON_INTERRUPT_DMIC)	0 - 21	none
AON - UARTC	aon_uart1_intr (AVIC) / aon2lic_uartgint (LIC)	(AON_INTERRUPT_UART_1 (AVIC) / UARTC (LIC))	0 - 22	114
AON - CAN1	aon_can1_intr[0] (AVIC) / aon2lic_can1int[0] (LIC)	(AON_INTERRUPT_CAN1_0 (AVIC) / CAN1_0 (LIC))	0 - 24	40
AON - CAN1	aon_can1_intr[1] (AVIC) / aon2lic_can1int[1] (LIC)	(AON_INTERRUPT_CAN1_1 (AVIC) / CAN1_1 (LIC))	0 - 25	41
AON - CAN2	aon_can2_intr[0] (AVIC) / aon2lic_can2int[0] (LIC)	(AON_INTERRUPT_CAN2_0 (AVIC) / CAN2_0 (LIC))	0 - 26	42
AON - CAN2	aon_can2_intr[1] (AVIC) / aon2lic_can2int[1] (LIC)	(AON_INTERRUPT_CAN2_1 (AVIC) / CAN2_1 (LIC))	0 - 27	43
LIC	lic2aon_int[0]	(AON_INTERRUPT_LIC0)	0 - 28	none
LIC	lic2aon_int[1]	(AON_INTERRUPT_LIC1)	0 - 29	none
LIC	lic2aon_int[2]	(AON_INTERRUPT_LIC2)	0 - 30	none

Source	Interrupt Name	Software Name	AVIC Connection	LIC Connection
LIC	lic2aon_int[3]	(AON_INTERRUPT_LIC3)	0 - 31	none
AON - GPIO	gpioaon_spe_rupt[0]	(AON_INTERRUPT_GPIO)	1 - 1	none
AON - Wake	wake_tier0 (AVIC) / aon2lic_tier0wake_int (LIC)	(AON_INTERRUPT_WAKE0 (AVIC) / AON_WAKE_0 (LIC))	1 - 2	212
AON - PMC	pmc2vic_intr	(AON_INTERRUPT_PMC)	1 - 3	none
AON - PM	pm2vic_intr	(AON_INTERRUPT_PM)	1 - 5	none
AON - R5 FPU	cpu2vic_fpuint	(AON_INTERRUPT_FPUINT)	1 - 6	none
AON - PM	pm_actmon_intr	(AON_INTERRUPT_ACTMON)	1 - 8	none
AON - RTC	aowdt_irq	(AON_INTERRUPT_AOWDT)	1 - 9	none
top HSP	shsp2spe_db	(ON_INTERRUPT_TOP0_HSP_DB)	1 - 10	none
AON - CTI	aon_cpu_csite_nctiirq	(AON_INTERRUPT_CTIIRQ)	1 - 11	none
AON - NOC	aon_noc_secure_intr (AVIC) / aon2lic_fabric_sec_intr (LIC)	(ON_INTERRUPT_NOC_SECURE (AVIC) / SECURE_AON_FABRIC (LIC))	1 - 12	172
AON - CAR	aon_car_interrupt (AVIC) / aon_car_interrupt (LIC)	(AON_INTERRUPT_CAR (AVIC) / AON_CAR (LIC))	1 - 13	226
UARTF	uart_f_uart_intr	(AON_INTERRUPT_UART6)	1 - 14	none
UARTH	uart_h_uart_intr	(AON_INTERRUPT_UART8)	1 - 15	none
AON - GPIO	gpioaon_spe_rupt[1]	(AON_INTERRUPT_GPIO_3)	1 - 16	none
AON - CEC	cec_intr	(AON_INTERRUPT_CEC)	1 - 17	279
AON - UARTJ	uart_j_intr	(AON_INTERRUPT_UART_J)	1 - 18	286
AON - TKE	aon_tke_wdt_remoteinterrupt	(SPE_WDT_REMOTE)	none	15
AON - HSP Mbox	aon2licmbox_int_out[0]	(AON_MBOX_OUT_0)	none	133
AON - HSP Mbox	aon2licmbox_int_out[1]	(AON_MBOX_OUT_1)	none	134
AON - HSP Mbox	aon2licmbox_int_out[2]	(AON_MBOX_OUT_2)	none	135
AON - HSP Mbox	aon2licmbox_int_out[3]	(AON_MBOX_OUT_3)	none	136
AON - GTE	aon2lic_gteint	(AON_GTE)	none	13
AON - GPIO	gpioaon_lic_rupt[0]	(AON_GPIO_0)	none	56
AON - GPIO	gpioaon_lic_rupt[1]	(AON_GPIO_1)	none	57
AON - GPIO	gpioaon_lic_rupt[2]	(AON_GPIO_2)	none	58

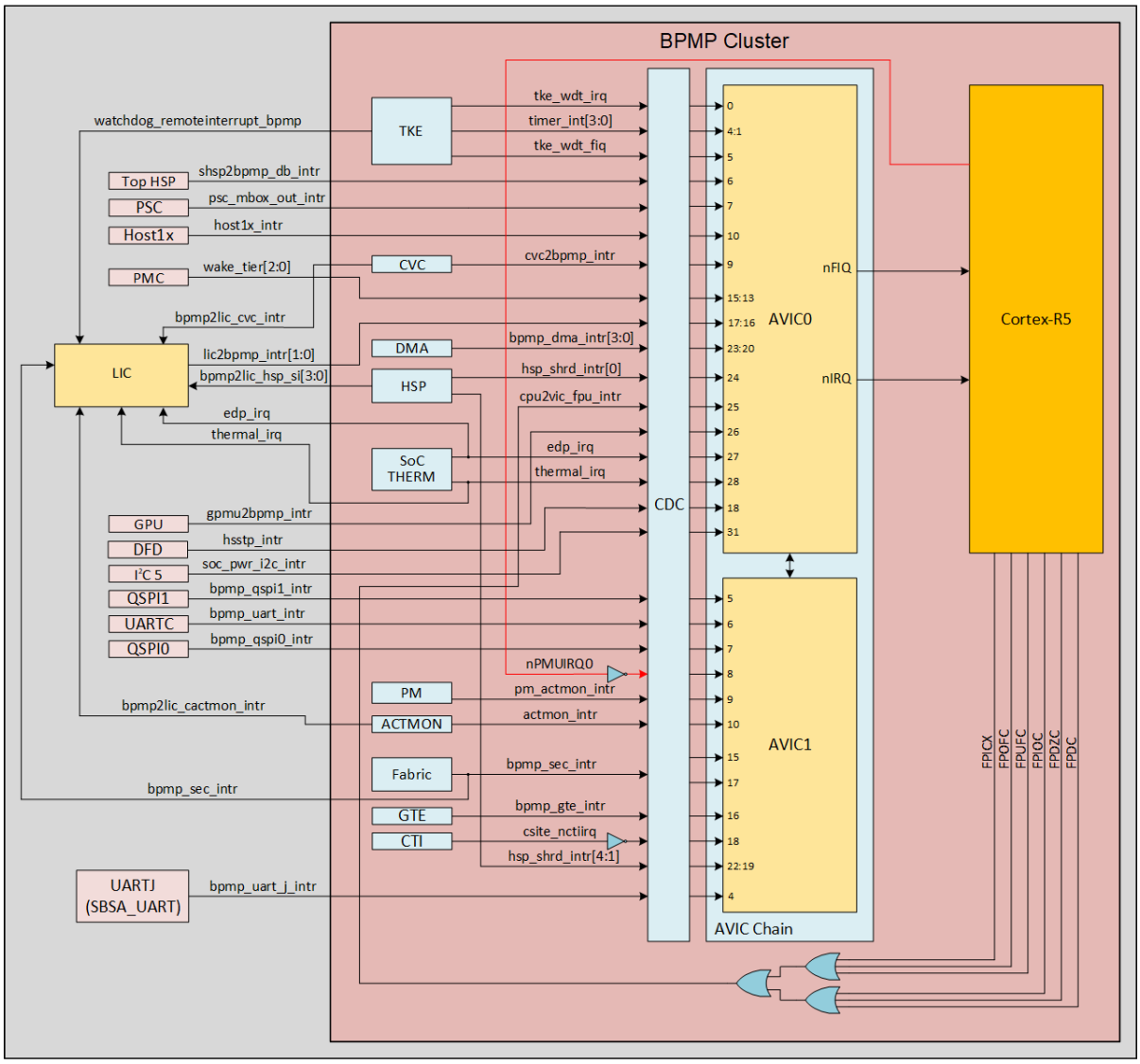
Source	Interrupt Name	Software Name	AVIC Connection	LIC Connection
AON - GPIO	gpioaon_lic_rupt[3]	(AON_GPIO_3)	none	59
AON - Wake	wake_tier0 / aon2lic_tier0wake_int	(AON_WAKE_1)	none	213
AON - Wake	wake_tier0 / aon2lic_tier0wake_int	(AON_WAKE_2)	none	214
AON - PMC	pmc_pwr_intr	(PMIC_EXT)	none	209
AON - PMC	pmc2lic_intr	(PMC2LIC_INTR)	none	211
AON - PMC	aon2lic_vfmon_int	(VFMON_INTR)	none	280
AON - PM	aon2lic_pmint	(AON_PM)	none	216
AON - RTC	aon2lic_rtcint	(RTC)	none	10
AON - RTC	aon2lic_aowdt_wakecold_intr	(AOWDT_REMOTE)	none	18
AON - TSC	tsc_irq	(TSC_IRQ)	none	74

For further information of the AON Cluster Interrupt controller, refer to the Always-On (AON) Cluster and SPE chapter of this TRM.

8.2.2.5.3 BPMP Cluster Interrupt Controller

The BPMP Cluster Interrupt controller with its AVIC Chain is shown in the diagram below.

Figure 8.4 BPMP Cluster Interrupt Structure



All the BPMP-related Interrupts are listed in the table below.

Table 8.6 BPMP Interrupts Connection

Source	Interrupt Name	Software Name	AVIC Connection	LIC Connection
BPMP - TKE	tke_wdt_irq	(BPMP_INTERRUPT_WDTIRQ)	0 - 0	none
BPMP - TKE	tke_timer_intr[0]	(BPMP_INTERRUPT_TIMER0)	0 - 1	none
BPMP - TKE	tke_timer_intr[1]	(BPMP_INTERRUPT_TIMER1)	0 - 2	none
BPMP - TKE	tke_timer_intr[2]	(BPMP_INTERRUPT_TIMER2)	0 - 3	none

Source	Interrupt Name	Software Name	AVIC Connection	LIC Connection
BPMP - TKE	tke_timer_intr[3]	(BPMP_INTERRUPT_TIMER3)	0 - 4	none
BPMP - TKE	tke_wdt_fiq	(BPMP_INTERRUPT_WDTFIQ)	0 - 5	none
top HSP	shsp2bpmp_db_intr	(BPMP_INTERRUPT_HSP_DB)	0 - 6	none
PSC	psc_mbox_out_intr[0]	(BPMP_INTERRUPT_PSC)	0 - 7	
BPMP - CVC	cvc2bpmp_intr	(BPMP_INTERRUPT_CVC)	0 - 9	none
Host Controller	host1x_intr	(BPMP_INTERRUPT_H1XSNC)	0 - 10	none
AON - PMC	wake_tier[0]	(BPMP_INTERRUPT_WAKE0)	0 - 13	none
AON - PMC	wake_tier[1]	(BPMP_INTERRUPT_WAKE1)	0 - 14	none
AON - PMC	wake_tier[2]	(BPMP_INTERRUPT_WAKE2)	0 - 15	none
LIC	lic2bpmp_intr[0]	(BPMP_INTERRUPT_LIC0)	0 - 16	none
LIC	lic2bpmp_intr[1]	(BPMP_INTERRUPT_LIC1)	0 - 17	none
DFD	hsstp_intr	(BPMP_INTERRUPT_HSSTP)	0 - 18	none
BPMP - DMA	bpmp_dma_intr[0]	(BPMP_INTERRUPT_DMA0)	0 - 20	none
BPMP - DMA	bpmp_dma_intr[1]	(BPMP_INTERRUPT_DMA1)	0 - 21	none
BPMP - DMA	bpmp_dma_intr[2]	(BPMP_INTERRUPT_DMA2)	0 - 22	none
BPMP - DMA	bpmp_dma_intr[3]	(BPMP_INTERRUPT_DMA3)	0 - 23	none
BPMP - HSP	hsp_shrd_intr[0]	(BPMP_INTERRUPT_HSP_SI_0)	0 - 24	none
BPMP - R5 FPU	cpu2vic_fpu_intr	(BPMP_INTERRUPT_FPUINT	0 - 25	none
GPU	gpmu2bpmp_intr	(BPMP_INTERRUPT_GPMU)	0 - 26	none
BPMP - SoC Therm	edp_irq (AVIC) / edp_irq (LIC)	(BPMP_INTERRUPT_EDP (AVIC) / EDP (LIC))	0 - 27	219
BPMP - SoC Therm	thermal_irq (AVIC) / thermal_irq (LIC)	(BPMP_INTERRUPT_SCTHRM (AVIC) / THERMAL (LIC))	0 - 28	218
I2C5	soc_pwr_i2c_intr	(BPMP_INTERRUPT_SOCPWRI2C (AVIC) / I2C5 (LIC))	0 - 31	29
UART_J	bpmp_uart_j_intr	(BPMP_INTERRUPT_UART_J)	1 - 4	none
QSPI1	bpmp_qspi1_intr	(BPMP_INTERRUPT_QSPI1 (AVIC) / QSPI1 (LIC))	1 - 5	39
AON - UARTC	bpmp_uart_intr (AVIC) / aon2lic_uartcint (LIC)	(BPMP_INTERRUPT_UART (AVIC) / UARTC (LIC))	1 - 6	114
QSPI0	bpmp_qspi0_intr	(BPMP_INTERRUPT_QSPI0 (AVIC) / QSPI0 (LIC))	1 - 7	35

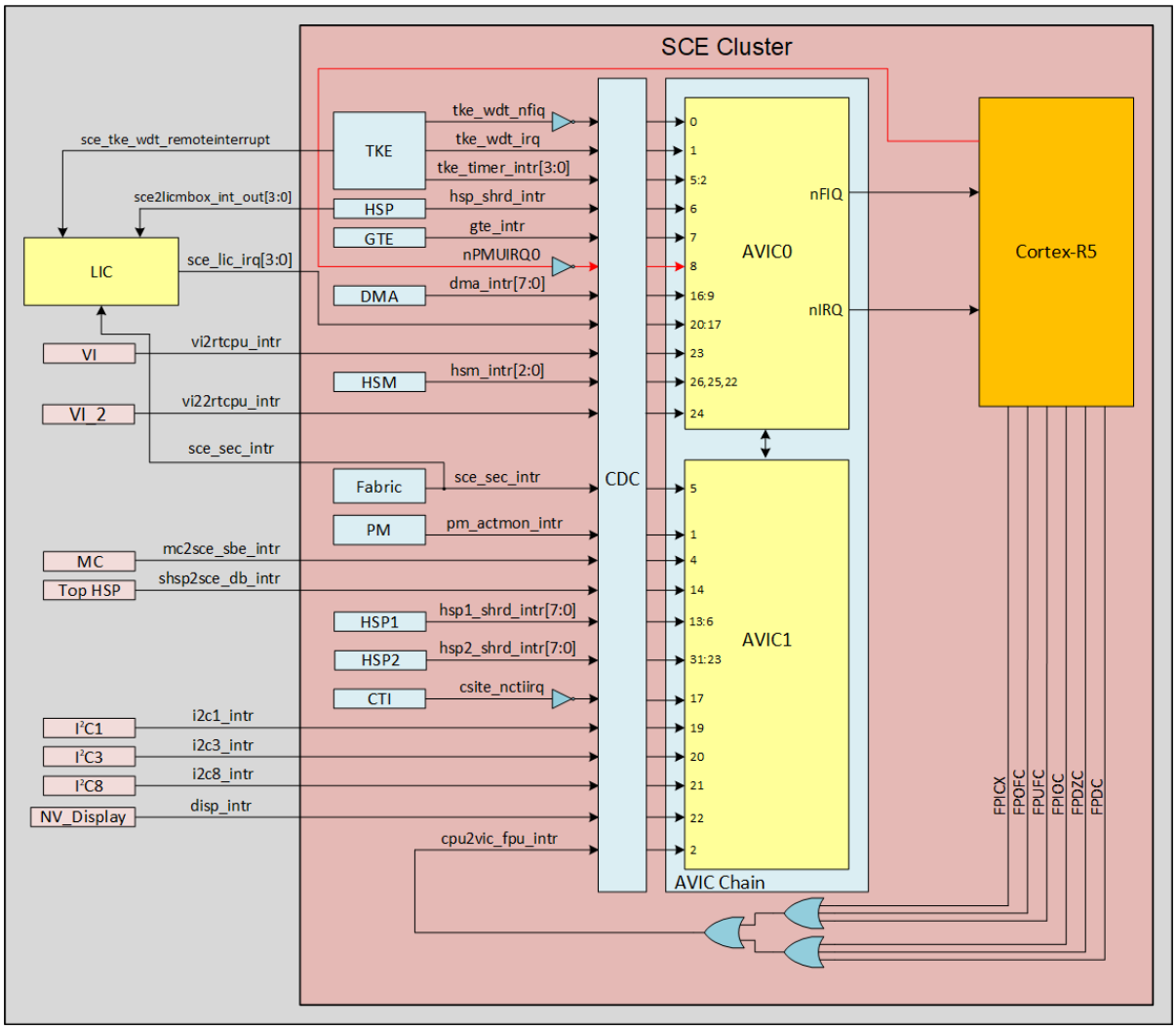
Source	Interrupt Name	Software Name	AVIC Connection	LIC Connection
BPMP - R5	nPMUIRQ0	(BPMP_INTERRUPT_PMUIRQ)	1 - 8	none
BPMP - ACTMON	actmon_intr	(MP_INTERRUPT_CNTRL_ACTMON)	1 - 9	none
BPMP - PM	pm_actmon_intr	(PMP_INTERRUPT_BPMP_ACTMON)	1 - 10	none
BPMP - GTE	bpmp_gte_intr	(BPMP_INTERRUPT_GTE)	1 - 16	none
BPMP - NOC	noc_sec_intr (AVIC) / bpmp2lic_noc_secure_intr (LIC)	(BPMP_INTERRUPT_NOC_SECURE (AVIC) / SECURE_BPMP_FABRIC (LIC))	1 - 17	174
BPMP - CTI	csite_nctiirq	(BPMP_INTERRUPT_CTIIRQ)	1 - 18	none
BPMP - HSP	hsp_shrd_intr[1]	(BPMP_INTERRUPT_HSP_SI_1)	1 - 19	none
BPMP - HSP	hsp_shrd_intr[2]	(BPMP_INTERRUPT_HSP_SI_2)	1 - 20	none
BPMP - HSP	hsp_shrd_intr[3]	(BPMP_INTERRUPT_HSP_SI_3)	1 - 21	none
BPMP - HSP	hsp_shrd_intr[4]	(BPMP_INTERRUPT_HSP_SI_4)	1 - 22	none
CAR	bpmp_car_intr	(BPMP_INTERRUPT_CAR)	1 - 23	none
BPMP - TKE	watchdog_remoteinterrupt_bpmp	(BPMP_WDT_REMOTE)	none	14
BPMP - CVC	bpmp2lic_cvc_intr	(CVC)	none	208
BPMP - HSP	bpmp2lic_hsp_si[0]	(BPMP_HSP_SHARED_1)	none	137
BPMP - HSP	bpmp2lic_hsp_si[1]	(BPMP_HSP_SHARED_2)	none	138
BPMP - HSP	bpmp2lic_hsp_si[2]	(BPMP_HSP_SHARED_3)	none	139
BPMP - HSP	bpmp2lic_hsp_si[3]	(BPMP_HSP_SHARED_4)	none	140
BPMP - ACTMON	bpmp2lic_cactmon_intr	(ACTMON)	none	210

For further information of the BPMP Cluster Interrupt controller, refer to the Boot and Power Management Processor chapter of this TRM.

8.2.2.5.4 SCE Cluster Interrupt Controller

The SCE Cluster Interrupt controller with its AVIC Chain is shown in the diagram below.

Figure 8.5 SCE Cluster Interrupt Structure



All the SCE-related Interrupts are listed in the table below.

Table 8.7 SCE Interrupts Connection

Source	Interrupt Name	Software Name	AVIC Connection	LIC Connection	FSI - GIC Connection
SCE - TKE	tke_wdt_nfiq	NV_SCE_INTERRUPT_WDTFIQ	0 - 0	none	none
SCE - TKE	tke_wdt_irq	NV_SCE_INTERRUPT_WDTIRQ	0 - 1	none	none
SCE - TKE	tke_timer_intr[0]	NV_SCE_INTERRUPT_TIMER0	0 - 2	none	none
SCE - TKE	tke_timer_intr[1]	NV_SCE_INTERRUPT_TIMER1	0 - 3	none	none
SCE - TKE	tke_timer_intr[2]	NV_SCE_INTERRUPT_TIMER2	0 - 4	none	none

Source	Interrupt Name	Software Name	AVIC Connection	LIC Connection	FSI - GIC Connection
SCE - TKE	tke_timer_intr[3]	NV_SCE_INTERRUPT_TIMER3	0 - 5	none	none
SCE - HSP	hsp_shrd_intr	NV_SCE_INTERRUPT_MBOX	0 - 6	none	none
SCE - GTE	gte_intr	NV_SCE_INTERRUPT_GTE	0 - 7	none	none
SCE - R5 PMU	nPMUIRQ0	NV_SCE_INTERRUPT_PMU	0 - 8	none	none
SCE - DMA	dma_intr[0]	NV_SCE_INTERRUPT_DMA0	0 - 9	none	none
SCE - DMA	dma_intr[1]	NV_SCE_INTERRUPT_DMA1	0 - 10	none	none
SCE - DMA	dma_intr[2]	NV_SCE_INTERRUPT_DMA2	0 - 11	none	none
SCE - DMA	dma_intr[3]	NV_SCE_INTERRUPT_DMA3	0 - 12	none	none
SCE - DMA	dma_intr[4]	NV_SCE_INTERRUPT_DMA4	0 - 13	none	none
SCE - DMA	dma_intr[5]	NV_SCE_INTERRUPT_DMA5	0 - 14	none	none
SCE - DMA	dma_intr[6]	NV_SCE_INTERRUPT_DMA6	0 - 15	none	none
SCE - DMA	dma_intr[7]	NV_SCE_INTERRUPT_DMA7	0 - 16	none	none
LIC	sce_lic_irq[0]	NV_SCE_INTERRUPT_LIC0	0 - 17	none	none
LIC	sce_lic_irq[1]	NV_SCE_INTERRUPT_LIC1	0 - 18	none	none
LIC	sce_lic_irq[2]	NV_SCE_INTERRUPT_LIC2	0 - 19	none	none
LIC	sce_lic_irq[3]	NV_SCE_INTERRUPT_LIC3	0 - 20	none	none
SCE - HSM	hsm_intr[0]	NV_SCE_INTERRUPT_HSM_CRITICAL_ERR	0 - 22	none	none
VI	vi2rtcpu_intr	NV_SCE_INTERRUPT_VI_HP	0 - 23	none	none
VI	vi22rtcpu_intr	NV_SCE_INTERRUPT_VI2_HP	0 - 24	none	none
SCE - HSM	hsm_intr[1]	NV_SCE_INTERRUPT_HSM_HP	0 - 25	none	none
SCE - HSM	hsm_intr[2]	NV_SCE_INTERRUPT_HSM_LP	0 - 26	none	none
SCE - PM	pm_actmon_intr	NV_SCE_INTERRUPT_ACTMON	1 - 1	none	none
SCE - R5 FPU	cpu2vic_fpu_intr	NV_SCE_INTERRUPT_FPUINT	1 - 2	none	none
SCE - PM	pm2vic_intr	NV_SCE_INTERRUPT_PM	1 - 3	none	none
MC - SBE	mc2sce_sbe_intr	NV_SCE_INTERRUPT_MC_SBE	1 - 4	none	none
SCE - NOC	noc_sec_intr (AVIC) / noc_sec_intr (LIC)	NV_SCE_INTERRUPT_NOC_SECURE (AVIC) / SECURE_SCE_FABRIC (LIC)	1 - 5	173	none

Source	Interrupt Name	Software Name	AVIC Connection	LIC Connection	FSI - GIC Connection
SCE - HSP1	hsp1_shrd_intr[0]	NV_SCE_INTERRUPT_HSP1_SI_0 (AVIC) / NV_FSI_GIC_SCE_HSP1_SI0 (FSI GIC)	1 - 6	none	13
SCE - HSP1	hsp1_shrd_intr[1]	NV_SCE_INTERRUPT_HSP1_SI_1 (AVIC) / NV_FSI_GIC_SCE_HSP1_SI1 (FSI GIC)	1 - 7	none	14
SCE - HSP1	hsp1_shrd_intr[2]	NV_SCE_INTERRUPT_HSP1_SI_2 (AVIC) / NV_FSI_GIC_SCE_HSP1_SI2 (FSI GIC)	1 - 8	none	15
SCE - HSP1	hsp1_shrd_intr[3]	NV_SCE_INTERRUPT_HSP1_SI_3 (AVIC) / NV_FSI_GIC_SCE_HSP1_SI3 (FSI GIC)	1 - 9	none	16
SCE - HSP1	hsp1_shrd_intr[4]	NV_SCE_INTERRUPT_HSP1_SI_4 (AVIC) / NV_FSI_GIC_SCE_HSP1_SI4 (FSI GIC)	1 - 10	none	36
SCE - HSP1	hsp1_shrd_intr[5]	NV_SCE_INTERRUPT_HSP1_SI_5 (AVIC) / NV_FSI_GIC_SCE_HSP1_SI5 (FSI GIC)	1 - 11	none	37
SCE - HSP1	hsp1_shrd_intr[6]	NV_SCE_INTERRUPT_HSP1_SI_6 (AVIC) / NV_FSI_GIC_SCE_HSP1_SI6 (FSI GIC)	1 - 12	none	38
SCE - HSP1	hsp1_shrd_intr[7]	NV_SCE_INTERRUPT_HSP1_SI_7 (AVIC) / NV_FSI_GIC_SCE_HSP1_SI7 (FSI GIC)	1 - 13	none	39
HSP	shsp2sce_db_intr	NV_SCE_INTERRUPT_TOP0_HSP_DB	1 - 14	none	none
CAR	sce_car_intr	NV_SCE_INTERRUPT_CAR	1 - 15	none	none
SCE - CTI	csite_nctiirq	NV_SCE_INTERRUPT_CTIIRQ	1 - 17	none	none
I2C1	i2c1_intr (AVIC) / i2c1_rupt (LIC)	NV_SCE_INTERRUPT_I2C1 (AVIC) / I2C (LIC)	1 - 19	25	none
I2C3	i2c3_intr (AVIC) / i2c3_rupt (LIC)	NV_SCE_INTERRUPT_I2C3 (AVIC) / I2C3 (LIC)	1 - 20	27	none
AON - I2C8	i2c8_intr (AVIC) / aon2lic_i2c3int (LIC)	NV_SCE_INTERRUPT_I2C8 (AVIC) / I2C8 (LIC)	1 - 21	32	none
SCE - HSP2	hsp2_shrd_intr[0]	NV_SCE_INTERRUPT_HSP2_SI_0 (AVIC) / NV_FSI_GIC_SCE_HSP2_SI0 (FSI GIC)	1 - 24	none	40
SCE - HSP2	hsp2_shrd_intr[1]	NV_SCE_INTERRUPT_HSP2_SI_1 (AVIC) / NV_FSI_GIC_SCE_HSP2_SI1 (FSI GIC)	1 - 25	none	41
SCE - HSP2	hsp2_shrd_intr[2]	NV_SCE_INTERRUPT_HSP2_SI_2 (AVIC) / NV_FSI_GIC_SCE_HSP2_SI2 (FSI GIC)	1 - 26	none	42

Source	Interrupt Name	Software Name	AVIC Connection	LIC Connection	FSI - GIC Connection
SCE - HSP2	hsp2_shrd_intr[3]	NV_SCE_INTERRUPT_HSP2_SI_3 (AVIC) / NV_FSI_GIC_SCE_HSP2_SI3 (FSI GIC)	1 - 27	none	43
SCE - HSP2	hsp2_shrd_intr[4]	NV_SCE_INTERRUPT_HSP2_SI_4 (AVIC) / NV_FSI_GIC_SCE_HSP2_SI4 (FSI GIC)	1 - 28	none	17
SCE - HSP2	hsp2_shrd_intr[5]	NV_SCE_INTERRUPT_HSP2_SI_5 (AVIC) / NV_FSI_GIC_SCE_HSP2_SI5 (FSI GIC)	1 - 29	none	18
SCE - HSP2	hsp2_shrd_intr[6]	NV_SCE_INTERRUPT_HSP2_SI_6 (AVIC) / NV_FSI_GIC_SCE_HSP2_SI6 (FSI GIC)	1 - 30	none	19
SCE - HSP2	hsp2_shrd_intr[7]	NV_SCE_INTERRUPT_HSP2_SI_7 (AVIC) / NV_FSI_GIC_SCE_HSP2_SI7 (FSI GIC)	1 - 31	none	20
SCE - HSP	sce2licmbox_int_out[0]	SCE_MBOX_OUT_0	none	141	none
SCE - HSP	sce2licmbox_int_out[1]	SCE_MBOX_OUT_1	none	142	none
SCE - HSP	sce2licmbox_int_out[2]	SCE_MBOX_OUT_2	none	143	none
SCE - HSP	sce2licmbox_int_out[3]	SCE_MBOX_OUT_3	none	144	none
SCE - TKE	sce_tke_wdt_remoteinterrupt	SCE_WDT_REMOTE	none	16	none

For further information of the SCE Cluster Interrupt controller, refer to the Safety Cluster Engine (SCE) chapter of this TRM.

8.2.2.5.5 RCE Cluster Interrupt Controller

The RCE is a duplicate instance of the Safety Cluster Engine (SCE) module; internally both SCE and RCE are the same, however, some I/O connections are different.

All the RCE-related Interrupts are listed in the table below.

Table 8.8 RCE Interrupts Connection

Source	Interrupt Name	Software Name	AVIC Connection	LIC Connection
RCE - TKE	tke_wdt_nfiq	NV_RCE_INTERRUPT_WDTFIQ	0 - 0	none
RCE - TKE	tke_wdt_irq	NV_RCE_INTERRUPT_WDTIRQ	0 - 1	none
RCE - TKE	tke_timer_intr[0]	NV_RCE_INTERRUPT_TIMER0	0 - 2	none
RCE - TKE	tke_timer_intr[1]	NV_RCE_INTERRUPT_TIMER1	0 - 3	none

Source	Interrupt Name	Software Name	AVIC Connection	LIC Connection
RCE - TKE	tke_timer_intr[2]	NV_RCE_INTERRUPT_TIMER2	0 - 4	none
RCE - TKE	tke_timer_intr[3]	NV_RCE_INTERRUPT_TIMER3	0 - 5	none
RCE - TKE	hsp_shrd_intr	NV_RCE_INTERRUPT_MBOX	0 - 6	none
RCE - TKE	gte_intr	NV_RCE_INTERRUPT_GTE	0 - 7	none
RCE - TKE	nPMUIRQ0	NV_RCE_INTERRUPT_PMU	0 - 8	none
RCE - TKE	dma_intr[0]	NV_RCE_INTERRUPT_DMA0	0 - 9	none
RCE - TKE	dma_intr[1]	NV_RCE_INTERRUPT_DMA1	0 - 10	none
RCE - TKE	dma_intr[2]	NV_RCE_INTERRUPT_DMA2	0 - 11	none
RCE - TKE	dma_intr[3]	NV_RCE_INTERRUPT_DMA3	0 - 12	none
RCE - TKE	dma_intr[4]	NV_RCE_INTERRUPT_DMA4	0 - 13	none
RCE - TKE	dma_intr[5]	NV_RCE_INTERRUPT_DMA5	0 - 14	none
RCE - TKE	dma_intr[6]	NV_RCE_INTERRUPT_DMA6	0 - 15	none
RCE - TKE	dma_intr[7]	NV_RCE_INTERRUPT_DMA7	0 - 16	none
RCE - TKE	rce_lic_irq[0]	NV_RCE_INTERRUPT_LIC0	0 - 17	none
RCE - TKE	rce_lic_irq[1]	NV_RCE_INTERRUPT_LIC1	0 - 18	none
RCE - TKE	rce_lic_irq[2]	NV_RCE_INTERRUPT_LIC2	0 - 19	none
RCE - TKE	rce_lic_irq[3]	NV_RCE_INTERRUPT_LIC3	0 - 20	none
RCE - TKE	vi2rtcpu_intr	NV_RCE_INTERRUPT_VI_HP	0 - 23	none
RCE - TKE	vi22rtcpu_intr	NV_RCE_INTERRUPT_VI2_HP	0 - 24	none
RCE - TKE	pm_actmon_intr	NV_RCE_INTERRUPT_ACTMON	1 - 1	none
RCE - TKE	cpu2vic_fpu_intr	NV_RCE_INTERRUPT_FPUINT	1 - 2	none
RCE - TKE	pm2vic_intr	NV_RCE_INTERRUPT_PM	1 - 3	none
RCE - TKE	mc2rce_sbe_intr	NV_RCE_INTERRUPT_MC_SBE	1 - 4	none
RCE - TKE	noc_sec_intr (AVIC) / noc_sec_intr (LIC)	NV_RCE_INTERRUPT_NOC_SECURE (AVIC) / SECURE_RCE_FABRIC (LIC)	1 - 5	175
RCE - TKE	shsp2rce_db_intr	NV_RCE_INTERRUPT_TOP0_HSP_DB	1 - 14	none
RCE - TKE	rce_car_intr	NV_RCE_INTERRUPT_CAR	1 - 15	none
RCE - TKE	csite_nctiirq	NV_RCE_INTERRUPT_CTIIRQ	1 - 17	none
RCE - TKE	i2c1_intr (AVIC) / i2c1_rupt (LIC)	NV_RCE_INTERRUPT_I2C1 (AVIC) / I2C (LIC)	1 - 19	25

Source	Interrupt Name	Software Name	AVIC Connection	LIC Connection
RCE - TKE	i2c3_intr (AVIC) / i2c3_rupt (LIC)	NV_RCE_INTERRUPT_I2C3 (AVIC) / I2C3 (LIC)	1 - 20	27
RCE - TKE	i2c8_intr (AVIC) / aon2lic_i2c3int (LIC)	NV_RCE_INTERRUPT_I2C8 (AVIC) / I2C8 (LIC)	1 - 21	32
RCE - TKE	rce2licmbox_int_out[0]	RCE_MBOX_OUT_0	none	182
RCE - TKE	rce2licmbox_int_out[1]	RCE_MBOX_OUT_1	none	183
RCE - TKE	rce2licmbox_int_out[2]	RCE_MBOX_OUT_2	none	184
RCE - TKE	rce2licmbox_int_out[3]	RCE_MBOX_OUT_3	none	185
RCE - TKE	rce_tke_wdt_remoteinterrupt	RCE_WDT_REMOTE	none	19

For further information of the RCE Cluster Interrupt controller, refer to the RCE chapter of this TRM.

8.2.2.5.6 DCE Cluster Interrupt Controller

The DCE is a duplicate instance of the Safety Cluster Engine (SCE) module; internally both SCE and DCE are the same, however, some I/O connections are different.

All the DCE-related Interrupts are listed in the table below.

Table 8.9 DCE Interrupts Connection

Source	Interrupt Name	Software Name	AVIC Connection	LIC Connection
DCE - TKE	tke_wdt_nfiq	NV_DCE_INTERRUPT_WDTFIQ	0 - 0	none
DCE - TKE	tke_wdt_irq	NV_DCE_INTERRUPT_WDTIRQ	0 - 1	none
DCE - TKE	tke_timer_intr[0]	NV_DCE_INTERRUPT_TIMER0	0 - 2	none
DCE - TKE	tke_timer_intr[1]	NV_DCE_INTERRUPT_TIMER1	0 - 3	none
DCE - TKE	tke_timer_intr[2]	NV_DCE_INTERRUPT_TIMER2	0 - 4	none
DCE - TKE	tke_timer_intr[3]	NV_DCE_INTERRUPT_TIMER3	0 - 5	none
DCE - TKE	hsp_shrd_intr	NV_DCE_INTERRUPT_MBOX	0 - 6	none
DCE - TKE	gte_intr	NV_DCE_INTERRUPT_GTE	0 - 7	none
DCE - TKE	nPMUIRQ0	NV_DCE_INTERRUPT_PMU	0 - 8	none
DCE - TKE	dma_intr[0]	NV_DCE_INTERRUPT_DMA0	0 - 9	none
DCE - TKE	dma_intr[1]	NV_DCE_INTERRUPT_DMA1	0 - 10	none

Source	Interrupt Name	Software Name	AVIC Connection	LIC Connection
DCE - TKE	dma_intr[2]	NV_DCE_INTERRUPT_DMA2	0 - 11	none
DCE - TKE	dma_intr[3]	NV_DCE_INTERRUPT_DMA3	0 - 12	none
DCE - TKE	dma_intr[4]	NV_DCE_INTERRUPT_DMA4	0 - 13	none
DCE - TKE	dma_intr[5]	NV_DCE_INTERRUPT_DMA5	0 - 14	none
DCE - TKE	dma_intr[6]	NV_DCE_INTERRUPT_DMA6	0 - 15	none
DCE - TKE	dma_intr[7]	NV_DCE_INTERRUPT_DMA7	0 - 16	none
DCE - TKE	dce_lic_irq[0]	NV_DCE_INTERRUPT_LIC0	0 - 17	none
DCE - TKE	dce_lic_irq[1]	NV_DCE_INTERRUPT_LIC1	0 - 18	none
DCE - TKE	dce_lic_irq[2]	NV_DCE_INTERRUPT_LIC2	0 - 19	none
DCE - TKE	dce_lic_irq[3]	NV_DCE_INTERRUPT_LIC3	0 - 20	none
DCE - TKE	vi2rtcpu_intr	NV_DCE_INTERRUPT_VI_HP	0 - 23	none
DCE - TKE	vi22rtcpu_intr	NV_DCE_INTERRUPT_VI2_HP	0 - 24	none
DCE - TKE	pm_actmon_intr	NV_DCE_INTERRUPT_ACTMON	1 - 1	none
DCE - TKE	cpu2vic_fpu_intr	NV_DCE_INTERRUPT_FPUINT	1 - 2	none
DCE - TKE	pm2vic_intr	NV_DCE_INTERRUPT_PM	1 - 3	none
DCE - TKE	mc2dce_sbe_intr	NV_DCE_INTERRUPT_MC_SBE	1 - 4	none
DCE - TKE	noc_sec_intr (AVIC) / noc_sec_intr (LIC)	NV_DCE_INTERRUPT_NOC_SECURE (AVIC) / SECURE_DCE_FABRIC (LIC)	1 - 5	381
DCE - TKE	shsp2dce_db_intr	NV_DCE_INTERRUPT_TOP0_HSP_DB	1 - 14	none
DCE - TKE	dce_car_intr	NV_DCE_INTERRUPT_CAR	1 - 15	none
DCE - TKE	csite_nctiirq	NV_DCE_INTERRUPT_CTIIRQ	1 - 17	none
DCE - TKE	i2c1_intr (AVIC) / i2c1_rupt (LIC)	NV_DCE_INTERRUPT_I2C1 (AVIC) / I2C (LIC)	1 - 19	25
DCE - TKE	i2c3_intr (AVIC) / i2c3_rupt (LIC)	NV_DCE_INTERRUPT_I2C3 (AVIC) / I2C3 (LIC)	1 - 20	27
DCE - TKE	i2c8_intr (AVIC) / aon2lic_i2c3int (LIC)	NV_DCE_INTERRUPT_I2C8 (AVIC) / I2C8 (LIC)	1 - 21	32
DISPLAY	disp_2gsp_intr_intr	NV_DCE_INTERRUPT_DISP	1 - 22	417
DCE - TKE	dce2licmbox_int_out[0]	DCE_MBOX_OUT_0	none	377
DCE - TKE	dce2licmbox_int_out[1]	DCE_MBOX_OUT_1	none	378
DCE - TKE	dce2licmbox_int_out[2]	DCE_MBOX_OUT_2	none	379

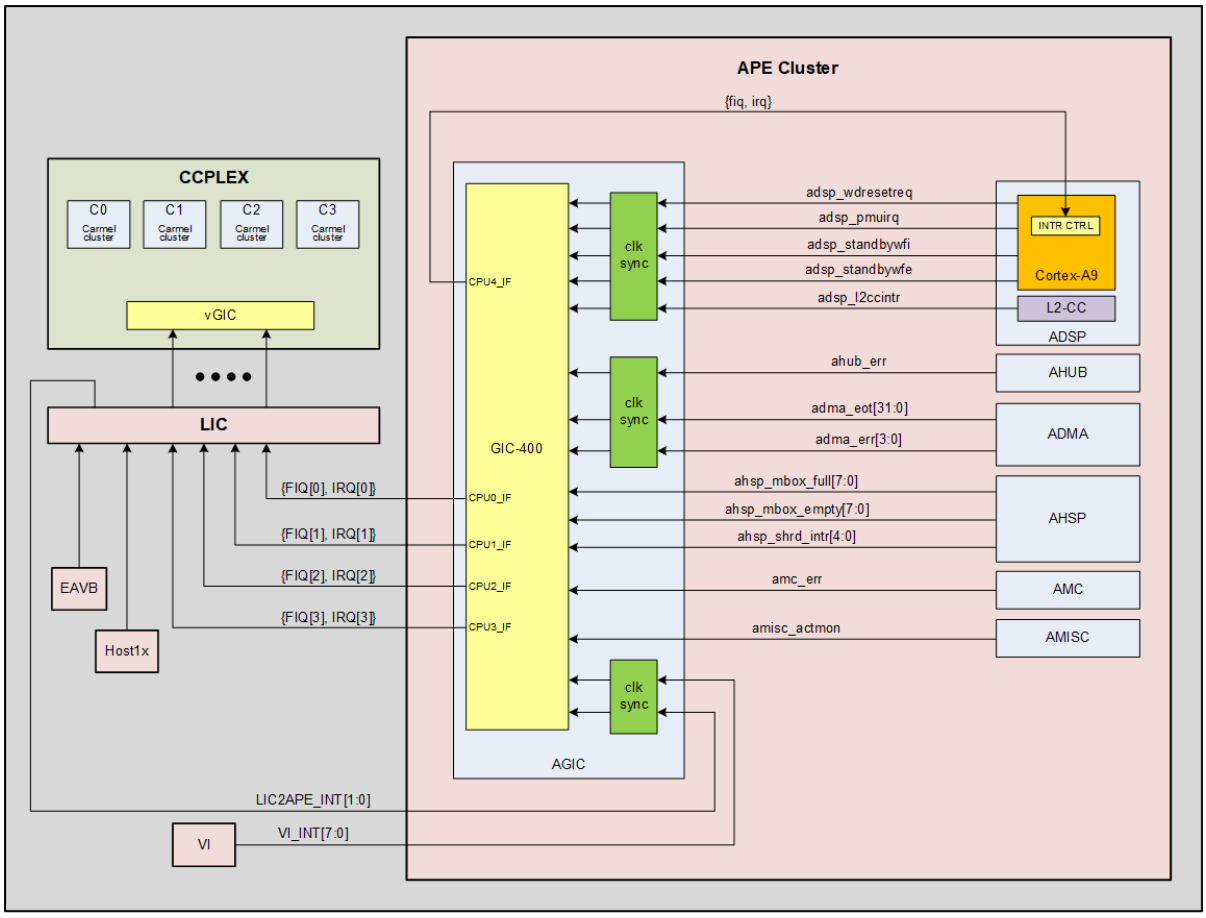
Source	Interrupt Name	Software Name	AVIC Connection	LIC Connection
DCE - TKE	dce2licmbox_int_out[3]	DCE_MBOX_OUT_3	none	380
DCE - TKE	dce_tke_wdt_remoteinterrupt	DCE_WDT_REMOTE	none	376

For further information of the DCE Cluster Interrupt controller, refer to the DCE chapter of this TRM.

8.2.2.5.7 APE Cluster Interrupt Controller

The APE Interrupt controllers together with Interrupt signals routing are shown in the diagram below.

Figure 8.6 APE Cluster Interrupt Structure



There are two Interrupt controllers in the APE, the local Interrupt controller in ADSP and the APE Generic Interrupt controller (AGIC). The APE Interrupts are routed to AGIC then aggregated to irq

and fiq that get sent to the ADSP local Interrupt controller's IRQ/FIQ input. The signals are configured as legacy mode, that is, IRQ as PPI(4) and FIQ as PPI(1). Other Interrupts in the ADSP local Interrupt controller are global timer PPI(0), private timer PPI(1), and watchdog timer PPI(3). The ADSP local Interrupt controller does not have any SGI or SPI Interrupts configured.

ADSP Interrupt Handling

ADSP's global timer PPI(0), private timer PPI(1), and watchdog timer PPI(3) Interrupts are not sent to the AGIC or CCPLEX. To service these Interrupts, the ADSP only needs to access the registers within the local Interrupt controller.

ADSP IRQ/FIQ Interrupts involve two Interrupt controllers, the ADSP local Interrupt controller and AGIC. ADSP reads from its local Interrupt controller then from AGIC to determine the Interrupt source. If the Interrupt source can be identified as from AGIC Interrupt acknowledge register, ADSP starts servicing the Interrupt. Some Interrupts, such as AHUB error Interrupt, require more Interrupt status access before the source can be identified. After servicing the Interrupts, ADSP writes end of Interrupt to AGIC and the local Interrupt controller to signal the completion of an Interrupt.

CCPLEX Interrupt Handling

There are four sets of AVIC-aggregated IRQ and FIQ Interrupt signals (IRQ[3:0] and FIQ[3:0]) from APE to CCPLEX. While the CCPLEX has no access to the ADSP local Interrupt controller, the Interrupt service by the processors (in CCPLEX) Interrupt service is similar to the ADSP's Interrupt handling, with the exception that they first read from the vGIC in CCPLEX and LIC to determine if a particular Interrupt is coming from APE.

All the APE-related Interrupts are listed in the table below.

Table 8.10 APE Interrupts Connection

Source	Interrupt Name	GIC Connection	LIC Connection
Processors	for Software-Generated Interrupts (SGI)	0 ~ 15	none
APE - ADMA	adma_eot[0]	32	none
APE - ADMA	adma_eot[1]	33	none
APE - ADMA	adma_eot[2]	34	none
APE - ADMA	adma_eot[3]	35	none
APE - ADMA	adma_eot[4]	36	none
APE - ADMA	adma_eot[5]	37	none
APE - ADMA	adma_eot[6]	38	none
APE - ADMA	adma_eot[7]	39	none

Source	Interrupt Name	GIC Connection	LIC Connection
APE - ADMA	adma_eot[8]	40	none
APE - ADMA	adma_eot[9]	41	none
APE - ADMA	adma_eot[10]	42	none
APE - ADMA	adma_eot[11]	43	none
APE - ADMA	adma_eot[12]	44	none
APE - ADMA	adma_eot[13]	45	none
APE - ADMA	adma_eot[14]	46	none
APE - ADMA	adma_eot[15]	47	none
APE - ADMA	adma_eot[16]	48	none
APE - ADMA	adma_eot[17]	49	none
APE - ADMA	adma_eot[18]	50	none
APE - ADMA	adma_eot[19]	51	none
APE - ADMA	adma_eot[20]	52	none
APE - ADMA	adma_eot[21]	53	none
APE - ADMA	adma_eot[22]	54	none
APE - ADMA	adma_eot[23]	55	none
APE - ADMA	adma_eot[24]	56	none
APE - ADMA	adma_eot[25]	57	none
APE - ADMA	adma_eot[26]	58	none
APE - ADMA	adma_eot[27]	59	none
APE - ADMA	adma_eot[28]	60	none
APE - ADMA	adma_eot[29]	61	none
APE - ADMA	adma_eot[30]	62	none
APE - ADMA	adma_eot[31]	63	none
APE - AHSP	ahsp_mbox_full[0]	64	none
APE - AHSP	ahsp_mbox_full[1]	65	none
APE - AHSP	ahsp_mbox_full[2]	66	none
APE - AHSP	ahsp_mbox_full[3]	67	none
APE - AHSP	ahsp_mbox_full[4]	68	none

Source	Interrupt Name	GIC Connection	LIC Connection
APE - AHSP	ahsp_mbox_full[5]	69	none
APE - AHSP	ahsp_mbox_full[6]	70	none
APE - AHSP	ahsp_mbox_full[7]	71	none
APE - AHSP	ahsp_mbox_empty[0]	72	none
APE - AHSP	ahsp_mbox_empty[1]	73	none
APE - AHSP	ahsp_mbox_empty[2]	74	none
APE - AHSP	ahsp_mbox_empty[3]	75	none
APE - AHSP	ahsp_mbox_empty[4]	76	none
APE - AHSP	ahsp_mbox_empty[5]	77	none
APE - AHSP	ahsp_mbox_empty[6]	78	none
APE - AHSP	ahsp_mbox_empty[7]	79	none
APE - AHSP	ashp_shrd_intr[0]	80	none
APE - AHSP	ashp_shrd_intr[1]	81	none
APE - AHSP	ashp_shrd_intr[2]	82	none
APE - AHSP	ashp_shrd_intr[3]	83	none
APE - AHSP	ashp_shrd_intr[4]	84	none
APE - ADSP	adsp_pmuirq	85	none
APE - ADSP	adsp_wdresetreq	86	none
APE - ADSP L2CC	adsp_l2ccintr	87	none
APE - AHUB	ahub_err	88	none
APE - AMC	amc_err	89	none
APE - ADMA	adma_err[0]	90	none
APE - ADMA	adma_err[1]	91	none
APE - ADMA	adma_err[2]	92	none
APE - ADMA	adma_err[3]	93	none
APE - ADSP	adsp_standbywfi	94	none
APE - ADSP	adsp_standbywfe	95	none
APE - ADSP	adsp_ctiirq	96	none
APE - AMISC	ape_actmon	97	none

Source	Interrupt Name	GIC Connection	LIC Connection
LIC	Lic2ape_int[0]	100	none
LIC	Lic2ape_int[1]	101	none
I2C1	i2c1_rupt	104	25
I2C3	i2c3_rupt	105	27
AON - I2C8	aon2lic_i2c3int	106	32
top HSP	shsp2ape_db	107	none
top WDT	top_wdt_fiq	108	none
top WDT	top_wdt_irq	109	none
APE - ATKE	atke_tmr_irq[0]	110	none
APE - ATKE	atke_tmr_irq[1]	111	none
APE - ATKE	atke_tmr_irq[2]	112	none
APE - ATKE	atke_tmr_irq[3]	113	none
APE - ATKE	atke_wdt_fiq	114	201
APE - ATKE	atke_wdt_irq	115	none
APE - ATKE	atke_wdt_error	116	none
APE - AGIC	IRQ[0]	none	145
APE - AGIC	IRQ[1]	none	146
APE - AGIC	IRQ[2]	none	147
APE - AGIC	IRQ[3]	none	148
APE - AGIC	FIQ[0]	none	149
APE - AGIC	FIQ[1]	none	150
APE - AGIC	FIQ[2]	none	151
APE - AGIC	FIQ[3]	none	152

For further information of the APE Interrupt controllers, refer to the Audio Processing Engine (APE) chapter of this TRM.

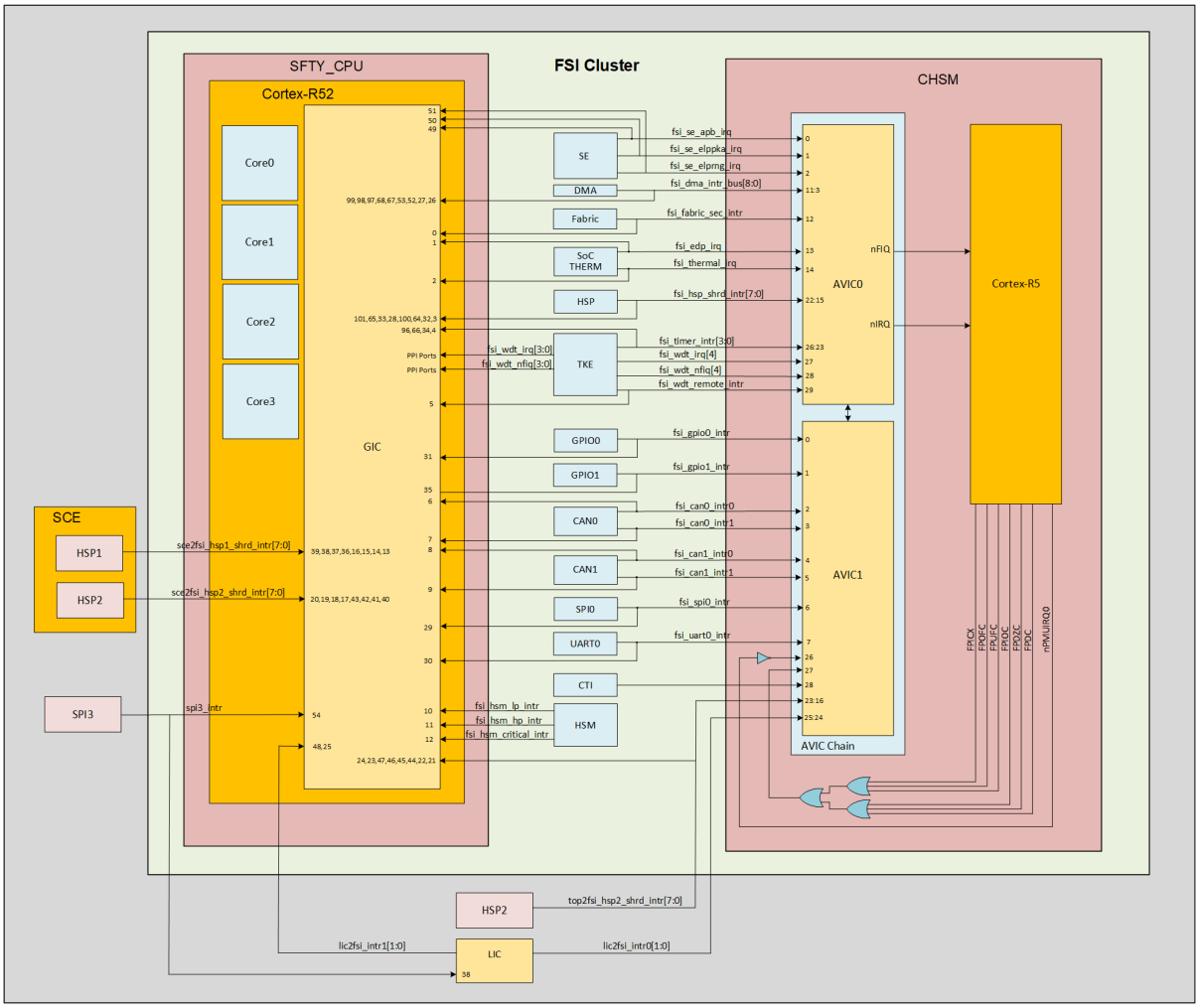
8.2.2.5.8 FSI Cluster Interrupt Controller

There are two interrupt controllers in FSI:

- The Cortex-R52 Processor has an integrated Generic Interrupt Controller (GIC)
- AVIC Chains for Cortex-R5 processor in CHSM

The FSI Cluster Interrupt controller with its AVIC Chain is shown in the diagram below.

Figure 8.7 FSI Cluster Interrupt Structure



All the FSI-related Interrupts are listed in the table below.

Table 8.11 FSI Interrupts Connection

Source	Interrupt Name	FSI AVIC Software Name	AVIC Connection	FSI GIC Software Name	FSI GIC Connection	LIC Connection
FSI - SE	fsi_se_apb_irq	FSI_AVIC0_SE_APB_INTR	0 - 0	NV_FSI_GIC_SE_APB_INTR	49	none
FSI - SE	fsi_se_elppka_irq	FSI_AVIC0_SE_ELPPKA_INTR	0 - 1	NV_FSI_GIC_SE_ELPPKA_INTR	50	none
FSI - SE	fsi_se_elprng_irq	FSI_AVIC0_SE_ELPRNG_INTR	0 - 2	NV_FSI_GIC_SE_ELPRNG_INTR	51	none

Source	Interrupt Name	FSI AVIC Software Name	AVIC Connection	FSI GIC Software Name	FSI GIC Connection	LIC Connection
FSI - DMA	fsi_dma_intr_bus[0]	NV_FSI_AVIC0_DMA_INTR0	0 - 3	NV_FSI_GIC_DMA_INTR0	26	none
FSI - DMA	fsi_dma_intr_bus[1]	NV_FSI_AVIC0_DMA_INTR1	0 - 4	NV_FSI_GIC_DMA_INTR1	27	none
FSI - DMA	fsi_dma_intr_bus[2]	NV_FSI_AVIC0_DMA_INTR2	0 - 5	NV_FSI_GIC_DMA_INTR2	52	none
FSI - DMA	fsi_dma_intr_bus[3]	NV_FSI_AVIC0_DMA_INTR3	0 - 6	NV_FSI_GIC_DMA_INTR3	53	none
FSI - DMA	fsi_dma_intr_bus[4]	NV_FSI_AVIC0_DMA_INTR4	0 - 7	NV_FSI_GIC_DMA_INTR4	67	none
FSI - DMA	fsi_dma_intr_bus[5]	NV_FSI_AVIC0_DMA_INTR5	0 - 8	NV_FSI_GIC_DMA_INTR5	68	none
FSI - DMA	fsi_dma_intr_bus[6]	NV_FSI_AVIC0_DMA_INTR6	0 - 9	NV_FSI_GIC_DMA_INTR6	97	none
FSI - DMA	fsi_dma_intr_bus[7]	NV_FSI_AVIC0_DMA_INTR7	0 - 10	NV_FSI_GIC_DMA_INTR7	98	none
FSI - DMA	fsi_dma_intr_bus[8]	NV_FSI_AVIC0_DMA_INTR8	0 - 11	NV_FSI_GIC_DMA_INTR8	99	none
FSI - FABRIC	fsi_fabric_sec_intr	NV_FSI_AVIC0_FABRIC_FUNC_INTR	0 - 12	NV_FSI_GIC_FABRIC_FUNC_INTR	0	none
FSI - THERM	fsi_edp_irq	NV_FSI_AVIC0_THERM_EDP_INTR	0 - 13	NV_FSI_GIC_THERM_EDP_INTR	1	none
FSI - THERM	fsi_thermal_irq	NV_FSI_AVIC0_THERM_IRQ_INTR	0 - 14	NV_FSI_GIC_THERM_IRQ_INTR	2	none
FSI - HSP	fsi_hsp_shrd_intr[0]	NV_FSI_AVIC0_HSP_SI0	0 - 15	NV_FSI_GIC_HSP_SI0	3	none
FSI - HSP	fsi_hsp_shrd_intr[1]	NV_FSI_AVIC0_HSP_SI1	0 - 16	NV_FSI_GIC_HSP_SI1	32	none
FSI - HSP	fsi_hsp_shrd_intr[2]	NV_FSI_AVIC0_HSP_SI2	0 - 17	NV_FSI_GIC_HSP_SI2	64	none
FSI - HSP	fsi_hsp_shrd_intr[3]	NV_FSI_AVIC0_HSP_SI3	0 - 18	NV_FSI_GIC_HSP_SI3	100	none
FSI - HSP	fsi_hsp_shrd_intr[4]	NV_FSI_AVIC0_HSP_SI4	0 - 19	NV_FSI_GIC_HSP_SI4	28	none
FSI - HSP	fsi_hsp_shrd_intr[5]	NV_FSI_AVIC0_HSP_SI5	0 - 20	NV_FSI_GIC_HSP_SI5	33	none
FSI - HSP	fsi_hsp_shrd_intr[6]	NV_FSI_AVIC0_HSP_SI6	0 - 21	NV_FSI_GIC_HSP_SI6	65	none

Source	Interrupt Name	FSI AVIC Software Name	AVIC Connection	FSI GIC Software Name	FSI GIC Connection	LIC Connection
FSI - HSP	fsi_hsp_shrd_intr[7]	NV_FSI_AVIC0_HSP_SI7	0 - 22	NV_FSI_GIC_HSP_SI7	101	none
FSI - TKE	fsi_timer_intr[0]	NV_FSI_AVIC0_TIMER_INTR0	0 - 23	NV_FSI_GIC_TIMER_INTR0	4	none
FSI - TKE	fsi_timer_intr[1]	NV_FSI_AVIC0_TIMER_INTR1	0 - 24	NV_FSI_GIC_TIMER_INTR1	34	none
FSI - TKE	fsi_timer_intr[2]	NV_FSI_AVIC0_TIMER_INTR2	0 - 25	NV_FSI_GIC_TIMER_INTR2	66	none
FSI - TKE	fsi_timer_intr[3]	NV_FSI_AVIC0_TIMER_INTR3	0 - 26	NV_FSI_GIC_TIMER_INTR3	96	none
FSI - TKE	fsi_wdt_irq[0]	none	none	NV_FSI_GIC_WDT_nIRQ0		none
FSI - TKE	fsi_wdt_irq[1]	none	none	NV_FSI_GIC_WDT_nIRQ1		none
FSI - TKE	fsi_wdt_irq[2]	none	none	NV_FSI_GIC_WDT_nIRQ2		none
FSI - TKE	fsi_wdt_irq[3]	none	none	NV_FSI_GIC_WDT_nIRQ3		none
FSI - TKE	fsi_wdt_irq[4]	NV_FSI_AVIC0_WDT_IRQ4	0 - 27	none	none	none
FSI - TKE	fsi_wdt_nfiq[0]	none	none	NV_FSI_GIC_WDT_nFIQ0		none
FSI - TKE	fsi_wdt_nfiq[1]	none	none	NV_FSI_GIC_WDT_nFIQ1		none
FSI - TKE	fsi_wdt_nfiq[2]	none	none	NV_FSI_GIC_WDT_nFIQ2		none
FSI - TKE	fsi_wdt_nfiq[3]	none	none	NV_FSI_GIC_WDT_nFIQ3		none
FSI - TKE	fsi_wdt_nfiq[4]	NV_FSI_AVIC0_WDT_nFIQ4	0 - 28	none	none	none
FSI - TKE	fsi_wdt_remote_intr	NV_FSI_AVIC0_WDT_REMOTE_INTR	0 - 29	NV_FSI_GIC_WDT_REMOTE_INTR	5	none
FSI - GPIO0	fsi_gpio0_intr	NV_FSI_AVIC1_GPIO0_INTR0	1 - 0	NV_FSI_GIC_GPIO0_INTR0	31	none
FSI - GPIO1	fsi_gpio1_intr	NV_FSI_AVIC1_GPIO1_INTR0	1 - 1	NV_FSI_GIC_GPIO1_INTR0	35	none
FSI - CAN0	fsi_can0_intr0	NV_FSI_AVIC1_CAN0_INTR0	1 - 2	NV_FSI_GIC_CAN0_INTR0	6	none
FSI - CAN0	fsi_can0_intr1	NV_FSI_AVIC1_CAN0_INTR1	1 - 3	NV_FSI_GIC_CAN0_INTR1	7	none
FSI - CAN1	fsi_can1_intr0	NV_FSI_AVIC1_CAN1_INTR0	1 - 4	NV_FSI_GIC_CAN1_INTR0	8	none
FSI - CAN1	fsi_can1_intr1	NV_FSI_AVIC1_CAN1_INTR1	1 - 5	NV_FSI_GIC_CAN1_INTR1	9	none

Source	Interrupt Name	FSI AVIC Software Name	AVIC Connection	FSI GIC Software Name	FSI GIC Connection	LIC Connection
FSI - SPI0	fsi_spi0_intr	NV_FSI_AVIC1_SPI0_INTR	1 - 6	NV_FSI_GIC_SPI0_INTR	29	none
FSI - UART0	fsi_uart0_intr	NV_FSI_AVIC1_UART0_INTR	1 - 7	NV_FSI_GIC_UART0_INTR	30	none
HSP2	top2fsi_hsp2_shrd_intr[0]	NV_FSI_AVIC1_TOP_HSP2_SI0	1 - 16	NV_FSI_GIC_TOP_HSP2_SI0	21	none
HSP2	top2fsi_hsp2_shrd_intr[1]	NV_FSI_AVIC1_TOP_HSP2_SI1	1 - 17	NV_FSI_GIC_TOP_HSP2_SI1	22	none
HSP2	top2fsi_hsp2_shrd_intr[2]	NV_FSI_AVIC1_TOP_HSP2_SI2	1 - 18	NV_FSI_GIC_TOP_HSP2_SI2	44	none
HSP2	top2fsi_hsp2_shrd_intr[3]	NV_FSI_AVIC1_TOP_HSP2_SI3	1 - 19	NV_FSI_GIC_TOP_HSP2_SI3	45	none
HSP2	top2fsi_hsp2_shrd_intr[4]	NV_FSI_AVIC1_TOP_HSP2_SI4	1 - 20	NV_FSI_GIC_TOP_HSP2_SI4	46	none
HSP2	top2fsi_hsp2_shrd_intr[5]	NV_FSI_AVIC1_TOP_HSP2_SI5	1 - 21	NV_FSI_GIC_TOP_HSP2_SI5	47	none
HSP2	top2fsi_hsp2_shrd_intr[6]	NV_FSI_AVIC1_TOP_HSP2_SI6	1 - 22	NV_FSI_GIC_TOP_HSP2_SI6	23	none
HSP2	top2fsi_hsp2_shrd_intr[7]	NV_FSI_AVIC1_TOP_HSP2_SI7	1 - 23	NV_FSI_GIC_TOP_HSP2_SI7	24	none
LIC	lic2fsi_intr1[1]	NV_FSI_AVIC1_LIC_IRQ1	1 - 24	none	none	none
LIC	lic2fsi_intr1[0]	NV_FSI_AVIC1_LIC_FIQ1	1 - 25	none	none	none
FSI - R5		NV_FSI_AVIC1_PMU_INTR	1 - 26	none	none	none
FSI - R5 FPU		NV_FSI_AVIC1_FPU_INTR	1 - 27	none	none	none
FSI - CTI		NV_FSI_AVIC1_CTI_INTR	1 - 28	none	none	none
FSI - HSM	fsi_hsm_lp_intr	none	none	NV_FSI_GIC_HSM_LP_INTR	10	none
FSI - HSM	fsi_hsm_hp_intr	none	none	NV_FSI_GIC_HSM_HP_INTR	11	none
FSI - HSM	fsi_hsm_critical_intr	none	none	NV_FSI_GIC_HSM_CRITICAL_ERR	12	none
SCE - HSP1	sce2fsi_hsp1_shrd_intr[0]	none	none	NV_FSI_GIC_SCE_HSP1_SI0	13	none
SCE - HSP1	sce2fsi_hsp1_shrd_intr[1]	none	none	NV_FSI_GIC_SCE_HSP1_SI1	14	none
SCE - HSP1	sce2fsi_hsp1_shrd_intr[2]	none	none	NV_FSI_GIC_SCE_HSP1_SI2	15	none

Source	Interrupt Name	FSI AVIC Software Name	AVIC Connection	FSI GIC Software Name	FSI GIC Connection	LIC Connection
SCE - HSP1	sce2fsi_hsp1_shrd_intr[3]	none	none	NV_FSI_GIC_SCE_HSP1_SI3	16	none
SCE - HSP1	sce2fsi_hsp1_shrd_intr[4]	none	none	NV_FSI_GIC_SCE_HSP1_SI4	36	none
SCE - HSP1	sce2fsi_hsp1_shrd_intr[5]	none	none	NV_FSI_GIC_SCE_HSP1_SI5	37	none
SCE - HSP1	sce2fsi_hsp1_shrd_intr[6]	none	none	NV_FSI_GIC_SCE_HSP1_SI6	38	none
SCE - HSP1	sce2fsi_hsp1_shrd_intr[7]	none	none	NV_FSI_GIC_SCE_HSP1_SI7	39	none
SCE - HSP2	sce2fsi_hsp2_shrd_intr[0]	none	none	NV_FSI_GIC_SCE_HSP2_SI0	40	none
SCE - HSP2	sce2fsi_hsp2_shrd_intr[1]	none	none	NV_FSI_GIC_SCE_HSP2_SI1	41	none
SCE - HSP2	sce2fsi_hsp2_shrd_intr[2]	none	none	NV_FSI_GIC_SCE_HSP2_SI2	42	none
SCE - HSP2	sce2fsi_hsp2_shrd_intr[3]	none	none	NV_FSI_GIC_SCE_HSP2_SI3	43	none
SCE - HSP2	sce2fsi_hsp2_shrd_intr[4]	none	none	NV_FSI_GIC_SCE_HSP2_SI4	17	none
SCE - HSP2	sce2fsi_hsp2_shrd_intr[5]	none	none	NV_FSI_GIC_SCE_HSP2_SI5	18	none
SCE - HSP2	sce2fsi_hsp2_shrd_intr[6]	none	none	NV_FSI_GIC_SCE_HSP2_SI6	19	none
SCE - HSP2	sce2fsi_hsp2_shrd_intr[7]	none	none	NV_FSI_GIC_SCE_HSP2_SI7	20	none
LIC	lic2fsi_intr0[0]	none	none	NV_FSI_GIC_LIC_FIQ0	25	none
LIC	lic2fsi_intr0[1]	none	none	NV_FSI_GIC_LIC_IRQ0	48	none
SPI3	spi3_intr	none	none	NV_FSI_GIC_SPI3_INTR	54	38

8.2.3 Interrupt Controller Registers

Note: The Interrupt controller register descriptions do not follow the standard register table protocol described in "Reading Register Tables" in the Introduction chapter. Each register's relative offset, access type, and power-on reset value are listed in the table below.

There are seventeen slices numbered 0 to 16, each with twelve channels numbered from 0 to 11. In particular, the slice <s> is associated with the Interrupts numbered between <s>*32 and <s>*32 + 31. And each slice has:

- Twelve sets of 10 registers with names using the form <*CHANNEL*>_<*SLICE*>_<*FUNCTION*>_0, where
  - <*CHANNEL*> (= 0, 1, ..., 11) specifies which of the twelve (0, 1, ..., 11) channels.
  - <*SLICE*> (= 0, 1, ..., 16) specifies which of the seventeen (0, 1, ..., 16) slices.
  - <*FUNCTION*> is the name of the register. In particular,
    - VIRQ and VFIQ: indicate the Valid status, as an IRQ or FIQ, respectively, for this channel.
    - IER (Interrupt Enable Register): indicates that the corresponding Interrupt is enabled for the channel.
    - IER_SET and IER_CLR: to control IER, bits equal to 1 are set or cleared, respectively, in the IER register. The use of SET and CLR registers avoids the need for a Read-Modify-Write operation.
    - IEP_CLASS: Interrupt Enable Priority Class. A bit set to 0 indicates IRQ, and a bit set to 1 indicates FIQ.
    - ISR: indicates the current value of the inputs to the channel after IDR.
    - IDR (Interrupt Disable Register): indicates that the corresponding Interrupt is disabled for the channel.
    - IDR_SET and IDR_CLR: to control IDR, bits equal to 1 are set or cleared, respectively, in the IDR register. The use of SET and CLR registers avoids the need for a Read-Modify-Write operation.
- Seven additional registers with names COMMON_<*SLICE*>_<*FUNCTION*>_0, where
  - <*SLICE*> specifies which of the seventeen (0, 1, ..., 16) slices.
  - <*FUNCTION*> is the name of the register. In particular,
    - GISR (Global Interrupt Status Register): indicates the status of the shared Interrupt inputs as they enter the LIC.
    - FIR: indicates Force Interrupt
    - FIR_SET and FIR_CLR: to control FIR, bits set to 1 are set or cleared, respectively, in the FIR. The use of SET and CLR registers avoids the need for a Read-Modify-Write operation.
    - CIDR (CCPLEX Interrupt Disable Register): indicates which shared Interrupts do not propagate towards the CCPLEX.
    - CIDR_SET and CIDR_CLR: to control CIDR, bits set to 1 are set or cleared, respectively, in the CIDR. The use of SET and CLR registers avoids the need for a Read-Modify-Write operation.

8.2.3.1 CHANNEL Registers Summary

The table below summarizes each <*CHANNEL*>/<*SLICE*> register set in its base offset and Interrupt mapping reference.

Table 8.12 <*CHANNEL*>/<*SLICE*> Mapping

CHANNEL	*SLICE*	Base Offset	Mapping (Interrupt Number)
0	0	0x0000	0 to 31
	1	0x0040	32 to 63 (60, 61, 63 reserved)
	2	0x0080	64 to 95 (67 reserved)
	3	0x00c0	96 to 127 (108, 109, 110, 111 reserved)
	4	0x0100	128 to 159
	5	0x0140	160 to 191 (179, 180, 181 reserved)
	6	0x0180	192 to 223
	7	0x01c0	224 to 255 (225 reserved)
	8	0x0200	256 to 287 (256, 257, 258, 287 reserved)
	9	0x0240	288 to 319
	10	0x0280	320 to 351
	11	0x02c0	352 to 383 (374, 375, 382, 383 reserved)
	12	0x0300	384 to 415
	13	0x0340	416 to 447 (420, 421, 422, 427 reserved)
	14	0x0380	448 to 479
	15	0x03c0	480 to 511
	16	0x0400	512 to 543
1	0	0x0800	0 to 31
	1	0x0840	32 to 63 (60, 61, 63 reserved)
	2	0x0880	64 to 95 (67 reserved)
	3	0x08c0	96 to 127 (108, 109, 110, 111 reserved)
	4	0x0900	128 to 159
	5	0x0940	160 to 191 (179, 180, 181 reserved)
	6	0x0980	192 to 223
	7	0x09c0	224 to 255 (225 reserved)
	8	0x0a00	256 to 287 (256, 257, 258, 287 reserved)
	9	0x0a40	288 to 319
	10	0x0a80	320 to 351

CHANNEL	*SLICE*	Base Offset	Mapping (Interrupt Number)
	11	0xac0	352 to 383 (374, 375, 382, 383 reserved)
	12	0x0b00	384 to 415
	13	0x0b40	416 to 447 (420, 421, 422, 427 reserved)
	14	0x0b80	448 to 479
	15	0x0c00	480 to 511
	16	0x0c40	512 to 543
2	0	0x1000	0 to 31
	1	0x1040	32 to 63 (60, 61, 63 reserved)
	2	0x1080	64 to 95 (67 reserved)
	3	0x10c0	96 to 127 (108, 109, 110, 111 reserved)
	4	0x1100	128 to 159
	5	0x1140	160 to 191 (179, 180, 181 reserved)
	6	0x1180	192 to 223
	7	0x11c0	224 to 255 (225 reserved)
	8	0x1200	256 to 287 (256, 257, 258, 287 reserved)
	9	0x1240	288 to 319
	10	0x1280	320 to 351
	11	0x12c0	352 to 383 (374, 375, 382, 383 reserved)
	12	0x1300	384 to 415
	13	0x1340	416 to 447 (420, 421, 422, 427 reserved)
	14	0x1380	448 to 479
	15	0x13c0	480 to 511
	16	0x1400	512 to 543
3	0	0x1800	0 to 31
	1	0x1840	32 to 63 (60, 61, 63 reserved)
	2	0x1880	64 to 95 (67 reserved)
	3	0x18c0	96 to 127 (108, 109, 110, 111 reserved)
	4	0x1900	128 to 159
	5	0x1940	160 to 191 (179, 180, 181 reserved)

CHANNEL	*SLICE*	Base Offset	Mapping (Interrupt Number)
	6	0x1980	192 to 223
	7	0x19c0	224 to 255 (225 reserved)
	8	0x1a00	256 to 287 (256, 257, 258, 287 reserved)
	9	0x1a40	288 to 319
	10	0x1a80	320 to 351
	11	0x1ac0	352 to 383 (374, 375, 382, 383 reserved)
	12	0x1b00	384 to 415
	13	0x1b40	416 to 447 (420, 421, 422, 427 reserved)
	14	0x1b80	448 to 479
	15	0x1bc0	480 to 511
	16	0x1c00	512 to 543
4	0	0x2000	0 to 31
	1	0x2040	32 to 63 (60, 61, 63 reserved)
	2	0x2080	64 to 95 (67 reserved)
	3	0x20c0	96 to 127 (108, 109, 110, 111 reserved)
	4	0x2100	128 to 159
	5	0x2140	160 to 191 (179, 180, 181 reserved)
	6	0x2180	192 to 223
	7	0x21c0	224 to 255 (225 reserved)
	8	0x2200	256 to 287 (256, 257, 258, 287 reserved)
	9	0x2240	288 to 319
	10	0x2280	320 to 351
	11	0x22c0	352 to 383 (374, 375, 382, 383 reserved)
	12	0x2300	384 to 415
	13	0x2340	416 to 447 (420, 421, 422, 427 reserved)
	14	0x2380	448 to 479
	15	0x23c0	480 to 511
	16	0x2400	512 to 543

CHANNEL	*SLICE*	Base Offset	Mapping (Interrupt Number)
5	0	0x2800	0 to 31
	1	0x2840	32 to 63 (60, 61, 63 reserved)
	2	0x2880	64 to 95 (67 reserved)
	3	0x28c0	96 to 127 (108, 109, 110, 111 reserved)
	4	0x2900	128 to 159
	5	0x2940	160 to 191 (179, 180, 181 reserved)
	6	0x2980	192 to 223
	7	0x29c0	224 to 255 (225 reserved)
	8	0x2a00	256 to 287 (256, 257, 258, 287 reserved)
	9	0x2a40	288 to 319
	10	0x2a80	320 to 351
	11	0x2ac0	352 to 383 (374, 375, 382, 383 reserved)
	12	0x2b00	384 to 415
	13	0x2b40	416 to 447 (420, 421, 422, 427 reserved)
	14	0x2b80	448 to 479
	15	0x2bc0	480 to 511
	16	0x2c00	512 to 543
6	0	0x3000	0 to 31
	1	0x3040	32 to 63 (60, 61, 63 reserved)
	2	0x3080	64 to 95 (67 reserved)
	3	0x30c0	96 to 127 (108, 109, 110, 111 reserved)
	4	0x3100	128 to 159
	5	0x3140	160 to 191 (179, 180, 181 reserved)
	6	0x3180	192 to 223
	7	0x31c0	224 to 255 (225 reserved)
	8	0x3200	256 to 287 (256, 257, 258, 287 reserved)
	9	0x3240	288 to 319
	10	0x3280	320 to 351
	11	0x32c0	352 to 383 (374, 375, 382, 383 reserved)

CHANNEL	*SLICE*	Base Offset	Mapping (Interrupt Number)
	12	0x3300	384 to 415
	13	0x3340	416 to 447 (420, 421, 422, 427 reserved)
	14	0x3380	448 to 479
	15	0x33c0	480 to 511
	16	0x3400	512 to 543
7	0	0x3800	0 to 31
	1	0x3840	32 to 63 (60, 61, 63 reserved)
	2	0x3880	64 to 95 (67 reserved)
	3	0x38c0	96 to 127 (108, 109, 110, 111 reserved)
	4	0x3900	128 to 159
	5	0x3940	160 to 191 (179, 180, 181 reserved)
	6	0x3980	192 to 223
	7	0x39c0	224 to 255 (225 reserved)
	8	0x3a00	256 to 287 (256, 257, 258, 287 reserved)
	9	0x3a40	288 to 319
	10	0x3a80	320 to 351
	11	0x3ac0	352 to 383 (374, 375, 382, 383 reserved)
	12	0x3b00	384 to 415
	13	0x3b40	416 to 447 (420, 421, 422, 427 reserved)
	14	0x3b80	448 to 479
	15	0x3bc0	480 to 511
	16	0x3c00	512 to 543
8	0	0x4000	0 to 31
	1	0x4040	32 to 63 (60, 61, 63 reserved)
	2	0x4080	64 to 95 (67 reserved)
	3	0x40c0	96 to 127 (108, 109, 110, 111 reserved)
	4	0x4100	128 to 159
	5	0x4140	160 to 191 (179, 180, 181 reserved)
	6	0x4180	192 to 223

CHANNEL	*SLICE*	Base Offset	Mapping (Interrupt Number)
	7	0x41c0	224 to 255 (225 reserved)
	8	0x4200	256 to 287 (256, 257, 258, 287 reserved)
	9	0x4240	288 to 319
	10	0x4280	320 to 351
	11	0x42c0	352 to 383 (374, 375, 382, 383 reserved)
	12	0x4300	384 to 415
	13	0x4340	416 to 447 (420, 421, 422, 427 reserved)
	14	0x4380	448 to 479
	15	0x43c0	480 to 511
	16	0x4400	512 to 543
9	0	0x4800	0 to 31
	1	0x4840	32 to 63 (60, 61, 63 reserved)
	2	0x4880	64 to 95 (67 reserved)
	3	0x48c0	96 to 127 (108, 109, 110, 111 reserved)
	4	0x4900	128 to 159
	5	0x4940	160 to 191 (179, 180, 181 reserved)
	6	0x4980	192 to 223
	7	0x49c0	224 to 255 (225 reserved)
	8	0x4a00	256 to 287 (256, 257, 258, 287 reserved)
	9	0x4a40	288 to 319
	10	0x4a80	320 to 351
	11	0x4ac0	352 to 383 (374, 375, 382, 383 reserved)
	12	0x4b00	384 to 415
	13	0x4b40	416 to 447 (420, 421, 422, 427 reserved)
	14	0x4b80	448 to 479
	15	0x4bc0	480 to 511
	16	0x4c00	512 to 543

CHANNEL	*SLICE*	Base Offset	Mapping (Interrupt Number)
10	0	0x5000	0 to 31
	1	0x5040	32 to 63 (60, 61, 63 reserved)
	2	0x5080	64 to 95 (67 reserved)
	3	0x50c0	96 to 127 (108, 109, 110, 111 reserved)
	4	0x5100	128 to 159
	5	0x5140	160 to 191 (179, 180, 181 reserved)
	6	0x5180	192 to 223
	7	0x51c0	224 to 255 (225 reserved)
	8	0x5200	256 to 287 (256, 257, 258, 287 reserved)
	9	0x5240	288 to 319
	10	0x5280	320 to 351
	11	0x52c0	352 to 383 (374, 375, 382, 383 reserved)
	12	0x5300	384 to 415
	13	0x5340	416 to 447 (420, 421, 422, 427 reserved)
	14	0x5380	448 to 479
	15	0x53c0	480 to 511
	16	0x5400	512 to 543
11	0	0x5800	0 to 31
	1	0x5840	32 to 63 (60, 61, 63 reserved)
	2	0x5880	64 to 95 (67 reserved)
	3	0x58c0	96 to 127 (108, 109, 110, 111 reserved)
	4	0x5900	128 to 159
	5	0x5940	160 to 191 (179, 180, 181 reserved)
	6	0x5980	192 to 223
	7	0x59c0	224 to 255 (225 reserved)
	8	0x5a00	256 to 287 (256, 257, 258, 287 reserved)
	9	0x5a40	288 to 319
	10	0x5a80	320 to 351
	11	0x5ac0	352 to 383 (374, 375, 382, 383 reserved)

CHANNEL	*SLICE*	Base Offset	Mapping (Interrupt Number)
	12	0x5b00	384 to 415
	13	0x5b40	416 to 447 (420, 421, 422, 427 reserved)
	14	0x5b80	448 to 479
	15	0x5bc0	480 to 511
	16	0x5c00	512 to 543

The table below lists the 10 registers common to each <*SLICE*>/<*CHANNEL*> combination (i.e., the <*SLICE*>/<*CHANNEL*> register set).

Note: The actual offset for each register is the Base Offset (from the table above) plus the Relative Offset in the table below.

Table 8.13 Register Set per < *CHANNEL* >/< *SLICE* >

Name	Relative Offset	RD/WR	Reset	SCR Protection	Description
INTR_CTLR_<CHANNEL>_<SLICE>_VIRQ_0	0x00	RO	x	<CHANNEL>_SCR E_0	Valid Interrupt Request Status Register
INTR_CTLR_<CHANNEL>_<SLICE>_VFIQ_0	0x04	RO	x		FIQ Valid Interrupt Request Status Register
INTR_CTLR_<CHANNEL>_<SLICE>_IER_0	0x08	RW	0		Interrupt Enable Register
INTR_CTLR_<CHANNEL>_<SLICE>_IER_SET_0	0x0c	RW	x		Set Interrupt Enable Register
INTR_CTLR_<CHANNEL>_<SLICE>_IER_CLR_0	0x10	RW	x		Clear Interrupt Enable Register
INTR_CTLR_<CHANNEL>_<SLICE>_IEP_CLASS_0	0x14	RW	0		Interrupt Enable Priority Class Register
INTR_CTLR_<CHANNEL>_<SLICE>_ISR_0	0x18	RO	0		Interrupt Status Register
INTR_CTLR_<CHANNEL>_<SLICE>_IDR_0	0x1c	RW	0	<CHANNEL>_SCR D_0	Interrupt Disable Register
INTR_CTLR_<CHANNEL>_<SLICE>_IDR_SET_0	0x20	RW	x		Set Interrupt Disable Register
INTR_CTLR_<CHANNEL>_<SLICE>_IDR_CLR_0	0x24	RW	x		Clear Interrupt Disable Register

8.2.3.2 COMMON Registers Summary

The table below summarizes each <SLICE> register set in its base offset and Interrupt mapping reference.

Table 8.14 COMMON < *SLICE* > Mapping

SLICE	Base Offset	Mapping (Interrupt Number)
0	0xf800	0 to 31
1	0xf840	32 to 63 (60, 61, 63 reserved)
2	0xf880	64 to 95 (67 reserved)
3	0xf8c0	96 to 127 (108, 109, 110, 111 reserved)
4	0xf900	128 to 159
5	0xf940	160 to 191 (179, 180, 181 reserved)
6	0xf980	192 to 223
7	0xf9c0	224 to 255 (225 reserved)
8	0xfa00	256 to 287 (256, 257, 258, 287 reserved)
9	0xfa40	288 to 319
10	0xfa80	320 to 351
11	0xfac0	352 to 383 (374, 375, 382, 383 reserved)
12	0xfb00	384 to 415
13	0xfb40	416 to 447 (420, 421, 422, 427 reserved)
14	0xfb80	448 to 479
15	0xfbc0	480 to 511
16	0xfc00	512 to 543

The table below lists the seven common registers for each <*SLICE*>.

Note: The actual offset for each register is the Base Offset (from the table above) plus the Relative Offset in the table below.

Table 8.15 COMMON Registers per < *SLICE* >

Name	Relative Offset	RD/ WR	Reset	SCR Protection	Description
INTR_CTLR_COMMON_<SLICE>_GISR_0	0x00	RO	x	SCR_COMMON_0	Global Interrupt Request Status Register
INTR_CTLR_COMMON_<SLICE>_FIR_0	0x04	RW	0		Force Interrupt Register
INTR_CTLR_COMMON_<SLICE>_FIR_SET_0	0x08	RW	x		Set Force Interrupt Register

Name	Relative Offset	RD/ WR	Reset	SCR Protection	Description
INTR_CTLR_COMMON_<SLICE>_FIR_CLR_0	0x0c	RW	x		Clear Force Interrupt Register
INTR_CTLR_COMMON_<SLICE>_CIDR_0	0x10	RW	0		CCPLEX Interrupt Disable Register
INTR_CTLR_COMMON_<SLICE>_CIDR_SET_0	0x14	RW	x		Set CCPLEX Interrupt Disable Register
INTR_CTLR_COMMON_<SLICE>_CIDR_CLR_0	0x18	RW	x		Clear CCPLEX Interrupt Disable Register

8.2.3.3 Interrupt Controller Common Registers

INTR_CTLR_COMMON_SLICE<*j*>_GISR_0,

where <*j*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16.

This Global Interrupt Status Register (GISR) specifies the status of the shared Interrupt inputs as they enter the LIC for the speific Slice.

INTR_CTLR_COMMON_SLICE0_GISR_0
Offset: 0xf800

INTR_CTLR_COMMON_SLICE1_GISR_0
Offset: 0xf840

INTR_CTLR_COMMON_SLICE2_GISR_0
Offset: 0xf880

INTR_CTLR_COMMON_SLICE3_GISR_0
Offset: 0xf8c0

INTR_CTLR_COMMON_SLICE4_GISR_0
Offset: 0xf900

INTR_CTLR_COMMON_SLICE5_GISR_0
Offset: 0xf940

INTR_CTLR_COMMON_SLICE6_GISR_0
Offset: 0xf980

NTR_CTLR_COMMON_SLICE7_GISR_0
Offset: 0xf9c0

INTR_CTLR_COMMON_SLICE8_GISR_0
Offset: 0xfa00

INTR_CTLR_COMMON_SLICE9_GISR_0
Offset: 0xfa40

INTR_CTLR_COMMON_SLICE10_GISR_0
Offset: 0xfa80

INTR_CTLR_COMMON_SLICE11_GISR_0
Offset: 0xfac0

INTR_CTLR_COMMON_SLICE12_GISR_0
Offset: 0xfb00

INTR_CTLR_COMMON_SLICE13_GISR_0
Offset: 0xfb40

INTR_CTLR_COMMON_SLICE14_GISR_0
Offset: 0xfb80

INTR_CTLR_COMMON_SLICE15_GISR_0
Offset: 0xfbc0

INTR_CTLR_COMMON_SLICE16_GISR_0
Offset: 0xfc00

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: INTR_CTLR_COMMON_SCR_SCR_COMMON_0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**GISR**

INTR_CTLR_COMMON_SLICE<*j*>_FIR_0,

where <*j*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16.

Each bit of this Force Interrupt Register (FIR) indiates whether the coresponding Interrupt is Force Interrupt for the speific Slice.

INTR_CTLR_COMMON_SLICE0_FIR_0
Offset: 0xf804

INTR_CTLR_COMMON_SLICE1_FIR_0
Offset: 0xf844

INTR_CTLR_COMMON_SLICE2_FIR_0
Offset: 0xf884

INTR_CTLR_COMMON_SLICE3_FIR_0
Offset: 0xf8c4

INTR_CTLR_COMMON_SLICE4_FIR_0
Offset: 0xf904

INTR_CTLR_COMMON_SLICE5_FIR_0
Offset: 0xf944

INTR_CTLR_COMMON_SLICE6_FIR_0
Offset: 0xf984

INTR_CTLR_COMMON_SLICE7_FIR_0
Offset: 0xf9c4

INTR_CTLR_COMMON_SLICE8_FIR_0
Offset: 0xfa04

INTR_CTLR_COMMON_SLICE9_FIR_0
Offset: 0xfa44

INTR_CTLR_COMMON_SLICE10_FIR_0
Offset: 0xfa84

INTR_CTLR_COMMON_SLICE11_FIR_0
Offset: 0xfac4

INTR_CTLR_COMMON_SLICE12_FIR_0
Offset: 0xfb04

INTR_CTLR_COMMON_SLICE13_FIR_0
Offset: 0xfb44

INTR_CTLR_COMMON_SLICE14_FIR_0
Offset: 0xfb84

INTR_CTLR_COMMON_SLICE15_FIR_0
Offset: 0xfbc4

INTR_CTLR_COMMON_SLICE16_FIR_0
Offset: 0xfc04

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: INTR_CTLR_COMMON_SCR_SCR_COMMON_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIR**

INTR_CTLR_COMMON_SLICE<*j*>_FIR_SET_0,

where <*j*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16.

Each bit of this register is used to set the corresponding bit in the Force Interrupt Register (FIR) for the specific Slice.

INTR_CTLR_COMMON_SLICE0_FIR_SET_0
Offset: 0xf808

INTR_CTLR_COMMON_SLICE1_FIR_SET_0
Offset: 0xf848

INTR_CTLR_COMMON_SLICE2_FIR_SET_0
Offset: 0xf888

INTR_CTLR_COMMON_SLICE3_FIR_SET_0
Offset: 0xf8c8

INTR_CTLR_COMMON_SLICE4_FIR_SET_0
Offset: 0xf908

INTR_CTLR_COMMON_SLICE5_FIR_SET_0
Offset: 0xf948

INTR_CTLR_COMMON_SLICE6_FIR_SET_0
Offset: 0xf988

INTR_CTLR_COMMON_SLICE7_FIR_SET_0
Offset: 0xf9c8

INTR_CTLR_COMMON_SLICE8_FIR_SET_0
Offset: 0xfa08

INTR_CTLR_COMMON_SLICE9_FIR_SET_0
Offset: 0xfa48

INTR_CTLR_COMMON_SLICE10_FIR_SET_0
Offset: 0xfa88

INTR_CTLR_COMMON_SLICE11_FIR_SET_0
Offset: 0xfac8

INTR_CTLR_COMMON_SLICE12_FIR_SET_0
Offset: 0xfb08

INTR_CTLR_COMMON_SLICE13_FIR_SET_0
Offset: 0xfb48

INTR_CTLR_COMMON_SLICE14_FIR_SET_0
Offset: 0xfb88

INTR_CTLR_COMMON_SLICE15_FIR_SET_0
Offset: 0xfbc8

INTR_CTLR_COMMON_SLICE16_FIR_SET_0
Offset: 0xfc08

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: INTR_CTLR_COMMON_SCR_SCR_COMMON_0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**FIR_SET** Each bit for the corresponding bit in FIR. 0: No action. 1: Set the bit.

INTR_CTLR_COMMON_SLICE<*j*>_FIR_CLR_0,

where <*j*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16.

Each bit of this register is used to clear the corresponding bit in the Force Interrupt Register (FIR) for the specific Slice.

INTR_CTLR_COMMON_SLICE0_FIR_CLR_0
Offset: 0xf80c

INTR_CTLR_COMMON_SLICE1_FIR_CLR_0
Offset: 0xf84c

INTR_CTLR_COMMON_SLICE2_FIR_CLR_0
Offset: 0xf88c

INTR_CTLR_COMMON_SLICE3_FIR_CLR_0
Offset: 0xf8cc

INTR_CTLR_COMMON_SLICE4_FIR_CLR_0
Offset: 0xf90c

INTR_CTLR_COMMON_SLICE5_FIR_CLR_0
Offset: 0xf94c

INTR_CTLR_COMMON_SLICE6_FIR_CLR_0
Offset: 0xf98c

INTR_CTLR_COMMON_SLICE7_FIR_CLR_0
Offset: 0xf9cc

INTR_CTLR_COMMON_SLICE8_FIR_CLR_0
Offset: 0xfa0c

INTR_CTLR_COMMON_SLICE9_FIR_CLR_0
Offset: 0xfa4c

INTR_CTLR_COMMON_SLICE10_FIR_CLR_0
Offset: 0xfa8c

INTR_CTLR_COMMON_SLICE11_FIR_CLR_0
Offset: 0xfacc

INTR_CTLR_COMMON_SLICE12_FIR_CLR_0
Offset: 0xfb0c

INTR_CTLR_COMMON_SLICE13_FIR_CLR_0
Offset: 0xfb4c

INTR_CTLR_COMMON_SLICE14_FIR_CLR_0
Offset: 0xfb8c

INTR_CTLR_COMMON_SLICE15_FIR_CLR_0
Offset: 0xfbcc

INTR_CTLR_COMMON_SLICE16_FIR_CLR_0
Offset: 0xfc0c

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: INTR_CTLR_COMMON_SCR_SCR_COMMON_0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**FIR_CLR** Each bit for the corresponding bit in FIR. 0: No action. 1: Clear the bit.

INTR_CTLR_COMMON_SLICE<*j*>_CIDR_0,

where <*j*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16.

Each bit in this CCPLEX Interrupt Disable Register (CIDR) for the specific Slice indicates whether the corresponding shared Interrupt popagates to the CCPLEX.

INTR_CTLR_COMMON_SLICE0_CIDR_0
Offset: 0xf810

INTR_CTLR_COMMON_SLICE1_CIDR_0
Offset: 0xf850

INTR_CTLR_COMMON_SLICE2_CIDR_0
Offset: 0xf890

INTR_CTLR_COMMON_SLICE3_CIDR_0
Offset: 0xf8d0

INTR_CTLR_COMMON_SLICE4_CIDR_0
Offset: 0xf910

INTR_CTLR_COMMON_SLICE5_CIDR_0
Offset: 0xf950

INTR_CTLR_COMMON_SLICE6_CIDR_0
Offset: 0xf990

INTR_CTLR_COMMON_SLICE7_CIDR_0
Offset: 0xf9d0

INTR_CTLR_COMMON_SLICE8_CIDR_0
Offset: 0xfa10

INTR_CTLR_COMMON_SLICE9_CIDR_0
Offset: 0xfa50

INTR_CTLR_COMMON_SLICE10_CIDR_0
Offset: 0xfa90

INTR_CTLR_COMMON_SLICE11_CIDR_0
Offset: 0xfad0

INTR_CTLR_COMMON_SLICE12_CIDR_0
Offset: 0xfb10

INTR_CTLR_COMMON_SLICE13_CIDR_0
Offset: 0xfb50

INTR_CTLR_COMMON_SLICE14_CIDR_0
Offset: 0xfb90

INTR_CTLR_COMMON_SLICE15_CIDR_0
Offset: 0xfbd0

INTR_CTLR_COMMON_SLICE16_CIDR_0
Offset: 0xfc10

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: INTR_CTLR_COMMON_SCR_SCR_COMMON_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CIDR** Each bit for the corresponding shared Interrupt. 0: Not propagated to CCPLEX. 1: Propagated to CCPLEX.

INTR_CTLR_COMMON_SLICE<*j*>_CIDR_SET_0,

where <*j*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16.

Each bit of this register is used to set the corresponding bit in the CCPLEX Interrupt Disable Register (CIDR) for the specific Slice.

INTR_CTLR_COMMON_SLICE0_CIDR_SET_0
Offset: 0xf814

INTR_CTLR_COMMON_SLICE1_CIDR_SET_0
Offset: 0xf854

INTR_CTLR_COMMON_SLICE2_CIDR_SET_0
Offset: 0xf894

INTR_CTLR_COMMON_SLICE3_CIDR_SET_0
Offset: 0xf8d4

INTR_CTLR_COMMON_SLICE4_CIDR_SET_0
Offset: 0xf914

INTR_CTLR_COMMON_SLICE5_CIDR_SET_0
Offset: 0xf954

INTR_CTLR_COMMON_SLICE6_CIDR_SET_0
Offset: 0xf994

INTR_CTLR_COMMON_SLICE7_CIDR_SET_0
Offset: 0xf9d4

INTR_CTLR_COMMON_SLICE8_CIDR_SET_0
Offset: 0xfa14

INTR_CTLR_COMMON_SLICE9_CIDR_SET_0
Offset: 0xfa54

INTR_CTLR_COMMON_SLICE10_CIDR_SET_0
Offset: 0xfa94

INTR_CTLR_COMMON_SLICE11_CIDR_SET_0
Offset: 0xfad4

INTR_CTLR_COMMON_SLICE12_CIDR_SET_0
Offset: 0xfb14

INTR_CTLR_COMMON_SLICE13_CIDR_SET_0
Offset: 0xfb54

INTR_CTLR_COMMON_SLICE14_CIDR_SET_0
Offset: 0xfb94

INTR_CTLR_COMMON_SLICE15_CIDR_SET_0
Offset: 0xfbd4

INTR_CTLR_COMMON_SLICE16_CIDR_SET_0
Offset: 0xfc14

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: INTR_CTLR_COMMON_SCR_SCR_COMMON_0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**CIDR_SET** Each bit for the corresponding bit in CIDR. 0: No action. 1: Set the bit.

INTR_CTLR_COMMON_SLICE<*j*>_CIDR_CLR_0,

where <*j*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16.

Each bit of this register is used to clear the corresponding bit in the CCPLEX Interrupt Disable Register (CIDR) for the specific Slice.

INTR_CTLR_COMMON_SLICE0_CIDR_CLR_0
Offset: 0xf818

INTR_CTLR_COMMON_SLICE1_CIDR_CLR_0
Offset: 0xf858

INTR_CTLR_COMMON_SLICE2_CIDR_CLR_0
Offset: 0xf898

INTR_CTLR_COMMON_SLICE3_CIDR_CLR_0
Offset: 0xf8d8

INTR_CTLR_COMMON_SLICE4_CIDR_CLR_0
Offset: 0xf918

INTR_CTLR_COMMON_SLICE5_CIDR_CLR_0
Offset: 0xf958

INTR_CTLR_COMMON_SLICE6_CIDR_CLR_0
Offset: 0xf998

INTR_CTLR_COMMON_SLICE7_CIDR_CLR_0
Offset: 0xf9d8

INTR_CTLR_COMMON_SLICE8_CIDR_CLR_0
Offset: 0xfa18

INTR_CTLR_COMMON_SLICE9_CIDR_CLR_0
Offset: 0xfa58

INTR_CTLR_COMMON_SLICE10_CIDR_CLR_0
Offset: 0xfa98

INTR_CTLR_COMMON_SLICE11_CIDR_CLR_0
Offset: 0xfad8

INTR_CTLR_COMMON_SLICE12_CIDR_CLR_0
Offset: 0xfb18

INTR_CTLR_COMMON_SLICE13_CIDR_CLR_0
Offset: 0xfb58

INTR_CTLR_COMMON_SLICE14_CIDR_CLR_0
Offset: 0xfb98

INTR_CTLR_COMMON_SLICE15_CIDR_CLR_0
Offset: 0xfbd8

INTR_CTLR_COMMON_SLICE16_CIDR_CLR_0
Offset: 0xfc18

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: INTR_CTLR_COMMON_SCR_SCR_COMMON_0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**CIDR_CLR** Each bit for the corresponding bit in CIDR. 0: No action. 1: Clear the bit.

8.2.3.4 Interrupt Controller Channel Registers

INTR_CTLR_CHANNEL<*i*>_SLICE<*j*>_VIRQ_0,

where <*i*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, and

<*j*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16.

This register indicates the valid IRQ status for the specific Channel/Slice.

INTR_CTLR_CHANNEL0_SLICE0_VIRQ_0

Offset: 0x0

INTR_CTLR_CHANNEL0_SLICE1_VIRQ_0
Offset: 0x40

INTR_CTLR_CHANNEL0_SLICE2_VIRQ_0
Offset: 0x80

INTR_CTLR_CHANNEL0_SLICE3_VIRQ_0
Offset: 0xc0

INTR_CTLR_CHANNEL0_SLICE4_VIRQ_0
Offset: 0x100

INTR_CTLR_CHANNEL0_SLICE5_VIRQ_0
Offset: 0x140

INTR_CTLR_CHANNEL0_SLICE6_VIRQ_0
Offset: 0x180

INTR_CTLR_CHANNEL0_SLICE7_VIRQ_0
Offset: 0x1c0

INTR_CTLR_CHANNEL0_SLICE8_VIRQ_0
Offset: 0x200

INTR_CTLR_CHANNEL0_SLICE9_VIRQ_0
Offset: 0x240

INTR_CTLR_CHANNEL0_SLICE10_VIRQ_0
Offset: 0x280

INTR_CTLR_CHANNEL0_SLICE11_VIRQ_0
Offset: 0x2c0

INTR_CTLR_CHANNEL0_SLICE12_VIRQ_0
Offset: 0x300

INTR_CTLR_CHANNEL0_SLICE13_VIRQ_0
Offset: 0x340

INTR_CTLR_CHANNEL0_SLICE14_VIRQ_0
Offset: 0x380

INTR_CTLR_CHANNEL0_SLICE15_VIRQ_0
Offset: 0x3c0

INTR_CTLR_CHANNEL0_SLICE16_VIRQ_0
Offset: 0x400

INTR_CTLR_CHANNEL1_SLICE0_VIRQ_0
Offset: 0x800

INTR_CTLR_CHANNEL1_SLICE1_VIRQ_0
Offset: 0x840

INTR_CTLR_CHANNEL1_SLICE2_VIRQ_0
Offset: 0x880

INTR_CTLR_CHANNEL1_SLICE3_VIRQ_0
Offset: 0x8c0

INTR_CTLR_CHANNEL1_SLICE4_VIRQ_0
Offset: 0x900

INTR_CTLR_CHANNEL1_SLICE5_VIRQ_0
Offset: 0x940

INTR_CTLR_CHANNEL1_SLICE6_VIRQ_0
Offset: 0x980

INTR_CTLR_CHANNEL1_SLICE7_VIRQ_0
Offset: 0x9c0

INTR_CTLR_CHANNEL1_SLICE8_VIRQ_0
Offset: 0xa00

INTR_CTLR_CHANNEL1_SLICE9_VIRQ_0
Offset: 0xa40

INTR_CTLR_CHANNEL1_SLICE10_VIRQ_0
Offset: 0xa80

INTR_CTLR_CHANNEL1_SLICE11_VIRQ_0
Offset: 0xac0

INTR_CTLR_CHANNEL1_SLICE12_VIRQ_0
Offset: 0xb00

INTR_CTLR_CHANNEL1_SLICE13_VIRQ_0
Offset: 0xb40

INTR_CTLR_CHANNEL1_SLICE14_VIRQ_0
Offset: 0xb80

INTR_CTLR_CHANNEL1_SLICE15_VIRQ_0
Offset: 0xbc0

INTR_CTLR_CHANNEL1_SLICE16_VIRQ_0
Offset: 0xc00

INTR_CTLR_CHANNEL2_SLICE0_VIRQ_0
Offset: 0x1000

INTR_CTLR_CHANNEL2_SLICE1_VIRQ_0
Offset: 0x1040

INTR_CTLR_CHANNEL2_SLICE2_VIRQ_0
Offset: 0x1080

INTR_CTLR_CHANNEL2_SLICE3_VIRQ_0
Offset: 0x10c0

INTR_CTLR_CHANNEL2_SLICE4_VIRQ_0
Offset: 0x1100

INTR_CTLR_CHANNEL2_SLICE5_VIRQ_0
Offset: 0x1140

INTR_CTLR_CHANNEL2_SLICE6_VIRQ_0
Offset: 0x1180

INTR_CTLR_CHANNEL2_SLICE7_VIRQ_0
Offset: 0x11c0

INTR_CTLR_CHANNEL2_SLICE8_VIRQ_0
Offset: 0x1200

INTR_CTLR_CHANNEL2_SLICE9_VIRQ_0
Offset: 0x1240

INTR_CTLR_CHANNEL2_SLICE10_VIRQ_0
Offset: 0x1280

INTR_CTLR_CHANNEL2_SLICE11_VIRQ_0
Offset: 0x12c0

INTR_CTLR_CHANNEL2_SLICE12_VIRQ_0
Offset: 0x1300

INTR_CTLR_CHANNEL2_SLICE13_VIRQ_0
Offset: 0x1340

INTR_CTLR_CHANNEL2_SLICE14_VIRQ_0
Offset: 0x1380

INTR_CTLR_CHANNEL2_SLICE15_VIRQ_0
Offset: 0x13c0

INTR_CTLR_CHANNEL2_SLICE16_VIRQ_0
Offset: 0x1400

INTR_CTLR_CHANNEL3_SLICE0_VIRQ_0
Offset: 0x1800

INTR_CTLR_CHANNEL3_SLICE1_VIRQ_0
Offset: 0x1840

INTR_CTLR_CHANNEL3_SLICE2_VIRQ_0
Offset: 0x1880

INTR_CTLR_CHANNEL3_SLICE3_VIRQ_0
Offset: 0x18c0

INTR_CTLR_CHANNEL3_SLICE4_VIRQ_0
Offset: 0x1900

INTR_CTLR_CHANNEL3_SLICE5_VIRQ_0
Offset: 0x1940

INTR_CTLR_CHANNEL3_SLICE6_VIRQ_0
Offset: 0x1980

INTR_CTLR_CHANNEL3_SLICE7_VIRQ_0
Offset: 0x19c0

INTR_CTLR_CHANNEL3_SLICE8_VIRQ_0
Offset: 0x1a00

INTR_CTLR_CHANNEL3_SLICE9_VIRQ_0
Offset: 0x1a40

INTR_CTLR_CHANNEL3_SLICE10_VIRQ_0
Offset: 0x1a80

INTR_CTLR_CHANNEL3_SLICE11_VIRQ_0
Offset: 0x1ac0

INTR_CTLR_CHANNEL3_SLICE12_VIRQ_0
Offset: 0x1b00

INTR_CTLR_CHANNEL3_SLICE13_VIRQ_0
Offset: 0x1b40

INTR_CTLR_CHANNEL3_SLICE14_VIRQ_0
Offset: 0x1b80

INTR_CTLR_CHANNEL3_SLICE15_VIRQ_0
Offset: 0x1bc0

INTR_CTLR_CHANNEL3_SLICE16_VIRQ_0
Offset: 0x1c00

INTR_CTLR_CHANNEL4_SLICE0_VIRQ_0
Offset: 0x2000

INTR_CTLR_CHANNEL4_SLICE1_VIRQ_0
Offset: 0x2040

INTR_CTLR_CHANNEL4_SLICE2_VIRQ_0
Offset: 0x2080

INTR_CTLR_CHANNEL4_SLICE3_VIRQ_0
Offset: 0x20c0

INTR_CTLR_CHANNEL4_SLICE4_VIRQ_0
Offset: 0x2100

INTR_CTLR_CHANNEL4_SLICE5_VIRQ_0
Offset: 0x2140

INTR_CTLR_CHANNEL4_SLICE6_VIRQ_0
Offset: 0x2180

INTR_CTLR_CHANNEL4_SLICE7_VIRQ_0
Offset: 0x21c0

INTR_CTLR_CHANNEL4_SLICE8_VIRQ_0
Offset: 0x2200

INTR_CTLR_CHANNEL4_SLICE9_VIRQ_0
Offset: 0x2240

INTR_CTLR_CHANNEL4_SLICE10_VIRQ_0
Offset: 0x2280

INTR_CTLR_CHANNEL4_SLICE11_VIRQ_0
Offset: 0x22c0

INTR_CTLR_CHANNEL4_SLICE12_VIRQ_0
Offset: 0x2300

INTR_CTLR_CHANNEL4_SLICE13_VIRQ_0
Offset: 0x2340

INTR_CTLR_CHANNEL4_SLICE14_VIRQ_0
Offset: 0x2380

INTR_CTLR_CHANNEL4_SLICE15_VIRQ_0
Offset: 0x23c0

INTR_CTLR_CHANNEL4_SLICE16_VIRQ_0
Offset: 0x2400

INTR_CTLR_CHANNEL5_SLICE0_VIRQ_0
Offset: 0x2800

INTR_CTLR_CHANNEL5_SLICE1_VIRQ_0
Offset: 0x2840

INTR_CTLR_CHANNEL5_SLICE2_VIRQ_0
Offset: 0x2880

INTR_CTLR_CHANNEL5_SLICE3_VIRQ_0
Offset: 0x28c0

INTR_CTLR_CHANNEL5_SLICE4_VIRQ_0
Offset: 0x2900

INTR_CTLR_CHANNEL5_SLICE5_VIRQ_0
Offset: 0x2940

INTR_CTLR_CHANNEL5_SLICE6_VIRQ_0
Offset: 0x2980

INTR_CTLR_CHANNEL5_SLICE7_VIRQ_0
Offset: 0x29c0

INTR_CTLR_CHANNEL5_SLICE8_VIRQ_0
Offset: 0x2a00

INTR_CTLR_CHANNEL5_SLICE9_VIRQ_0
Offset: 0x2a40

INTR_CTLR_CHANNEL5_SLICE10_VIRQ_0
Offset: 0x2a80

INTR_CTLR_CHANNEL5_SLICE11_VIRQ_0
Offset: 0x2ac0

INTR_CTLR_CHANNEL5_SLICE12_VIRQ_0
Offset: 0x2b00

INTR_CTLR_CHANNEL5_SLICE13_VIRQ_0
Offset: 0x2b40

INTR_CTLR_CHANNEL5_SLICE14_VIRQ_0
Offset: 0x2b80

INTR_CTLR_CHANNEL5_SLICE15_VIRQ_0
Offset: 0x2bc0

INTR_CTLR_CHANNEL5_SLICE16_VIRQ_0
Offset: 0x2c00

INTR_CTLR_CHANNEL6_SLICE0_VIRQ_0
Offset: 0x3000

INTR_CTLR_CHANNEL6_SLICE1_VIRQ_0
Offset: 0x3040

INTR_CTLR_CHANNEL6_SLICE2_VIRQ_0
Offset: 0x3080

INTR_CTLR_CHANNEL6_SLICE3_VIRQ_0
Offset: 0x30c0

INTR_CTLR_CHANNEL6_SLICE4_VIRQ_0
Offset: 0x3100

INTR_CTLR_CHANNEL6_SLICE5_VIRQ_0
Offset: 0x3140

INTR_CTLR_CHANNEL6_SLICE6_VIRQ_0
Offset: 0x3180

INTR_CTLR_CHANNEL6_SLICE7_VIRQ_0
Offset: 0x31c0

INTR_CTLR_CHANNEL6_SLICE8_VIRQ_0
Offset: 0x3200

INTR_CTLR_CHANNEL6_SLICE9_VIRQ_0
Offset: 0x3240

INTR_CTLR_CHANNEL6_SLICE10_VIRQ_0
Offset: 0x3280

INTR_CTLR_CHANNEL6_SLICE11_VIRQ_0
Offset: 0x32c0

INTR_CTLR_CHANNEL6_SLICE12_VIRQ_0
Offset: 0x3300

INTR_CTLR_CHANNEL6_SLICE13_VIRQ_0
Offset: 0x3340

INTR_CTLR_CHANNEL6_SLICE14_VIRQ_0
Offset: 0x3380

INTR_CTLR_CHANNEL6_SLICE15_VIRQ_0
Offset: 0x33c0

INTR_CTLR_CHANNEL6_SLICE16_VIRQ_0
Offset: 0x3400

INTR_CTLR_CHANNEL7_SLICE0_VIRQ_0
Offset: 0x3800

INTR_CTLR_CHANNEL7_SLICE1_VIRQ_0
Offset: 0x3840

INTR_CTLR_CHANNEL7_SLICE2_VIRQ_0
Offset: 0x3880

INTR_CTLR_CHANNEL7_SLICE3_VIRQ_0
Offset: 0x38c0

INTR_CTLR_CHANNEL7_SLICE4_VIRQ_0
Offset: 0x3900

INTR_CTLR_CHANNEL7_SLICE5_VIRQ_0
Offset: 0x3940

INTR_CTLR_CHANNEL7_SLICE6_VIRQ_0
Offset: 0x3980

INTR_CTLR_CHANNEL7_SLICE7_VIRQ_0
Offset: 0x39c0

INTR_CTLR_CHANNEL7_SLICE8_VIRQ_0
Offset: 0x3a00

INTR_CTLR_CHANNEL7_SLICE9_VIRQ_0
Offset: 0x3a40

INTR_CTLR_CHANNEL7_SLICE10_VIRQ_0
Offset: 0x3a80

INTR_CTLR_CHANNEL7_SLICE11_VIRQ_0
Offset: 0x3ac0

INTR_CTLR_CHANNEL7_SLICE12_VIRQ_0
Offset: 0x3b00

INTR_CTLR_CHANNEL7_SLICE13_VIRQ_0
Offset: 0x3b40

INTR_CTLR_CHANNEL7_SLICE14_VIRQ_0
Offset: 0x3b80

INTR_CTLR_CHANNEL7_SLICE15_VIRQ_0
Offset: 0x3bc0

INTR_CTLR_CHANNEL7_SLICE16_VIRQ_0
Offset: 0x3c00

INTR_CTLR_CHANNEL8_SLICE0_VIRQ_0
Offset: 0x4000

INTR_CTLR_CHANNEL8_SLICE1_VIRQ_0
Offset: 0x4040

INTR_CTLR_CHANNEL8_SLICE2_VIRQ_0
Offset: 0x4080

INTR_CTLR_CHANNEL8_SLICE3_VIRQ_0
Offset: 0x40c0

INTR_CTLR_CHANNEL8_SLICE4_VIRQ_0
Offset: 0x4100

INTR_CTLR_CHANNEL8_SLICE5_VIRQ_0
Offset: 0x4140

INTR_CTLR_CHANNEL8_SLICE6_VIRQ_0
Offset: 0x4180

INTR_CTLR_CHANNEL8_SLICE7_VIRQ_0
Offset: 0x41c0

INTR_CTLR_CHANNEL8_SLICE8_VIRQ_0
Offset: 0x4200

INTR_CTLR_CHANNEL8_SLICE9_VIRQ_0
Offset: 0x4240

INTR_CTLR_CHANNEL8_SLICE10_VIRQ_0
Offset: 0x4280

INTR_CTLR_CHANNEL8_SLICE11_VIRQ_0
Offset: 0x42c0

INTR_CTLR_CHANNEL8_SLICE12_VIRQ_0
Offset: 0x4300

INTR_CTLR_CHANNEL8_SLICE13_VIRQ_0
Offset: 0x4340

INTR_CTLR_CHANNEL8_SLICE14_VIRQ_0
Offset: 0x4380

INTR_CTLR_CHANNEL8_SLICE15_VIRQ_0
Offset: 0x43c0

INTR_CTLR_CHANNEL8_SLICE16_VIRQ_0
Offset: 0x4400

INTR_CTLR_CHANNEL9_SLICE0_VIRQ_0
Offset: 0x4800

INTR_CTLR_CHANNEL9_SLICE1_VIRQ_0
Offset: 0x4840

INTR_CTLR_CHANNEL9_SLICE2_VIRQ_0
Offset: 0x4880

INTR_CTLR_CHANNEL9_SLICE3_VIRQ_0
Offset: 0x48c0

INTR_CTLR_CHANNEL9_SLICE4_VIRQ_0
Offset: 0x4900

INTR_CTLR_CHANNEL9_SLICE5_VIRQ_0
Offset: 0x4940

INTR_CTLR_CHANNEL9_SLICE6_VIRQ_0
Offset: 0x4980

INTR_CTLR_CHANNEL9_SLICE7_VIRQ_0
Offset: 0x49c0

INTR_CTLR_CHANNEL9_SLICE8_VIRQ_0
Offset: 0x4a00

INTR_CTLR_CHANNEL9_SLICE9_VIRQ_0
Offset: 0x4a40

INTR_CTLR_CHANNEL9_SLICE10_VIRQ_0
Offset: 0x4a80

INTR_CTLR_CHANNEL9_SLICE11_VIRQ_0
Offset: 0x4ac0

INTR_CTLR_CHANNEL9_SLICE12_VIRQ_0
Offset: 0x4b00

INTR_CTLR_CHANNEL9_SLICE13_VIRQ_0
Offset: 0x4b40

INTR_CTLR_CHANNEL9_SLICE14_VIRQ_0
Offset: 0x4b80

INTR_CTLR_CHANNEL9_SLICE15_VIRQ_0
Offset: 0x4bc0

INTR_CTLR_CHANNEL9_SLICE16_VIRQ_0
Offset: 0x4c00

INTR_CTLR_CHANNEL10_SLICE0_VIRQ_0
Offset: 0x5000

INTR_CTLR_CHANNEL10_SLICE1_VIRQ_0
Offset: 0x5040

INTR_CTLR_CHANNEL10_SLICE2_VIRQ_0
Offset: 0x5080

INTR_CTLR_CHANNEL10_SLICE3_VIRQ_0
Offset: 0x50c0

INTR_CTLR_CHANNEL10_SLICE4_VIRQ_0
Offset: 0x5100

INTR_CTLR_CHANNEL10_SLICE5_VIRQ_0
Offset: 0x5140

INTR_CTLR_CHANNEL10_SLICE6_VIRQ_0
Offset: 0x5180

INTR_CTLR_CHANNEL10_SLICE7_VIRQ_0
Offset: 0x51c0

INTR_CTLR_CHANNEL10_SLICE8_VIRQ_0
Offset: 0x5200

INTR_CTLR_CHANNEL10_SLICE9_VIRQ_0
Offset: 0x5240

INTR_CTLR_CHANNEL10_SLICE10_VIRQ_0
Offset: 0x5280

INTR_CTLR_CHANNEL10_SLICE11_VIRQ_0
Offset: 0x52c0

INTR_CTLR_CHANNEL10_SLICE12_VIRQ_0
Offset: 0x5300

INTR_CTLR_CHANNEL10_SLICE13_VIRQ_0
Offset: 0x5340

INTR_CTLR_CHANNEL10_SLICE14_VIRQ_0
Offset: 0x5380

INTR_CTLR_CHANNEL10_SLICE15_VIRQ_0
Offset: 0x53c0

INTR_CTLR_CHANNEL10_SLICE16_VIRQ_0
Offset: 0x5400

INTR_CTLR_CHANNEL11_SLICE0_VIRQ_0
Offset: 0x5800

INTR_CTLR_CHANNEL11_SLICE1_VIRQ_0
Offset: 0x5840

INTR_CTLR_CHANNEL11_SLICE2_VIRQ_0
Offset: 0x5880

INTR_CTLR_CHANNEL11_SLICE3_VIRQ_0
Offset: 0x58c0

INTR_CTLR_CHANNEL11_SLICE4_VIRQ_0
Offset: 0x5900

INTR_CTLR_CHANNEL11_SLICE5_VIRQ_0
Offset: 0x5940

INTR_CTLR_CHANNEL11_SLICE6_VIRQ_0
Offset: 0x5980

INTR_CTLR_CHANNEL11_SLICE7_VIRQ_0
Offset: 0x59c0

INTR_CTLR_CHANNEL11_SLICE8_VIRQ_0
Offset: 0x5a00

INTR_CTLR_CHANNEL11_SLICE9_VIRQ_0
Offset: 0x5a40

INTR_CTLR_CHANNEL11_SLICE10_VIRQ_0
Offset: 0x5a80

INTR_CTLR_CHANNEL11_SLICE11_VIRQ_0
Offset: 0x5ac0

INTR_CTLR_CHANNEL11_SLICE12_VIRQ_0
Offset: 0x5b00

INTR_CTLR_CHANNEL11_SLICE13_VIRQ_0
Offset: 0x5b40

INTR_CTLR_CHANNEL11_SLICE14_VIRQ_0
Offset: 0x5b80

INTR_CTLR_CHANNEL11_SLICE15_VIRQ_0
Offset: 0x5bc0

INTR_CTLR_CHANNEL11_SLICE16_VIRQ_0
Offset: 0x5c00

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: INTR_CTLR_CHANNEL<*i*>_SCR_CHANNEL<*i*>_SCRE_0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**VIRQ**

INTR_CTLR_CHANNEL<*i*>_SLICE<*j*>_VFIQ_0,

where <*i*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, and

<*j*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16.

This register indicates the valid FIQ status for the specific Channel/Slice.

INTR_CTLR_CHANNEL0_SLICE0_VFIQ_0
Offset: 0x4

INTR_CTLR_CHANNEL0_SLICE1_VFIQ_0
Offset: 0x44

INTR_CTLR_CHANNEL0_SLICE2_VFIQ_0
Offset: 0x84

INTR_CTLR_CHANNEL0_SLICE3_VFIQ_0
Offset: 0xc4

INTR_CTLR_CHANNEL0_SLICE4_VFIQ_0
Offset: 0x104

INTR_CTLR_CHANNEL0_SLICE5_VFIQ_0
Offset: 0x144

INTR_CTLR_CHANNEL0_SLICE6_VFIQ_0
Offset: 0x184

INTR_CTLR_CHANNEL0_SLICE7_VFIQ_0
Offset: 0x1c4

INTR_CTLR_CHANNEL0_SLICE8_VFIQ_0
Offset: 0x204

INTR_CTLR_CHANNEL0_SLICE9_VFIQ_0
Offset: 0x244

INTR_CTLR_CHANNEL0_SLICE10_VFIQ_0
Offset: 0x284

INTR_CTLR_CHANNEL0_SLICE11_VFIQ_0
Offset: 0x2c4

INTR_CTLR_CHANNEL0_SLICE12_VFIQ_0
Offset: 0x30

INTR_CTLR_CHANNEL0_SLICE13_VFIQ_0
Offset: 0x344

INTR_CTLR_CHANNEL0_SLICE14_VFIQ_0
Offset: 0x384

INTR_CTLR_CHANNEL0_SLICE15_VFIQ_0
Offset: 0x3c4

INTR_CTLR_CHANNEL0_SLICE16_VFIQ_0
Offset: 0x404

INTR_CTLR_CHANNEL1_SLICE0_VFIQ_0
Offset: 0x804

INTR_CTLR_CHANNEL1_SLICE1_VFIQ_0
Offset: 0x844

INTR_CTLR_CHANNEL1_SLICE2_VFIQ_0
Offset: 0x884

INTR_CTLR_CHANNEL1_SLICE3_VFIQ_0
Offset: 0x8c4

INTR_CTLR_CHANNEL1_SLICE4_VFIQ_0
Offset: 0x904

INTR_CTLR_CHANNEL1_SLICE5_VFIQ_0
Offset: 0x944

INTR_CTLR_CHANNEL1_SLICE6_VFIQ_0
Offset: 0x984

INTR_CTLR_CHANNEL1_SLICE7_VFIQ_0
Offset: 0x9c4

INTR_CTLR_CHANNEL1_SLICE8_VFIQ_0
Offset: 0xa04

INTR_CTLR_CHANNEL1_SLICE9_VFIQ_0
Offset: 0xa44

INTR_CTLR_CHANNEL1_SLICE10_VFIQ_0
Offset: 0xa84

INTR_CTLR_CHANNEL1_SLICE11_VFIQ_0
Offset: 0xac4

INTR_CTLR_CHANNEL1_SLICE12_VFIQ_0
Offset: 0xb04

INTR_CTLR_CHANNEL1_SLICE13_VFIQ_0
Offset: 0xb44

INTR_CTLR_CHANNEL1_SLICE14_VFIQ_0
Offset: 0xb84

INTR_CTLR_CHANNEL1_SLICE15_VFIQ_0
Offset: 0xbc4

INTR_CTLR_CHANNEL1_SLICE16_VFIQ_0
Offset: 0xc04

INTR_CTLR_CHANNEL2_SLICE0_VFIQ_0
Offset: 0x1004

INTR_CTLR_CHANNEL2_SLICE1_VFIQ_0
Offset: 0x1044

INTR_CTLR_CHANNEL2_SLICE2_VFIQ_0
Offset: 0x1084

INTR_CTLR_CHANNEL2_SLICE3_VFIQ_0
Offset: 0x10c4

INTR_CTLR_CHANNEL2_SLICE4_VFIQ_0
Offset: 0x1104

INTR_CTLR_CHANNEL2_SLICE5_VFIQ_0
Offset: 0x1144

INTR_CTLR_CHANNEL2_SLICE6_VFIQ_0
Offset: 0x1184

INTR_CTLR_CHANNEL2_SLICE7_VFIQ_0
Offset: 0x11c4

INTR_CTLR_CHANNEL2_SLICE8_VFIQ_0
Offset: 0x1204

INTR_CTLR_CHANNEL2_SLICE9_VFIQ_0
Offset: 0x1244

INTR_CTLR_CHANNEL2_SLICE10_VFIQ_0
Offset: 0x1284

INTR_CTLR_CHANNEL2_SLICE11_VFIQ_0
Offset: 0x12c4

INTR_CTLR_CHANNEL2_SLICE12_VFIQ_0
Offset: 0x1304

INTR_CTLR_CHANNEL2_SLICE13_VFIQ_0
Offset: 0x1344

INTR_CTLR_CHANNEL2_SLICE14_VFIQ_0
Offset: 0x1384

INTR_CTLR_CHANNEL2_SLICE15_VFIQ_0
Offset: 0x13c4

INTR_CTLR_CHANNEL2_SLICE16_VFIQ_0
Offset: 0x1404

INTR_CTLR_CHANNEL3_SLICE0_VFIQ_0
Offset: 0x1804

INTR_CTLR_CHANNEL3_SLICE1_VFIQ_0
Offset: 0x1844

INTR_CTLR_CHANNEL3_SLICE2_VFIQ_0
Offset: 0x1884

INTR_CTLR_CHANNEL3_SLICE3_VFIQ_0
Offset: 0x18c4

INTR_CTLR_CHANNEL3_SLICE4_VFIQ_0
Offset: 0x1904

INTR_CTLR_CHANNEL3_SLICE5_VFIQ_0
Offset: 0x1944

INTR_CTLR_CHANNEL3_SLICE6_VFIQ_0
Offset: 0x1984

INTR_CTLR_CHANNEL3_SLICE7_VFIQ_0
Offset: 0x19c4

INTR_CTLR_CHANNEL3_SLICE8_VFIQ_0
Offset: 0x1a04

INTR_CTLR_CHANNEL3_SLICE9_VFIQ_0
Offset: 0x1a44

INTR_CTLR_CHANNEL3_SLICE10_VFIQ_0
Offset: 0x1a84

INTR_CTLR_CHANNEL3_SLICE11_VFIQ_0
Offset: 0x1ac4

INTR_CTLR_CHANNEL3_SLICE12_VFIQ_0
Offset: 0x1b04

INTR_CTLR_CHANNEL3_SLICE13_VFIQ_0
Offset: 0x1b44

INTR_CTLR_CHANNEL3_SLICE14_VFIQ_0
Offset: 0x1b84

INTR_CTLR_CHANNEL3_SLICE15_VFIQ_0
Offset: 0x1bc4

INTR_CTLR_CHANNEL3_SLICE16_VFIQ_0
Offset: 0x1c04

INTR_CTLR_CHANNEL4_SLICE0_VFIQ_0
Offset: 0x2004

INTR_CTLR_CHANNEL4_SLICE1_VFIQ_0
Offset: 0x2044

INTR_CTLR_CHANNEL4_SLICE2_VFIQ_0
Offset: 0x2084

INTR_CTLR_CHANNEL4_SLICE3_VFIQ_0
Offset: 0x20c4

INTR_CTLR_CHANNEL4_SLICE4_VFIQ_0
Offset: 0x2104

INTR_CTLR_CHANNEL4_SLICE5_VFIQ_0
Offset: 0x2144

INTR_CTLR_CHANNEL4_SLICE6_VFIQ_0
Offset: 0x2184

INTR_CTLR_CHANNEL4_SLICE7_VFIQ_0
Offset: 0x21c4

INTR_CTLR_CHANNEL4_SLICE8_VFIQ_0
Offset: 0x2204

INTR_CTLR_CHANNEL4_SLICE9_VFIQ_0
Offset: 0x2244

INTR_CTLR_CHANNEL4_SLICE10_VFIQ_0
Offset: 0x2284

INTR_CTLR_CHANNEL4_SLICE11_VFIQ_0
Offset: 0x22c4

INTR_CTLR_CHANNEL4_SLICE12_VFIQ_0
Offset: 0x2304

INTR_CTLR_CHANNEL4_SLICE13_VFIQ_0
Offset: 0x2344

INTR_CTLR_CHANNEL4_SLICE14_VFIQ_0
Offset: 0x2384

INTR_CTLR_CHANNEL4_SLICE15_VFIQ_0
Offset: 0x23c4

INTR_CTLR_CHANNEL4_SLICE16_VFIQ_0
Offset: 0x2404

INTR_CTLR_CHANNEL5_SLICE0_VFIQ_0
Offset: 0x2804

INTR_CTLR_CHANNEL5_SLICE1_VFIQ_0
Offset: 0x2844

INTR_CTLR_CHANNEL5_SLICE2_VFIQ_0
Offset: 0x2884

INTR_CTLR_CHANNEL5_SLICE3_VFIQ_0
Offset: 0x28c4

INTR_CTLR_CHANNEL5_SLICE4_VFIQ_0
Offset: 0x2904

INTR_CTLR_CHANNEL5_SLICE5_VFIQ_0
Offset: 0x2944

INTR_CTLR_CHANNEL5_SLICE6_VFIQ_0
Offset: 0x2984

INTR_CTLR_CHANNEL5_SLICE7_VFIQ_0
Offset: 0x29c4

INTR_CTLR_CHANNEL5_SLICE8_VFIQ_0
Offset: 0x2a04

INTR_CTLR_CHANNEL5_SLICE9_VFIQ_0
Offset: 0x2a44

INTR_CTLR_CHANNEL5_SLICE10_VFIQ_0
Offset: 0x2a84

INTR_CTLR_CHANNEL5_SLICE11_VFIQ_0
Offset: 0x2ac4

INTR_CTLR_CHANNEL5_SLICE12_VFIQ_0
Offset: 0x2b04

INTR_CTLR_CHANNEL5_SLICE13_VFIQ_0
Offset: 0x2b44

INTR_CTLR_CHANNEL5_SLICE14_VFIQ_0
Offset: 0x2b84

INTR_CTLR_CHANNEL5_SLICE15_VFIQ_0
Offset: 0x2bc4

INTR_CTLR_CHANNEL5_SLICE16_VFIQ_0
Offset: 0x2c04

INTR_CTLR_CHANNEL6_SLICE0_VFIQ_0
Offset: 0x3004

INTR_CTLR_CHANNEL6_SLICE1_VFIQ_0
Offset: 0x3044

INTR_CTLR_CHANNEL6_SLICE2_VFIQ_0
Offset: 0x3084

INTR_CTLR_CHANNEL6_SLICE3_VFIQ_0
Offset: 0x30c4

INTR_CTLR_CHANNEL6_SLICE4_VFIQ_0
Offset: 0x3104

INTR_CTLR_CHANNEL6_SLICE5_VFIQ_0
Offset: 0x3144

INTR_CTLR_CHANNEL6_SLICE6_VFIQ_0
Offset: 0x3184

INTR_CTLR_CHANNEL6_SLICE7_VFIQ_0
Offset: 0x31c4

INTR_CTLR_CHANNEL6_SLICE8_VFIQ_0
Offset: 0x3204

INTR_CTLR_CHANNEL6_SLICE9_VFIQ_0
Offset: 0x3244

INTR_CTLR_CHANNEL6_SLICE10_VFIQ_0
Offset: 0x3284

INTR_CTLR_CHANNEL6_SLICE11_VFIQ_0
Offset: 0x32c4

INTR_CTLR_CHANNEL6_SLICE12_VFIQ_0
Offset: 0x3304

INTR_CTLR_CHANNEL6_SLICE13_VFIQ_0
Offset: 0x3344

INTR_CTLR_CHANNEL6_SLICE14_VFIQ_0
Offset: 0x3384

INTR_CTLR_CHANNEL6_SLICE15_VFIQ_0
Offset: 0x33c4

INTR_CTLR_CHANNEL6_SLICE16_VFIQ_0
Offset: 0x3404

INTR_CTLR_CHANNEL7_SLICE0_VFIQ_0
Offset: 0x3804

INTR_CTLR_CHANNEL7_SLICE1_VFIQ_0
Offset: 0x3844

INTR_CTLR_CHANNEL7_SLICE2_VFIQ_0
Offset: 0x3884

INTR_CTLR_CHANNEL7_SLICE3_VFIQ_0
Offset: 0x38c4

INTR_CTLR_CHANNEL7_SLICE4_VFIQ_0
Offset: 0x3904

INTR_CTLR_CHANNEL7_SLICE5_VFIQ_0
Offset: 0x3944

INTR_CTLR_CHANNEL7_SLICE6_VFIQ_0
Offset: 0x3984

INTR_CTLR_CHANNEL7_SLICE7_VFIQ_0
Offset: 0x39c4

INTR_CTLR_CHANNEL7_SLICE8_VFIQ_0
Offset: 0x3a04

INTR_CTLR_CHANNEL7_SLICE9_VFIQ_0
Offset: 0x3a44

INTR_CTLR_CHANNEL7_SLICE10_VFIQ_0
Offset: 0x3a84

INTR_CTLR_CHANNEL7_SLICE11_VFIQ_0
Offset: 0x3ac4

INTR_CTLR_CHANNEL7_SLICE12_VFIQ_0
Offset: 0x3b04

INTR_CTLR_CHANNEL7_SLICE13_VFIQ_0
Offset: 0x3b44

INTR_CTLR_CHANNEL7_SLICE14_VFIQ_0
Offset: 0x3b84

INTR_CTLR_CHANNEL7_SLICE15_VFIQ_0
Offset: 0x3bc4

INTR_CTLR_CHANNEL7_SLICE16_VFIQ_0
Offset: 0x3c04

INTR_CTLR_CHANNEL8_SLICE0_VFIQ_0
Offset: 0x4004

INTR_CTLR_CHANNEL8_SLICE1_VFIQ_0
Offset: 0x4044

INTR_CTLR_CHANNEL8_SLICE2_VFIQ_0
Offset: 0x4084

INTR_CTLR_CHANNEL8_SLICE3_VFIQ_0
Offset: 0x40c4

INTR_CTLR_CHANNEL8_SLICE4_VFIQ_0
Offset: 0x4104

INTR_CTLR_CHANNEL8_SLICE5_VFIQ_0
Offset: 0x4144

INTR_CTLR_CHANNEL8_SLICE6_VFIQ_0
Offset: 0x4184

INTR_CTLR_CHANNEL8_SLICE7_VFIQ_0
Offset: 0x41c4

INTR_CTLR_CHANNEL8_SLICE8_VFIQ_0
Offset: 0x4204

INTR_CTLR_CHANNEL8_SLICE9_VFIQ_0
Offset: 0x4244

INTR_CTLR_CHANNEL8_SLICE10_VFIQ_0
Offset: 0x4284

INTR_CTLR_CHANNEL8_SLICE11_VFIQ_0
Offset: 0x42c4

INTR_CTLR_CHANNEL8_SLICE12_VFIQ_0
Offset: 0x4304

INTR_CTLR_CHANNEL8_SLICE13_VFIQ_0
Offset: 0x4344

INTR_CTLR_CHANNEL8_SLICE14_VFIQ_0
Offset: 0x4384

INTR_CTLR_CHANNEL8_SLICE15_VFIQ_0
Offset: 0x43c4

INTR_CTLR_CHANNEL8_SLICE16_VFIQ_0
Offset: 0x4404

INTR_CTLR_CHANNEL9_SLICE0_VFIQ_0
Offset: 0x4804

INTR_CTLR_CHANNEL9_SLICE1_VFIQ_0
Offset: 0x4844

INTR_CTLR_CHANNEL9_SLICE2_VFIQ_0
Offset: 0x4884

INTR_CTLR_CHANNEL9_SLICE3_VFIQ_0
Offset: 0x48c4

INTR_CTLR_CHANNEL9_SLICE4_VFIQ_0
Offset: 0x4904

INTR_CTLR_CHANNEL9_SLICE5_VFIQ_0
Offset: 0x4944

INTR_CTLR_CHANNEL9_SLICE6_VFIQ_0
Offset: 0x4984

INTR_CTLR_CHANNEL9_SLICE7_VFIQ_0
Offset: 0x49c4

INTR_CTLR_CHANNEL9_SLICE8_VFIQ_0
Offset: 0x4a04

INTR_CTLR_CHANNEL9_SLICE9_VFIQ_0
Offset: 0x4a44

INTR_CTLR_CHANNEL9_SLICE10_VFIQ_0
Offset: 0x4a84

INTR_CTLR_CHANNEL9_SLICE11_VFIQ_0
Offset: 0x4ac4

INTR_CTLR_CHANNEL9_SLICE12_VFIQ_0
Offset: 0x4b04

INTR_CTLR_CHANNEL9_SLICE13_VFIQ_0
Offset: 0x4b44

INTR_CTLR_CHANNEL9_SLICE14_VFIQ_0
Offset: 0x4b84

INTR_CTLR_CHANNEL9_SLICE15_VFIQ_0
Offset: 0x4bc4

INTR_CTLR_CHANNEL9_SLICE16_VFIQ_0
Offset: 0x4c04

INTR_CTLR_CHANNEL10_SLICE0_VFIQ_0
Offset: 0x5004

INTR_CTLR_CHANNEL10_SLICE1_VFIQ_0
Offset: 0x5044

INTR_CTLR_CHANNEL10_SLICE2_VFIQ_0
Offset: 0x5084

INTR_CTLR_CHANNEL10_SLICE3_VFIQ_0
Offset: 0x50c4

INTR_CTLR_CHANNEL10_SLICE4_VFIQ_0
Offset: 0x5104

INTR_CTLR_CHANNEL10_SLICE5_VFIQ_0
Offset: 0x5144

INTR_CTLR_CHANNEL10_SLICE6_VFIQ_0
Offset: 0x5184

INTR_CTLR_CHANNEL10_SLICE7_VFIQ_0
Offset: 0x51c4

INTR_CTLR_CHANNEL10_SLICE8_VFIQ_0
Offset: 0x5204

INTR_CTLR_CHANNEL10_SLICE9_VFIQ_0
Offset: 0x5244

INTR_CTLR_CHANNEL10_SLICE10_VFIQ_0
Offset: 0x5284

INTR_CTLR_CHANNEL10_SLICE11_VFIQ_0
Offset: 0x52c4

INTR_CTLR_CHANNEL10_SLICE12_VFIQ_0
Offset: 0x5304

INTR_CTLR_CHANNEL10_SLICE13_VFIQ_0
Offset: 0x5344

INTR_CTLR_CHANNEL10_SLICE14_VFIQ_0
Offset: 0x5384

INTR_CTLR_CHANNEL10_SLICE15_VFIQ_0
Offset: 0x53c4

INTR_CTLR_CHANNEL10_SLICE16_VFIQ_0
Offset: 0x5404

INTR_CTLR_CHANNEL11_SLICE0_VFIQ_0
Offset: 0x5804

INTR_CTLR_CHANNEL11_SLICE1_VFIQ_0
Offset: 0x5844

INTR_CTLR_CHANNEL11_SLICE2_VFIQ_0
Offset: 0x5884

INTR_CTLR_CHANNEL11_SLICE3_VFIQ_0
Offset: 0x58c4

INTR_CTLR_CHANNEL11_SLICE4_VFIQ_0
Offset: 0x5904

INTR_CTLR_CHANNEL11_SLICE5_VFIQ_0
Offset: 0x5944

INTR_CTLR_CHANNEL11_SLICE6_VFIQ_0
Offset: 0x5984

INTR_CTLR_CHANNEL11_SLICE7_VFIQ_0
Offset: 0x59c4

INTR_CTLR_CHANNEL11_SLICE8_VFIQ_0
Offset: 0x5a04

INTR_CTLR_CHANNEL11_SLICE9_VFIQ_0
Offset: 0x5a44

INTR_CTLR_CHANNEL11_SLICE10_VFIQ_0
Offset: 0x5a84

INTR_CTLR_CHANNEL11_SLICE11_VFIQ_0
Offset: 0x5ac4

INTR_CTLR_CHANNEL11_SLICE12_VFIQ_0
Offset: 0x5b04

INTR_CTLR_CHANNEL11_SLICE13_VFIQ_0
Offset: 0x5b44

INTR_CTLR_CHANNEL11_SLICE14_VFIQ_0
Offset: 0x5b84

INTR_CTLR_CHANNEL11_SLICE15_VFIQ_0
Offset: 0x5bc4

INTR_CTLR_CHANNEL11_SLICE16_VFIQ_0
Offset: 0x5c04

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: INTR_CTLR_CHANNEL<*i*>_SCR_CHANNEL<*i*>_SCRE_0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**VFIQ**

INTR_CTLR_CHANNEL<*i*>_SLICE<*j*>_IER_0,

where <*i*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, and

<*j*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16.

This Interrupt Enable Register (IER) indicates whether that the corresponding Interrupt is enabled for the specific Channel/Slice.

INTR_CTLR_CHANNEL0_SLICE0_IER_0
Offset: 0x8

INTR_CTLR_CHANNEL0_SLICE1_IER_0
Offset: 0x48

INTR_CTLR_CHANNEL0_SLICE2_IER_0
Offset: 0x88

INTR_CTLR_CHANNEL0_SLICE3_IER_0
Offset: 0xc8

INTR_CTLR_CHANNEL0_SLICE4_IER_0
Offset: 0x108

INTR_CTLR_CHANNEL0_SLICE5_IER_0
Offset: 0x148

INTR_CTLR_CHANNEL0_SLICE6_IER_0
Offset: 0x188

INTR_CTLR_CHANNEL0_SLICE7_IER_0
Offset: 0x1c8

INTR_CTLR_CHANNEL0_SLICE8_IER_0
Offset: 0x208

INTR_CTLR_CHANNEL0_SLICE9_IER_0
Offset: 0x248

INTR_CTLR_CHANNEL0_SLICE10_IER_0
Offset: 0x288

INTR_CTLR_CHANNEL0_SLICE11_IER_0
Offset: 0x2c8

INTR_CTLR_CHANNEL0_SLICE12_IER_0
Offset: 0x308

INTR_CTLR_CHANNEL0_SLICE13_IER_0
Offset: 0x348

INTR_CTLR_CHANNEL0_SLICE14_IER_0
Offset: 0x388

INTR_CTLR_CHANNEL0_SLICE15_IER_0
Offset: 0x3c8

INTR_CTLR_CHANNEL0_SLICE16_IER_0
Offset: 0x408

INTR_CTLR_CHANNEL1_SLICE0_IER_0
Offset: 0x808

INTR_CTLR_CHANNEL1_SLICE1_IER_0
Offset: 0x848

INTR_CTLR_CHANNEL1_SLICE2_IER_0
Offset: 0x888

INTR_CTLR_CHANNEL1_SLICE3_IER_0
Offset: 0x8c8

INTR_CTLR_CHANNEL1_SLICE4_IER_0
Offset: 0x909

INTR_CTLR_CHANNEL1_SLICE5_IER_0
Offset: 0x948

INTR_CTLR_CHANNEL1_SLICE6_IER_0
Offset: 0x988

INTR_CTLR_CHANNEL1_SLICE7_IER_0
Offset: 0x9c8

INTR_CTLR_CHANNEL1_SLICE8_IER_0
Offset: 0xa08

INTR_CTLR_CHANNEL1_SLICE9_IER_0
Offset: 0xa48

INTR_CTLR_CHANNEL1_SLICE10_IER_0
Offset: 0xa88

INTR_CTLR_CHANNEL1_SLICE11_IER_0
Offset: 0xac8

INTR_CTLR_CHANNEL1_SLICE12_IER_0
Offset: 0xb08

INTR_CTLR_CHANNEL1_SLICE13_IER_0
Offset: 0xb48

INTR_CTLR_CHANNEL1_SLICE14_IER_0
Offset: 0xb88

INTR_CTLR_CHANNEL1_SLICE15_IER_0
Offset: 0xbc8

INTR_CTLR_CHANNEL1_SLICE16_IER_0
Offset: 0xc08

INTR_CTLR_CHANNEL2_SLICE0_IER_0
Offset: 0x1008

INTR_CTLR_CHANNEL2_SLICE1_IER_0
Offset: 0x1048

INTR_CTLR_CHANNEL2_SLICE2_IER_0
Offset: 0x1088

INTR_CTLR_CHANNEL2_SLICE3_IER_0
Offset: 0x10c8

INTR_CTLR_CHANNEL2_SLICE4_IER_0
Offset: 0x1108

INTR_CTLR_CHANNEL2_SLICE5_IER_0
Offset: 0x1148

INTR_CTLR_CHANNEL2_SLICE6_IER_0
Offset: 0x1188

INTR_CTLR_CHANNEL2_SLICE7_IER_0
Offset: 0x11c8

INTR_CTLR_CHANNEL2_SLICE8_IER_0
Offset: 0x1208

INTR_CTLR_CHANNEL2_SLICE9_IER_0
Offset: 0x1248

INTR_CTLR_CHANNEL2_SLICE10_IER_0
Offset: 0x1288

INTR_CTLR_CHANNEL2_SLICE11_IER_0
Offset: 0x12c8

INTR_CTLR_CHANNEL2_SLICE12_IER_0
Offset: 0x1308

INTR_CTLR_CHANNEL2_SLICE13_IER_0
Offset: 0x1348

INTR_CTLR_CHANNEL2_SLICE14_IER_0
Offset: 0x1388

INTR_CTLR_CHANNEL2_SLICE15_IER_0
Offset: 0x13c8

INTR_CTLR_CHANNEL2_SLICE16_IER_0
Offset: 0x1408

INTR_CTLR_CHANNEL3_SLICE0_IER_0
Offset: 0x1808

INTR_CTLR_CHANNEL3_SLICE1_IER_0
Offset: 0x1848

INTR_CTLR_CHANNEL3_SLICE2_IER_0
Offset: 0x1888

INTR_CTLR_CHANNEL3_SLICE3_IER_0
Offset: 0x18c8

INTR_CTLR_CHANNEL3_SLICE4_IER_0
Offset: 0x1908

INTR_CTLR_CHANNEL3_SLICE5_IER_0
Offset: 0x1948

INTR_CTLR_CHANNEL3_SLICE6_IER_0
Offset: 0x1988

INTR_CTLR_CHANNEL3_SLICE7_IER_0
Offset: 0x19c8

INTR_CTLR_CHANNEL3_SLICE8_IER_0
Offset: 0x1a08

INTR_CTLR_CHANNEL3_SLICE9_IER_0
Offset: 0x1a48

INTR_CTLR_CHANNEL3_SLICE10_IER_0
Offset: 0x1a88

INTR_CTLR_CHANNEL3_SLICE11_IER_0
Offset: 0x1ac8

INTR_CTLR_CHANNEL3_SLICE12_IER_0
Offset: 0x1b08

INTR_CTLR_CHANNEL3_SLICE13_IER_0
Offset: 0x1b48

INTR_CTLR_CHANNEL3_SLICE14_IER_0
Offset: 0x1b88

INTR_CTLR_CHANNEL3_SLICE15_IER_0
Offset: 0x1bc8

INTR_CTLR_CHANNEL3_SLICE16_IER_0
Offset: 0x1c08

INTR_CTLR_CHANNEL4_SLICE0_IER_0
Offset: 0x2008

INTR_CTLR_CHANNEL4_SLICE1_IER_0
Offset: 0x2048

INTR_CTLR_CHANNEL4_SLICE2_IER_0
Offset: 0x2088

INTR_CTLR_CHANNEL4_SLICE3_IER_0
Offset: 0x20c8

INTR_CTLR_CHANNEL4_SLICE4_IER_0
Offset: 0x2108

INTR_CTLR_CHANNEL4_SLICE5_IER_0
Offset: 0x2148

INTR_CTLR_CHANNEL4_SLICE6_IER_0
Offset: 0x2188

INTR_CTLR_CHANNEL4_SLICE7_IER_0
Offset: 0x21c8

INTR_CTLR_CHANNEL4_SLICE8_IER_0
Offset: 0x2208

INTR_CTLR_CHANNEL4_SLICE9_IER_0
Offset: 0x2248

INTR_CTLR_CHANNEL4_SLICE10_IER_0
Offset: 0x2288

INTR_CTLR_CHANNEL4_SLICE11_IER_0
Offset: 0x22c8

INTR_CTLR_CHANNEL4_SLICE12_IER_0
Offset: 0x2308

INTR_CTLR_CHANNEL4_SLICE13_IER_0
Offset: 0x2348

INTR_CTLR_CHANNEL4_SLICE14_IER_0
Offset: 0x2388

INTR_CTLR_CHANNEL4_SLICE15_IER_0
Offset: 0x23c8

INTR_CTLR_CHANNEL4_SLICE16_IER_0
Offset: 0x2408

INTR_CTLR_CHANNEL5_SLICE0_IER_0
Offset: 0x2808

INTR_CTLR_CHANNEL5_SLICE1_IER_0
Offset: 0x2848

INTR_CTLR_CHANNEL5_SLICE2_IER_0
Offset: 0x2888

INTR_CTLR_CHANNEL5_SLICE3_IER_0
Offset: 0x28c8

INTR_CTLR_CHANNEL5_SLICE4_IER_0
Offset: 0x2908

INTR_CTLR_CHANNEL5_SLICE5_IER_0
Offset: 0x2948

INTR_CTLR_CHANNEL5_SLICE6_IER_0
Offset: 0x2988

INTR_CTLR_CHANNEL5_SLICE7_IER_0
Offset: 0x29c8

INTR_CTLR_CHANNEL5_SLICE8_IER_0
Offset: 0x2a08

INTR_CTLR_CHANNEL5_SLICE9_IER_0
Offset: 0x2a48

INTR_CTLR_CHANNEL5_SLICE10_IER_0
Offset: 0x2a88

INTR_CTLR_CHANNEL5_SLICE11_IER_0
Offset: 0x2ac8

INTR_CTLR_CHANNEL5_SLICE12_IER_0
Offset: 0x2b08

INTR_CTLR_CHANNEL5_SLICE13_IER_0
Offset: 0x2b48

INTR_CTLR_CHANNEL5_SLICE14_IER_0
Offset: 0x2b88

INTR_CTLR_CHANNEL5_SLICE15_IER_0
Offset: 0x2bc8

INTR_CTLR_CHANNEL5_SLICE16_IER_0
Offset: 0x2c08

INTR_CTLR_CHANNEL6_SLICE0_IER_0
Offset: 0x3008

INTR_CTLR_CHANNEL6_SLICE1_IER_0
Offset: 0x3048

INTR_CTLR_CHANNEL6_SLICE2_IER_0
Offset: 0x3088

INTR_CTLR_CHANNEL6_SLICE3_IER_0
Offset: 0x30c8

INTR_CTLR_CHANNEL6_SLICE4_IER_0
Offset: 0x3108

INTR_CTLR_CHANNEL6_SLICE5_IER_0
Offset: 0x3148

INTR_CTLR_CHANNEL6_SLICE6_IER_0
Offset: 0x3188

INTR_CTLR_CHANNEL6_SLICE7_IER_0
Offset: 0x31c8

INTR_CTLR_CHANNEL6_SLICE8_IER_0
Offset: 0x3208

INTR_CTLR_CHANNEL6_SLICE9_IER_0
Offset: 0x3248

INTR_CTLR_CHANNEL6_SLICE10_IER_0
Offset: 0x3288

INTR_CTLR_CHANNEL6_SLICE11_IER_0
Offset: 0x32c8

INTR_CTLR_CHANNEL6_SLICE12_IER_0
Offset: 0x3308

INTR_CTLR_CHANNEL6_SLICE13_IER_0
Offset: 0x3348

INTR_CTLR_CHANNEL6_SLICE14_IER_0
Offset: 0x3388

INTR_CTLR_CHANNEL6_SLICE15_IER_0
Offset: 0x33c8

INTR_CTLR_CHANNEL6_SLICE16_IER_0
Offset: 0x3408

INTR_CTLR_CHANNEL7_SLICE0_IER_0
Offset: 0x3808

INTR_CTLR_CHANNEL7_SLICE1_IER_0
Offset: 0x3848

INTR_CTLR_CHANNEL7_SLICE2_IER_0
Offset: 0x3888

INTR_CTLR_CHANNEL7_SLICE3_IER_0
Offset: 0x38c8

INTR_CTLR_CHANNEL7_SLICE4_IER_0
Offset: 0x3908

INTR_CTLR_CHANNEL7_SLICE5_IER_0
Offset: 0x3948

INTR_CTLR_CHANNEL7_SLICE6_IER_0
Offset: 0x3988

INTR_CTLR_CHANNEL7_SLICE7_IER_0
Offset: 0x39c8

INTR_CTLR_CHANNEL7_SLICE8_IER_0
Offset: 0x3a08

INTR_CTLR_CHANNEL7_SLICE9_IER_0
Offset: 0x3a48

INTR_CTLR_CHANNEL7_SLICE10_IER_0
Offset: 0x3a88

INTR_CTLR_CHANNEL7_SLICE11_IER_0
Offset: 0x3ac8

INTR_CTLR_CHANNEL7_SLICE12_IER_0
Offset: 0x3b08

INTR_CTLR_CHANNEL7_SLICE13_IER_0
Offset: 0x3b48

INTR_CTLR_CHANNEL7_SLICE14_IER_0
Offset: 0x3b88

INTR_CTLR_CHANNEL7_SLICE15_IER_0
Offset: 0x3bc8

INTR_CTLR_CHANNEL7_SLICE16_IER_0
Offset: 0x3c08

INTR_CTLR_CHANNEL8_SLICE0_IER_0
Offset: 0x4008

INTR_CTLR_CHANNEL8_SLICE1_IER_0
Offset: 0x4048

INTR_CTLR_CHANNEL8_SLICE2_IER_0
Offset: 0x4088

INTR_CTLR_CHANNEL8_SLICE3_IER_0
Offset: 0x40c8

INTR_CTLR_CHANNEL8_SLICE4_IER_0
Offset: 0x4108

INTR_CTLR_CHANNEL8_SLICE5_IER_0
Offset: 0x4148

INTR_CTLR_CHANNEL8_SLICE6_IER_0
Offset: 0x4188

INTR_CTLR_CHANNEL8_SLICE7_IER_0
Offset: 0x41c8

INTR_CTLR_CHANNEL8_SLICE8_IER_0
Offset: 0x4208

INTR_CTLR_CHANNEL8_SLICE9_IER_0
Offset: 0x4248

INTR_CTLR_CHANNEL8_SLICE10_IER_0
Offset: 0x4288

INTR_CTLR_CHANNEL8_SLICE11_IER_0
Offset: 0x42c8

INTR_CTLR_CHANNEL8_SLICE12_IER_0
Offset: 0x4308

INTR_CTLR_CHANNEL8_SLICE13_IER_0
Offset: 0x4348

INTR_CTLR_CHANNEL8_SLICE14_IER_0
Offset: 0x4388

INTR_CTLR_CHANNEL8_SLICE15_IER_0
Offset: 0x43c8

INTR_CTLR_CHANNEL8_SLICE16_IER_0
Offset: 0x4408

INTR_CTLR_CHANNEL9_SLICE0_IER_0
Offset: 0x4808

INTR_CTLR_CHANNEL9_SLICE1_IER_0
Offset: 0x4848

INTR_CTLR_CHANNEL9_SLICE2_IER_0
Offset: 0x4888

INTR_CTLR_CHANNEL9_SLICE3_IER_0
Offset: 0x48c8

INTR_CTLR_CHANNEL9_SLICE4_IER_0
Offset: 0x4908

INTR_CTLR_CHANNEL9_SLICE5_IER_0
Offset: 0x4948

INTR_CTLR_CHANNEL9_SLICE6_IER_0
Offset: 0x4988

INTR_CTLR_CHANNEL9_SLICE7_IER_0
Offset: 0x49c8

INTR_CTLR_CHANNEL9_SLICE8_IER_0
Offset: 0x4a08

INTR_CTLR_CHANNEL9_SLICE9_IER_0
Offset: 0x4a48

INTR_CTLR_CHANNEL9_SLICE10_IER_0
Offset: 0x4a88

INTR_CTLR_CHANNEL9_SLICE11_IER_0
Offset: 0x4ac8

INTR_CTLR_CHANNEL9_SLICE12_IER_0
Offset: 0x4b08

INTR_CTLR_CHANNEL9_SLICE13_IER_0
Offset: 0x4b48

INTR_CTLR_CHANNEL9_SLICE14_IER_0
Offset: 0x4b88

INTR_CTLR_CHANNEL9_SLICE15_IER_0
Offset: 0x4bc8

INTR_CTLR_CHANNEL9_SLICE16_IER_0
Offset: 0x4c08

INTR_CTLR_CHANNEL10_SLICE0_IER_0
Offset: 0x5008

INTR_CTLR_CHANNEL10_SLICE1_IER_0
Offset: 0x5048

INTR_CTLR_CHANNEL10_SLICE2_IER_0
Offset: 0x5088

INTR_CTLR_CHANNEL10_SLICE3_IER_0
Offset: 0x50c8

INTR_CTLR_CHANNEL10_SLICE4_IER_0
Offset: 0x5108

INTR_CTLR_CHANNEL10_SLICE5_IER_0
Offset: 0x5148

INTR_CTLR_CHANNEL10_SLICE6_IER_0
Offset: 0x5188

INTR_CTLR_CHANNEL10_SLICE7_IER_0
Offset: 0x51c8

INTR_CTLR_CHANNEL10_SLICE8_IER_0
Offset: 0x5208

INTR_CTLR_CHANNEL10_SLICE9_IER_0
Offset: 0x5248

INTR_CTLR_CHANNEL10_SLICE10_IER_0
Offset: 0x5288

INTR_CTLR_CHANNEL10_SLICE11_IER_0
Offset: 0x52c8

INTR_CTLR_CHANNEL10_SLICE12_IER_0
Offset: 0x5308

INTR_CTLR_CHANNEL10_SLICE13_IER_0
Offset: 0x5348

INTR_CTLR_CHANNEL10_SLICE14_IER_0
Offset: 0x5388

INTR_CTLR_CHANNEL10_SLICE15_IER_0
Offset: 0x53c8

INTR_CTLR_CHANNEL10_SLICE16_IER_0
Offset: 0x5408

INTR_CTLR_CHANNEL11_SLICE0_IER_0
Offset: 0x5808

INTR_CTLR_CHANNEL11_SLICE1_IER_0
Offset: 0x5848

INTR_CTLR_CHANNEL11_SLICE2_IER_0
Offset: 0x5888

INTR_CTLR_CHANNEL11_SLICE3_IER_0
Offset: 0x58c8

INTR_CTLR_CHANNEL11_SLICE4_IER_0
Offset: 0x5908

INTR_CTLR_CHANNEL11_SLICE5_IER_0
Offset: 0x5948

INTR_CTLR_CHANNEL11_SLICE6_IER_0
Offset: 0x5988

INTR_CTLR_CHANNEL11_SLICE7_IER_0
Offset: 0x59c8

INTR_CTLR_CHANNEL11_SLICE8_IER_0
Offset: 0x5a08

INTR_CTLR_CHANNEL11_SLICE9_IER_0
Offset: 0x5a48

INTR_CTLR_CHANNEL11_SLICE10_IER_0
Offset: 0x5a88

INTR_CTLR_CHANNEL11_SLICE11_IER_0
Offset: 0x5ac8

INTR_CTLR_CHANNEL11_SLICE12_IER_0
Offset: 0x5b08

INTR_CTLR_CHANNEL11_SLICE13_IER_0
Offset: 0x5b48

INTR_CTLR_CHANNEL11_SLICE14_IER_0
Offset: 0x5b88

INTR_CTLR_CHANNEL11_SLICE15_IER_0
Offset: 0x5bc8

INTR_CTLR_CHANNEL11_SLICE16_IER_0
Offset: 0x5c08

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: INTR_CTLR_CHANNEL<*i*>_SCR_CHANNEL<*i*>_SCRE_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**IER**

INTR_CTLR_CHANNEL<*i*>_SLICE<*j*>_IER_SET_0,

where <*i*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, and

<*j*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16.

Each bit of this register is used to set the corresponding bit in the Interrupt Enable Register (IER) for the specific Channel/Slice.

INTR_CTLR_CHANNEL0_SLICE0_IER_SET_0
Offset: 0xc

INTR_CTLR_CHANNEL0_SLICE1_IER_SET_0
Offset: 0x4c

INTR_CTLR_CHANNEL0_SLICE2_IER_SET_0
Offset: 0x8c

INTR_CTLR_CHANNEL0_SLICE3_IER_SET_0
Offset: 0xcc

INTR_CTLR_CHANNEL0_SLICE4_IER_SET_0
Offset: 0x10c

INTR_CTLR_CHANNEL0_SLICE5_IER_SET_0
Offset: 0x14c

INTR_CTLR_CHANNEL0_SLICE6_IER_SET_0
Offset: 0x18c

INTR_CTLR_CHANNEL0_SLICE7_IER_SET_0
Offset: 0x1cc

INTR_CTLR_CHANNEL0_SLICE8_IER_SET_0
Offset: 0x20c

INTR_CTLR_CHANNEL0_SLICE9_IER_SET_0
Offset: 0x24c

INTR_CTLR_CHANNEL0_SLICE10_IER_SET_0
Offset: 0x28c

INTR_CTLR_CHANNEL0_SLICE11_IER_SET_0
Offset: 0x2cc

INTR_CTLR_CHANNEL0_SLICE12_IER_SET_0
Offset: 0x30c

INTR_CTLR_CHANNEL0_SLICE13_IER_SET_0
Offset: 0x34c

INTR_CTLR_CHANNEL0_SLICE14_IER_SET_0
Offset: 0x38c

INTR_CTLR_CHANNEL0_SLICE15_IER_SET_0
Offset: 0x3cc

INTR_CTLR_CHANNEL0_SLICE16_IER_SET_0
Offset: 0x40c

INTR_CTLR_CHANNEL1_SLICE0_IER_SET_0
Offset: 0x80c

INTR_CTLR_CHANNEL1_SLICE1_IER_SET_0
Offset: 0x84c

INTR_CTLR_CHANNEL1_SLICE2_IER_SET_0
Offset: 0x88c

INTR_CTLR_CHANNEL1_SLICE3_IER_SET_0
Offset: 0x8cc

INTR_CTLR_CHANNEL1_SLICE4_IER_SET_0
Offset: 0x90c

INTR_CTLR_CHANNEL1_SLICE5_IER_SET_0
Offset: 0x94c

INTR_CTLR_CHANNEL1_SLICE6_IER_SET_0
Offset: 0x98c

INTR_CTLR_CHANNEL1_SLICE7_IER_SET_0
Offset: 0x9cc

INTR_CTLR_CHANNEL1_SLICE8_IER_SET_0
Offset: 0xa0c

INTR_CTLR_CHANNEL1_SLICE9_IER_SET_0
Offset: 0xa4c

INTR_CTLR_CHANNEL1_SLICE10_IER_SET_0
Offset: 0xa8c

INTR_CTLR_CHANNEL1_SLICE11_IER_SET_0
Offset: 0xacc

INTR_CTLR_CHANNEL1_SLICE12_IER_SET_0
Offset: 0xb0c

INTR_CTLR_CHANNEL1_SLICE13_IER_SET_0
Offset: 0xb4c

INTR_CTLR_CHANNEL1_SLICE14_IER_SET_0
Offset: 0xb8c

INTR_CTLR_CHANNEL1_SLICE15_IER_SET_0
Offset: 0xbcc

INTR_CTLR_CHANNEL1_SLICE16_IER_SET_0
Offset: 0xc0c

INTR_CTLR_CHANNEL2_SLICE0_IER_SET_0
Offset: 0x100c

INTR_CTLR_CHANNEL2_SLICE1_IER_SET_0
Offset: 0x104c

INTR_CTLR_CHANNEL2_SLICE2_IER_SET_0
Offset: 0x108c

INTR_CTLR_CHANNEL2_SLICE3_IER_SET_0
Offset: 0x10cc

INTR_CTLR_CHANNEL2_SLICE4_IER_SET_0
Offset: 0x110c

INTR_CTLR_CHANNEL2_SLICE5_IER_SET_0
Offset: 0x114c

INTR_CTLR_CHANNEL2_SLICE6_IER_SET_0
Offset: 0x118c

INTR_CTLR_CHANNEL2_SLICE7_IER_SET_0
Offset: 0x11cc

INTR_CTLR_CHANNEL2_SLICE8_IER_SET_0
Offset: 0x120c

INTR_CTLR_CHANNEL2_SLICE9_IER_SET_0
Offset: 0x124c

INTR_CTLR_CHANNEL2_SLICE10_IER_SET_0
Offset: 0x128c

INTR_CTLR_CHANNEL2_SLICE11_IER_SET_0
Offset: 0x12cc

INTR_CTLR_CHANNEL2_SLICE12_IER_SET_0
Offset: 0x130c

INTR_CTLR_CHANNEL2_SLICE13_IER_SET_0
Offset: 0x134c

INTR_CTLR_CHANNEL2_SLICE14_IER_SET_0
Offset: 0x138c

INTR_CTLR_CHANNEL2_SLICE15_IER_SET_0
Offset: 0x13cc

INTR_CTLR_CHANNEL2_SLICE16_IER_SET_0
Offset: 0x140c

INTR_CTLR_CHANNEL3_SLICE0_IER_SET_0
Offset: 0x180c

INTR_CTLR_CHANNEL3_SLICE1_IER_SET_0
Offset: 0x184c

INTR_CTLR_CHANNEL3_SLICE2_IER_SET_0
Offset: 0x188c

INTR_CTLR_CHANNEL3_SLICE3_IER_SET_0
Offset: 0x18cc

INTR_CTLR_CHANNEL3_SLICE4_IER_SET_0
Offset: 0x190c

INTR_CTLR_CHANNEL3_SLICE5_IER_SET_0
Offset: 0x194c

INTR_CTLR_CHANNEL3_SLICE6_IER_SET_0
Offset: 0x198c

INTR_CTLR_CHANNEL3_SLICE7_IER_SET_0
Offset: 0x19cc

INTR_CTLR_CHANNEL3_SLICE8_IER_SET_0
Offset: 0x1a0c

INTR_CTLR_CHANNEL3_SLICE9_IER_SET_0
Offset: 0x1a4c

INTR_CTLR_CHANNEL3_SLICE10_IER_SET_0
Offset: 0x1a8c

INTR_CTLR_CHANNEL3_SLICE11_IER_SET_0
Offset: 0x1acc

INTR_CTLR_CHANNEL3_SLICE12_IER_SET_0
Offset: 0x1b0c

INTR_CTLR_CHANNEL3_SLICE13_IER_SET_0
Offset: 0x1b4c

INTR_CTLR_CHANNEL3_SLICE14_IER_SET_0
Offset: 0x1b8c

INTR_CTLR_CHANNEL3_SLICE15_IER_SET_0
Offset: 0x1bcc

INTR_CTLR_CHANNEL3_SLICE16_IER_SET_0
Offset: 0x1c0c

INTR_CTLR_CHANNEL4_SLICE0_IER_SET_0
Offset: 0x200c

INTR_CTLR_CHANNEL4_SLICE1_IER_SET_0
Offset: 0x204c

INTR_CTLR_CHANNEL4_SLICE2_IER_SET_0
Offset: 0x208c

INTR_CTLR_CHANNEL4_SLICE3_IER_SET_0
Offset: 0x20cc

INTR_CTLR_CHANNEL4_SLICE4_IER_SET_0
Offset: 0x210c

INTR_CTLR_CHANNEL4_SLICE5_IER_SET_0
Offset: 0x214c

INTR_CTLR_CHANNEL4_SLICE6_IER_SET_0
Offset: 0x218c

INTR_CTLR_CHANNEL4_SLICE7_IER_SET_0
Offset: 0x21cc

INTR_CTLR_CHANNEL4_SLICE8_IER_SET_0
Offset: 0x220c

INTR_CTLR_CHANNEL4_SLICE9_IER_SET_0
Offset: 0x224c

INTR_CTLR_CHANNEL4_SLICE10_IER_SET_0
Offset: 0x228c

INTR_CTLR_CHANNEL4_SLICE11_IER_SET_0
Offset: 0x22cc

INTR_CTLR_CHANNEL4_SLICE12_IER_SET_0
Offset: 0x230c

INTR_CTLR_CHANNEL4_SLICE13_IER_SET_0
Offset: 0x234c

INTR_CTLR_CHANNEL4_SLICE14_IER_SET_0
Offset: 0x238c

INTR_CTLR_CHANNEL4_SLICE15_IER_SET_0
Offset: 0x23cc

INTR_CTLR_CHANNEL4_SLICE16_IER_SET_0
Offset: 0x240c

INTR_CTLR_CHANNEL5_SLICE0_IER_SET_0
Offset: 0x280c

INTR_CTLR_CHANNEL5_SLICE1_IER_SET_0
Offset: 0x284c

INTR_CTLR_CHANNEL5_SLICE2_IER_SET_0
Offset: 0x288c

INTR_CTLR_CHANNEL5_SLICE3_IER_SET_0
Offset: 0x28cc

INTR_CTLR_CHANNEL5_SLICE4_IER_SET_0
Offset: 0x290c

INTR_CTLR_CHANNEL5_SLICE5_IER_SET_0
Offset: 0x294c

INTR_CTLR_CHANNEL5_SLICE6_IER_SET_0
Offset: 0x298c

INTR_CTLR_CHANNEL5_SLICE7_IER_SET_0
Offset: 0x29cc

INTR_CTLR_CHANNEL5_SLICE8_IER_SET_0
Offset: 0x2a0c

INTR_CTLR_CHANNEL5_SLICE9_IER_SET_0
Offset: 0x2a4c

INTR_CTLR_CHANNEL5_SLICE10_IER_SET_0
Offset: 0x2a8c

INTR_CTLR_CHANNEL5_SLICE11_IER_SET_0
Offset: 0x2acc

INTR_CTLR_CHANNEL5_SLICE12_IER_SET_0
Offset: 0x2b0c

INTR_CTLR_CHANNEL5_SLICE13_IER_SET_0
Offset: 0x2b4c

INTR_CTLR_CHANNEL5_SLICE14_IER_SET_0
Offset: 0x2b8c

INTR_CTLR_CHANNEL5_SLICE15_IER_SET_0
Offset: 0x2bcc

INTR_CTLR_CHANNEL5_SLICE16_IER_SET_0
Offset: 0x2c0c

INTR_CTLR_CHANNEL6_SLICE0_IER_SET_0
Offset: 0x300c

INTR_CTLR_CHANNEL6_SLICE1_IER_SET_0
Offset: 0x304c

INTR_CTLR_CHANNEL6_SLICE2_IER_SET_0
Offset: 0x308c

INTR_CTLR_CHANNEL6_SLICE3_IER_SET_0
Offset: 0x30cc

INTR_CTLR_CHANNEL6_SLICE4_IER_SET_0
Offset: 0x310c

INTR_CTLR_CHANNEL6_SLICE5_IER_SET_0
Offset: 0x314c

INTR_CTLR_CHANNEL6_SLICE6_IER_SET_0
Offset: 0x318c

INTR_CTLR_CHANNEL6_SLICE7_IER_SET_0
Offset: 0x31cc

INTR_CTLR_CHANNEL6_SLICE8_IER_SET_0
Offset: 0x320c

INTR_CTLR_CHANNEL6_SLICE9_IER_SET_0
Offset: 0x324c

INTR_CTLR_CHANNEL6_SLICE10_IER_SET_0
Offset: 0x328c

INTR_CTLR_CHANNEL6_SLICE11_IER_SET_0
Offset: 0x32cc

INTR_CTLR_CHANNEL6_SLICE12_IER_SET_0
Offset: 0x330c

INTR_CTLR_CHANNEL6_SLICE13_IER_SET_0
Offset: 0x334c

INTR_CTLR_CHANNEL6_SLICE14_IER_SET_0
Offset: 0x338c

INTR_CTLR_CHANNEL6_SLICE15_IER_SET_0
Offset: 0x33cc

INTR_CTLR_CHANNEL6_SLICE16_IER_SET_0
Offset: 0x340c

INTR_CTLR_CHANNEL7_SLICE0_IER_SET_0
Offset: 0x380c

INTR_CTLR_CHANNEL7_SLICE1_IER_SET_0
Offset: 0x384c

INTR_CTLR_CHANNEL7_SLICE2_IER_SET_0
Offset: 0x388c

INTR_CTLR_CHANNEL7_SLICE3_IER_SET_0
Offset: 0x38cc

INTR_CTLR_CHANNEL7_SLICE4_IER_SET_0
Offset: 0x390c

INTR_CTLR_CHANNEL7_SLICE5_IER_SET_0
Offset: 0x394c

INTR_CTLR_CHANNEL7_SLICE6_IER_SET_0
Offset: 0x398c

INTR_CTLR_CHANNEL7_SLICE7_IER_SET_0
Offset: 0x39cc

INTR_CTLR_CHANNEL7_SLICE8_IER_SET_0
Offset: 0x3a0c

INTR_CTLR_CHANNEL7_SLICE9_IER_SET_0
Offset: 0x3a4c

INTR_CTLR_CHANNEL7_SLICE10_IER_SET_0
Offset: 0x3a8c

INTR_CTLR_CHANNEL7_SLICE11_IER_SET_0
Offset: 0x3acc

INTR_CTLR_CHANNEL7_SLICE12_IER_SET_0
Offset: 0x3b0c

INTR_CTLR_CHANNEL7_SLICE13_IER_SET_0
Offset: 0x3b4c

INTR_CTLR_CHANNEL7_SLICE14_IER_SET_0
Offset: 0x3b8c

INTR_CTLR_CHANNEL7_SLICE15_IER_SET_0
Offset: 0x3bcc

INTR_CTLR_CHANNEL7_SLICE16_IER_SET_0
Offset: 0x3c0c

INTR_CTLR_CHANNEL8_SLICE0_IER_SET_0
Offset: 0x400c

INTR_CTLR_CHANNEL8_SLICE1_IER_SET_0
Offset: 0x404c

INTR_CTLR_CHANNEL8_SLICE2_IER_SET_0
Offset: 0x408c

INTR_CTLR_CHANNEL8_SLICE3_IER_SET_0
Offset: 0x40cc

INTR_CTLR_CHANNEL8_SLICE4_IER_SET_0
Offset: 0x410c

INTR_CTLR_CHANNEL8_SLICE5_IER_SET_0
Offset: 0x414c

INTR_CTLR_CHANNEL8_SLICE6_IER_SET_0
Offset: 0x418c

INTR_CTLR_CHANNEL8_SLICE7_IER_SET_0
Offset: 0x41cc

INTR_CTLR_CHANNEL8_SLICE8_IER_SET_0
Offset: 0x420c

INTR_CTLR_CHANNEL8_SLICE9_IER_SET_0
Offset: 0x424c

INTR_CTLR_CHANNEL8_SLICE10_IER_SET_0
Offset: 0x428c

INTR_CTLR_CHANNEL8_SLICE11_IER_SET_0
Offset: 0x42cc

INTR_CTLR_CHANNEL8_SLICE12_IER_SET_0
Offset: 0x430c

INTR_CTLR_CHANNEL8_SLICE13_IER_SET_0
Offset: 0x434c

INTR_CTLR_CHANNEL8_SLICE14_IER_SET_0
Offset: 0x438c

INTR_CTLR_CHANNEL8_SLICE15_IER_SET_0
Offset: 0x43cc

INTR_CTLR_CHANNEL8_SLICE16_IER_SET_0
Offset: 0x440c

INTR_CTLR_CHANNEL9_SLICE0_IER_SET_0
Offset: 0x480c

INTR_CTLR_CHANNEL9_SLICE1_IER_SET_0
Offset: 0x484c

INTR_CTLR_CHANNEL9_SLICE2_IER_SET_0
Offset: 0x488c

INTR_CTLR_CHANNEL9_SLICE3_IER_SET_0
Offset: 0x48cc

INTR_CTLR_CHANNEL9_SLICE4_IER_SET_0
Offset: 0x490c

INTR_CTLR_CHANNEL9_SLICE5_IER_SET_0
Offset: 0x494c

INTR_CTLR_CHANNEL9_SLICE6_IER_SET_0
Offset: 0x498c

INTR_CTLR_CHANNEL9_SLICE7_IER_SET_0
Offset: 0x49cc

INTR_CTLR_CHANNEL9_SLICE8_IER_SET_0
Offset: 0x4a0c

INTR_CTLR_CHANNEL9_SLICE9_IER_SET_0
Offset: 0x4a4c

INTR_CTLR_CHANNEL9_SLICE10_IER_SET_0
Offset: 0x4a8c

INTR_CTLR_CHANNEL9_SLICE11_IER_SET_0
Offset: 0x4acc

INTR_CTLR_CHANNEL9_SLICE12_IER_SET_0
Offset: 0x4b0c

INTR_CTLR_CHANNEL9_SLICE13_IER_SET_0
Offset: 0x4b4c

INTR_CTLR_CHANNEL9_SLICE14_IER_SET_0
Offset: 0x4b8c

INTR_CTLR_CHANNEL9_SLICE15_IER_SET_0
Offset: 0x4bcc

INTR_CTLR_CHANNEL9_SLICE16_IER_SET_0
Offset: 0x4c0c

INTR_CTLR_CHANNEL10_SLICE0_IER_SET_0
Offset: 0x500c

INTR_CTLR_CHANNEL10_SLICE1_IER_SET_0
Offset: 0x504c

INTR_CTLR_CHANNEL10_SLICE2_IER_SET_0
Offset: 0x508c

INTR_CTLR_CHANNEL10_SLICE3_IER_SET_0
Offset: 0x50cc

INTR_CTLR_CHANNEL10_SLICE4_IER_SET_0
Offset: 0x510c

INTR_CTLR_CHANNEL10_SLICE5_IER_SET_0
Offset: 0x514c

INTR_CTLR_CHANNEL10_SLICE6_IER_SET_0
Offset: 0x518c

INTR_CTLR_CHANNEL10_SLICE7_IER_SET_0
Offset: 0x51cc

INTR_CTLR_CHANNEL10_SLICE8_IER_SET_0
Offset: 0x520c

INTR_CTLR_CHANNEL10_SLICE9_IER_SET_0
Offset: 0x524c

INTR_CTLR_CHANNEL10_SLICE10_IER_SET_0
Offset: 0x528c

INTR_CTLR_CHANNEL10_SLICE11_IER_SET_0
Offset: 0x52cc

INTR_CTLR_CHANNEL10_SLICE12_IER_SET_0
Offset: 0x530c

INTR_CTLR_CHANNEL10_SLICE13_IER_SET_0
Offset: 0x534c

INTR_CTLR_CHANNEL10_SLICE14_IER_SET_0
Offset: 0x538c

INTR_CTLR_CHANNEL10_SLICE15_IER_SET_0
Offset: 0x53cc

INTR_CTLR_CHANNEL10_SLICE16_IER_SET_0
Offset: 0x540c

INTR_CTLR_CHANNEL11_SLICE0_IER_SET_0
Offset: 0x580c

INTR_CTLR_CHANNEL11_SLICE1_IER_SET_0
Offset: 0x584c

INTR_CTLR_CHANNEL11_SLICE2_IER_SET_0
Offset: 0x588c

INTR_CTLR_CHANNEL11_SLICE3_IER_SET_0
Offset: 0x58cc

INTR_CTLR_CHANNEL11_SLICE4_IER_SET_0
Offset: 0x590c

INTR_CTLR_CHANNEL11_SLICE5_IER_SET_0
Offset: 0x594c

INTR_CTLR_CHANNEL11_SLICE6_IER_SET_0
Offset: 0x598c

INTR_CTLR_CHANNEL11_SLICE7_IER_SET_0
Offset: 0x59cc

INTR_CTLR_CHANNEL11_SLICE8_IER_SET_0
Offset: 0x5a0c

INTR_CTLR_CHANNEL11_SLICE9_IER_SET_0
Offset: 0x5a4c

INTR_CTLR_CHANNEL11_SLICE10_IER_SET_0
Offset: 0x5a8c

INTR_CTLR_CHANNEL11_SLICE11_IER_SET_0
Offset: 0x5acc

INTR_CTLR_CHANNEL11_SLICE12_IER_SET_0
Offset: 0x5b0c

INTR_CTLR_CHANNEL11_SLICE13_IER_SET_0
Offset: 0x5b4c

INTR_CTLR_CHANNEL11_SLICE14_IER_SET_0
Offset: 0x5b8c

INTR_CTLR_CHANNEL11_SLICE15_IER_SET_0
Offset: 0x5bcc

INTR_CTLR_CHANNEL11_SLICE16_IER_SET_0
Offset: 0x5c0c

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: INTR_CTLR_CHANNEL<*i*>_SCR_CHANNEL<*i*>_SCRE_0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**IER_SET** Each bit for the corresponding bit in IER. 0: No action. 1: Set the bit.

INTR_CTLR_CHANNEL<*i*>_SLICE<*j*>_IER_CLR_0,

where <*i*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, and

<*j*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16.

Each bit of this register is used to clear the corresponding bit in the Interrupt Enable Register (IER) for the specific Channel/Slice.

INTR_CTLR_CHANNEL0_SLICE0_IER_CLR_0
Offset: 0x10

INTR_CTLR_CHANNEL0_SLICE1_IER_CLR_0
Offset: 0x50

INTR_CTLR_CHANNEL0_SLICE2_IER_CLR_0
Offset: 0x90

INTR_CTLR_CHANNEL0_SLICE3_IER_CLR_0
Offset: 0xd0

INTR_CTLR_CHANNEL0_SLICE4_IER_CLR_0
Offset: 0x110

INTR_CTLR_CHANNEL0_SLICE5_IER_CLR_0
Offset: 0x150

INTR_CTLR_CHANNEL0_SLICE6_IER_CLR_0
Offset: 0x190

INTR_CTLR_CHANNEL0_SLICE7_IER_CLR_0
Offset: 0x1d0

INTR_CTLR_CHANNEL0_SLICE8_IER_CLR_0
Offset: 0x210

INTR_CTLR_CHANNEL0_SLICE9_IER_CLR_0
Offset: 0x250

INTR_CTLR_CHANNEL0_SLICE10_IER_CLR_0
Offset: 0x290

INTR_CTLR_CHANNEL0_SLICE11_IER_CLR_0
Offset: 0x2d0

INTR_CTLR_CHANNEL0_SLICE12_IER_CLR_0
Offset: 0x310

INTR_CTLR_CHANNEL0_SLICE13_IER_CLR_0
Offset: 0x350

INTR_CTLR_CHANNEL0_SLICE14_IER_CLR_0
Offset: 0x390

INTR_CTLR_CHANNEL0_SLICE15_IER_CLR_0
Offset: 0x3d0

INTR_CTLR_CHANNEL0_SLICE16_IER_CLR_0
Offset: 0x410

INTR_CTLR_CHANNEL1_SLICE0_IER_CLR_0
Offset: 0x810

INTR_CTLR_CHANNEL1_SLICE1_IER_CLR_0
Offset: 0x850

INTR_CTLR_CHANNEL1_SLICE2_IER_CLR_0
Offset: 0x890

INTR_CTLR_CHANNEL1_SLICE3_IER_CLR_0
Offset: 0x8d0

INTR_CTLR_CHANNEL1_SLICE4_IER_CLR_0
Offset: 0x910

INTR_CTLR_CHANNEL1_SLICE5_IER_CLR_0
Offset: 0x950

INTR_CTLR_CHANNEL1_SLICE6_IER_CLR_0
Offset: 0x990

INTR_CTLR_CHANNEL1_SLICE7_IER_CLR_0
Offset: 0x9d0

INTR_CTLR_CHANNEL1_SLICE8_IER_CLR_0
Offset: 0xa10

INTR_CTLR_CHANNEL1_SLICE9_IER_CLR_0
Offset: 0xa50

INTR_CTLR_CHANNEL1_SLICE10_IER_CLR_0
Offset: 0xa90

INTR_CTLR_CHANNEL1_SLICE11_IER_CLR_0
Offset: 0xad0

INTR_CTLR_CHANNEL1_SLICE12_IER_CLR_0
Offset: 0xb10

INTR_CTLR_CHANNEL1_SLICE13_IER_CLR_0
Offset: 0xb50

INTR_CTLR_CHANNEL1_SLICE14_IER_CLR_0
Offset: 0xb90

INTR_CTLR_CHANNEL1_SLICE15_IER_CLR_0
Offset: 0xbd0

INTR_CTLR_CHANNEL1_SLICE16_IER_CLR_0
Offset: 0xc10

INTR_CTLR_CHANNEL2_SLICE0_IER_CLR_0
Offset: 0x1010

INTR_CTLR_CHANNEL2_SLICE1_IER_CLR_0
Offset: 0x1050

INTR_CTLR_CHANNEL2_SLICE2_IER_CLR_0
Offset: 0x1090

INTR_CTLR_CHANNEL2_SLICE3_IER_CLR_0
Offset: 0x10d0

INTR_CTLR_CHANNEL2_SLICE4_IER_CLR_0
Offset: 0x1110

INTR_CTLR_CHANNEL2_SLICE5_IER_CLR_0
Offset: 0x1150

INTR_CTLR_CHANNEL2_SLICE6_IER_CLR_0
Offset: 0x1190

INTR_CTLR_CHANNEL2_SLICE7_IER_CLR_0
Offset: 0x11d0

INTR_CTLR_CHANNEL2_SLICE8_IER_CLR_0
Offset: 0x1210

INTR_CTLR_CHANNEL2_SLICE9_IER_CLR_0
Offset: 0x1250

INTR_CTLR_CHANNEL2_SLICE10_IER_CLR_0
Offset: 0x1290

INTR_CTLR_CHANNEL2_SLICE11_IER_CLR_0
Offset: 0x12d0

INTR_CTLR_CHANNEL2_SLICE12_IER_CLR_0
Offset: 0x1310

INTR_CTLR_CHANNEL2_SLICE13_IER_CLR_0
Offset: 0x1350

INTR_CTLR_CHANNEL2_SLICE14_IER_CLR_0
Offset: 0x1390

INTR_CTLR_CHANNEL2_SLICE15_IER_CLR_0
Offset: 0x13d0

INTR_CTLR_CHANNEL2_SLICE16_IER_CLR_0
Offset: 0x1410

INTR_CTLR_CHANNEL3_SLICE0_IER_CLR_0
Offset: 0x1810

INTR_CTLR_CHANNEL3_SLICE1_IER_CLR_0
Offset: 0x1850

INTR_CTLR_CHANNEL3_SLICE2_IER_CLR_0
Offset: 0x1890

INTR_CTLR_CHANNEL3_SLICE3_IER_CLR_0
Offset: 0x18d0

INTR_CTLR_CHANNEL3_SLICE4_IER_CLR_0
Offset: 0x1910

INTR_CTLR_CHANNEL3_SLICE5_IER_CLR_0
Offset: 0x1950

INTR_CTLR_CHANNEL3_SLICE6_IER_CLR_0
Offset: 0x1990

INTR_CTLR_CHANNEL3_SLICE7_IER_CLR_0
Offset: 0x19d0

INTR_CTLR_CHANNEL3_SLICE8_IER_CLR_0
Offset: 0x1a10

INTR_CTLR_CHANNEL3_SLICE9_IER_CLR_0
Offset: 0x1a50

INTR_CTLR_CHANNEL3_SLICE10_IER_CLR_0
Offset: 0x1a90

INTR_CTLR_CHANNEL3_SLICE11_IER_CLR_0
Offset: 0x1ad0

INTR_CTLR_CHANNEL3_SLICE12_IER_CLR_0
Offset: 0x1b10

INTR_CTLR_CHANNEL3_SLICE13_IER_CLR_0
Offset: 0x1b50

INTR_CTLR_CHANNEL3_SLICE14_IER_CLR_0
Offset: 0x1b90

INTR_CTLR_CHANNEL3_SLICE15_IER_CLR_0
Offset: 0x1bd0

INTR_CTLR_CHANNEL3_SLICE16_IER_CLR_0
Offset: 0x1c10

INTR_CTLR_CHANNEL4_SLICE0_IER_CLR_0
Offset: 0x2010

INTR_CTLR_CHANNEL4_SLICE1_IER_CLR_0
Offset: 0x2050

INTR_CTLR_CHANNEL4_SLICE2_IER_CLR_0
Offset: 0x2090

INTR_CTLR_CHANNEL4_SLICE3_IER_CLR_0
Offset: 0x20d0

INTR_CTLR_CHANNEL4_SLICE4_IER_CLR_0
Offset: 0x2110

INTR_CTLR_CHANNEL4_SLICE5_IER_CLR_0
Offset: 0x2150

INTR_CTLR_CHANNEL4_SLICE6_IER_CLR_0
Offset: 0x2190

INTR_CTLR_CHANNEL4_SLICE7_IER_CLR_0
Offset: 0x21d0

INTR_CTLR_CHANNEL4_SLICE8_IER_CLR_0
Offset: 0x2210

INTR_CTLR_CHANNEL4_SLICE9_IER_CLR_0
Offset: 0x2250

INTR_CTLR_CHANNEL4_SLICE10_IER_CLR_0
Offset: 0x2290

INTR_CTLR_CHANNEL4_SLICE11_IER_CLR_0
Offset: 0x22d0

INTR_CTLR_CHANNEL4_SLICE12_IER_CLR_0
Offset: 0x2310

INTR_CTLR_CHANNEL4_SLICE13_IER_CLR_0
Offset: 0x2350

INTR_CTLR_CHANNEL4_SLICE14_IER_CLR_0
Offset: 0x2390

INTR_CTLR_CHANNEL4_SLICE15_IER_CLR_0
Offset: 0x23d0

INTR_CTLR_CHANNEL4_SLICE16_IER_CLR_0
Offset: 0x2410

INTR_CTLR_CHANNEL5_SLICE0_IER_CLR_0
Offset: 0x2810

INTR_CTLR_CHANNEL5_SLICE1_IER_CLR_0
Offset: 0x2850

INTR_CTLR_CHANNEL5_SLICE2_IER_CLR_0
Offset: 0x2890

INTR_CTLR_CHANNEL5_SLICE3_IER_CLR_0
Offset: 0x28d0

INTR_CTLR_CHANNEL5_SLICE4_IER_CLR_0
Offset: 0x2910

INTR_CTLR_CHANNEL5_SLICE5_IER_CLR_0
Offset: 0x2950

INTR_CTLR_CHANNEL5_SLICE6_IER_CLR_0
Offset: 0x2990

INTR_CTLR_CHANNEL5_SLICE7_IER_CLR_0
Offset: 0x29d0

INTR_CTLR_CHANNEL5_SLICE8_IER_CLR_0
Offset: 0x2a10

INTR_CTLR_CHANNEL5_SLICE9_IER_CLR_0
Offset: 0x2a50

INTR_CTLR_CHANNEL5_SLICE10_IER_CLR_0
Offset: 0x2a90

INTR_CTLR_CHANNEL5_SLICE11_IER_CLR_0
Offset: 0x2ad0

INTR_CTLR_CHANNEL5_SLICE12_IER_CLR_0
Offset: 0x2b10

INTR_CTLR_CHANNEL5_SLICE13_IER_CLR_0
Offset: 0x2b50

INTR_CTLR_CHANNEL5_SLICE14_IER_CLR_0
Offset: 0x2b90

INTR_CTLR_CHANNEL5_SLICE15_IER_CLR_0
Offset: 0x2bd0

INTR_CTLR_CHANNEL5_SLICE16_IER_CLR_0
Offset: 0x2c10

INTR_CTLR_CHANNEL6_SLICE0_IER_CLR_0
Offset: 0x3010

INTR_CTLR_CHANNEL6_SLICE1_IER_CLR_0
Offset: 0x3050

INTR_CTLR_CHANNEL6_SLICE2_IER_CLR_0
Offset: 0x3090

INTR_CTLR_CHANNEL6_SLICE3_IER_CLR_0
Offset: 0x30d0

INTR_CTLR_CHANNEL6_SLICE4_IER_CLR_0
Offset: 0x3110

INTR_CTLR_CHANNEL6_SLICE5_IER_CLR_0
Offset: 0x3150

INTR_CTLR_CHANNEL6_SLICE6_IER_CLR_0
Offset: 0x3190

INTR_CTLR_CHANNEL6_SLICE7_IER_CLR_0
Offset: 0x31d0

INTR_CTLR_CHANNEL6_SLICE8_IER_CLR_0
Offset: 0x3210

INTR_CTLR_CHANNEL6_SLICE9_IER_CLR_0
Offset: 0x3250

INTR_CTLR_CHANNEL6_SLICE10_IER_CLR_0
Offset: 0x3290

INTR_CTLR_CHANNEL6_SLICE11_IER_CLR_0
Offset: 0x32d0

INTR_CTLR_CHANNEL6_SLICE12_IER_CLR_0
Offset: 0x3310

INTR_CTLR_CHANNEL6_SLICE13_IER_CLR_0
Offset: 0x3350

INTR_CTLR_CHANNEL6_SLICE14_IER_CLR_0
Offset: 0x3390

INTR_CTLR_CHANNEL6_SLICE15_IER_CLR_0
Offset: 0x33d0

INTR_CTLR_CHANNEL6_SLICE16_IER_CLR_0
Offset: 0x3410

INTR_CTLR_CHANNEL7_SLICE0_IER_CLR_0
Offset: 0x3810

INTR_CTLR_CHANNEL7_SLICE1_IER_CLR_0
Offset: 0x3850

INTR_CTLR_CHANNEL7_SLICE2_IER_CLR_0
Offset: 0x3890

INTR_CTLR_CHANNEL7_SLICE3_IER_CLR_0
Offset: 0x38d0

INTR_CTLR_CHANNEL7_SLICE4_IER_CLR_0
Offset: 0x3910

INTR_CTLR_CHANNEL7_SLICE5_IER_CLR_0
Offset: 0x3950

INTR_CTLR_CHANNEL7_SLICE6_IER_CLR_0
Offset: 0x3990

INTR_CTLR_CHANNEL7_SLICE7_IER_CLR_0
Offset: 0x39d0

INTR_CTLR_CHANNEL7_SLICE8_IER_CLR_0
Offset: 0x3a10

INTR_CTLR_CHANNEL7_SLICE9_IER_CLR_0
Offset: 0x3a50

INTR_CTLR_CHANNEL7_SLICE10_IER_CLR_0
Offset: 0x3a90

INTR_CTLR_CHANNEL7_SLICE11_IER_CLR_0
Offset: 0x3ad0

INTR_CTLR_CHANNEL7_SLICE12_IER_CLR_0
Offset: 0x3b10

INTR_CTLR_CHANNEL7_SLICE13_IER_CLR_0
Offset: 0x3b50

INTR_CTLR_CHANNEL7_SLICE14_IER_CLR_0
Offset: 0x3b90

INTR_CTLR_CHANNEL7_SLICE15_IER_CLR_0
Offset: 0x3bd0

INTR_CTLR_CHANNEL7_SLICE16_IER_CLR_0
Offset: 0x3c10

INTR_CTLR_CHANNEL8_SLICE0_IER_CLR_0
Offset: 0x4010

INTR_CTLR_CHANNEL8_SLICE1_IER_CLR_0
Offset: 0x4050

INTR_CTLR_CHANNEL8_SLICE2_IER_CLR_0
Offset: 0x4090

INTR_CTLR_CHANNEL8_SLICE3_IER_CLR_0
Offset: 0x40d0

INTR_CTLR_CHANNEL8_SLICE4_IER_CLR_0
Offset: 0x4110

INTR_CTLR_CHANNEL8_SLICE5_IER_CLR_0
Offset: 0x4150

INTR_CTLR_CHANNEL8_SLICE6_IER_CLR_0
Offset: 0x4190

INTR_CTLR_CHANNEL8_SLICE7_IER_CLR_0
Offset: 0x41d0

INTR_CTLR_CHANNEL8_SLICE8_IER_CLR_0
Offset: 0x4210

INTR_CTLR_CHANNEL8_SLICE9_IER_CLR_0
Offset: 0x4250

INTR_CTLR_CHANNEL8_SLICE10_IER_CLR_0
Offset: 0x4290

INTR_CTLR_CHANNEL8_SLICE11_IER_CLR_0
Offset: 0x42d0

INTR_CTLR_CHANNEL8_SLICE12_IER_CLR_0
Offset: 0x4310

INTR_CTLR_CHANNEL8_SLICE13_IER_CLR_0
Offset: 0x4350

INTR_CTLR_CHANNEL8_SLICE14_IER_CLR_0
Offset: 0x4390

INTR_CTLR_CHANNEL8_SLICE15_IER_CLR_0
Offset: 0x43d0

INTR_CTLR_CHANNEL8_SLICE16_IER_CLR_0
Offset: 0x4410

INTR_CTLR_CHANNEL9_SLICE0_IER_CLR_0
Offset: 0x4810

INTR_CTLR_CHANNEL9_SLICE1_IER_CLR_0
Offset: 0x4850

INTR_CTLR_CHANNEL9_SLICE2_IER_CLR_0
Offset: 0x4890

INTR_CTLR_CHANNEL9_SLICE3_IER_CLR_0
Offset: 0x48d0

INTR_CTLR_CHANNEL9_SLICE4_IER_CLR_0
Offset: 0x4910

INTR_CTLR_CHANNEL9_SLICE5_IER_CLR_0
Offset: 0x4950

INTR_CTLR_CHANNEL9_SLICE6_IER_CLR_0
Offset: 0x4990

INTR_CTLR_CHANNEL9_SLICE7_IER_CLR_0
Offset: 0x49d0

INTR_CTLR_CHANNEL9_SLICE8_IER_CLR_0
Offset: 0x4a10

INTR_CTLR_CHANNEL9_SLICE9_IER_CLR_0
Offset: 0x4a50

INTR_CTLR_CHANNEL9_SLICE10_IER_CLR_0
Offset: 0x4a90

INTR_CTLR_CHANNEL9_SLICE11_IER_CLR_0
Offset: 0x4ad0

INTR_CTLR_CHANNEL9_SLICE12_IER_CLR_0
Offset: 0x4b10

INTR_CTLR_CHANNEL9_SLICE13_IER_CLR_0
Offset: 0x4b50

INTR_CTLR_CHANNEL9_SLICE14_IER_CLR_0
Offset: 0x4b90

INTR_CTLR_CHANNEL9_SLICE15_IER_CLR_0
Offset: 0x4bd0

INTR_CTLR_CHANNEL9_SLICE16_IER_CLR_0
Offset: 0x4c10

INTR_CTLR_CHANNEL10_SLICE0_IER_CLR_0
Offset: 0x5010

INTR_CTLR_CHANNEL10_SLICE1_IER_CLR_0
Offset: 0x5050

INTR_CTLR_CHANNEL10_SLICE2_IER_CLR_0
Offset: 0x5090

INTR_CTLR_CHANNEL10_SLICE3_IER_CLR_0
Offset: 0x50d0

INTR_CTLR_CHANNEL10_SLICE4_IER_CLR_0
Offset: 0x5110

INTR_CTLR_CHANNEL10_SLICE5_IER_CLR_0
Offset: 0x5150

INTR_CTLR_CHANNEL10_SLICE6_IER_CLR_0
Offset: 0x5190

INTR_CTLR_CHANNEL10_SLICE7_IER_CLR_0
Offset: 0x51d0

INTR_CTLR_CHANNEL10_SLICE8_IER_CLR_0
Offset: 0x5210

INTR_CTLR_CHANNEL10_SLICE9_IER_CLR_0
Offset: 0x5250

INTR_CTLR_CHANNEL10_SLICE10_IER_CLR_0
Offset: 0x5290

INTR_CTLR_CHANNEL10_SLICE11_IER_CLR_0
Offset: 0x52d0

INTR_CTLR_CHANNEL10_SLICE12_IER_CLR_0
Offset: 0x5310

INTR_CTLR_CHANNEL10_SLICE13_IER_CLR_0
Offset: 0x5350

INTR_CTLR_CHANNEL10_SLICE14_IER_CLR_0
Offset: 0x5390

INTR_CTLR_CHANNEL10_SLICE15_IER_CLR_0
Offset: 0x53d0

INTR_CTLR_CHANNEL10_SLICE16_IER_CLR_0
Offset: 0x5410

INTR_CTLR_CHANNEL11_SLICE0_IER_CLR_0
Offset: 0x5810

INTR_CTLR_CHANNEL11_SLICE1_IER_CLR_0
Offset: 0x5850

INTR_CTLR_CHANNEL11_SLICE2_IER_CLR_0
Offset: 0x5890

INTR_CTLR_CHANNEL11_SLICE3_IER_CLR_0
Offset: 0x58d0

INTR_CTLR_CHANNEL11_SLICE4_IER_CLR_0
Offset: 0x5910

INTR_CTLR_CHANNEL11_SLICE5_IER_CLR_0
Offset: 0x5950

INTR_CTLR_CHANNEL11_SLICE6_IER_CLR_0
Offset: 0x5990

INTR_CTLR_CHANNEL11_SLICE7_IER_CLR_0
Offset: 0x59d0

INTR_CTLR_CHANNEL11_SLICE8_IER_CLR_0
Offset: 0x5a10

INTR_CTLR_CHANNEL11_SLICE9_IER_CLR_0
Offset: 0x5a50

INTR_CTLR_CHANNEL11_SLICE10_IER_CLR_0
Offset: 0x5a90

INTR_CTLR_CHANNEL11_SLICE11_IER_CLR_0
Offset: 0x5ad0

INTR_CTLR_CHANNEL11_SLICE12_IER_CLR_0
Offset: 0x5b10

INTR_CTLR_CHANNEL11_SLICE13_IER_CLR_0
Offset: 0x5b50

INTR_CTLR_CHANNEL11_SLICE14_IER_CLR_0
Offset: 0x5b90

INTR_CTLR_CHANNEL11_SLICE15_IER_CLR_0
Offset: 0x5bd0

INTR_CTLR_CHANNEL11_SLICE16_IER_CLR_0
Offset: 0x5c10

Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: INTR_CTLR_CHANNEL<*i*>_SCR_CHANNEL<*i*>_SCRE_0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**IER_CLR** Each bit for the corresponding bit in IER. 0: No action. 1: Clear the bit.

INTR_CTLR_CHANNEL<*i*>_SLICE<*j*>_IEP_CLASS_0,

where <*i*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, and

<*j*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16.

Each bit of this Interrupt Enable Priority Class (IEP_Class) Register selects between IRQ and IFQ for the corresponding Interrupt in the specfc Channel/Slice.

INTR_CTLR_CHANNEL0_SLICE0_IEP_CLASS_0
Offset: 0x14

INTR_CTLR_CHANNEL0_SLICE1_IER_CLASS_0
Offset: 0x54

INTR_CTLR_CHANNEL0_SLICE2_IEP_CLASS_0
Offset: 0x94

INTR_CTLR_CHANNEL0_SLICE3_IEP_CLASS_0
Offset: 0xd4

INTR_CTLR_CHANNEL0_SLICE4_IEP_CLASS_0
Offset: 0x114

INTR_CTLR_CHANNEL0_SLICE5_IEP_CLASS_0
Offset: 0x154

INTR_CTLR_CHANNEL0_SLICE6_IEP_CLASS_0
Offset: 0x194

INTR_CTLR_CHANNEL0_SLICE7_IEP_CLASS_0
Offset: 0x1d4

INTR_CTLR_CHANNEL0_SLICE8_IEP_CLASS_0
Offset: 0x214

INTR_CTLR_CHANNEL0_SLICE9_IEP_CLASS_0
Offset: 0x254

INTR_CTLR_CHANNEL0_SLICE10_IEP_CLASS_0
Offset: 0x294

INTR_CTLR_CHANNEL0_SLICE11_IEP_CLASS_0
Offset: 0x2d4

INTR_CTLR_CHANNEL0_SLICE12_IEP_CLASS_0
Offset: 0x314

INTR_CTLR_CHANNEL0_SLICE13_IEP_CLASS_0
Offset: 0x354

INTR_CTLR_CHANNEL0_SLICE14_IEP_CLASS_0
Offset: 0x394

INTR_CTLR_CHANNEL0_SLICE15_IEP_CLASS_0
Offset: 0x3d4

INTR_CTLR_CHANNEL0_SLICE16_IEP_CLASS_0
Offset: 0x414

INTR_CTLR_CHANNEL1_SLICE0_IEP_CLASS_0
Offset: 0x814

INTR_CTLR_CHANNEL1_SLICE1_IEP_CLASS_0
Offset: 0x854

INTR_CTLR_CHANNEL1_SLICE2_IEP_CLASS_0
Offset: 0x894

INTR_CTLR_CHANNEL1_SLICE3_IEP_CLASS_0
Offset: 0x8d4

INTR_CTLR_CHANNEL1_SLICE4_IEP_CLASS_0
Offset: 0x914

INTR_CTLR_CHANNEL1_SLICE5_IEP_CLASS_0
Offset: 0x954

INTR_CTLR_CHANNEL1_SLICE6_IEP_CLASS_0
Offset: 0x994

INTR_CTLR_CHANNEL1_SLICE7_IEP_CLASS_0
Offset: 0x9d4

INTR_CTLR_CHANNEL1_SLICE8_IEP_CLASS_0
Offset: 0xa14

INTR_CTLR_CHANNEL1_SLICE9_IEP_CLASS_0
Offset: 0xa54

INTR_CTLR_CHANNEL1_SLICE10_IEP_CLASS_0
Offset: 0xa94

INTR_CTLR_CHANNEL1_SLICE11_IEP_CLASS_0
Offset: 0xad4

INTR_CTLR_CHANNEL1_SLICE12_IEP_CLASS_0
Offset: 0xb14

INTR_CTLR_CHANNEL1_SLICE13_IEP_CLASS_0
Offset: 0xb54

INTR_CTLR_CHANNEL1_SLICE14_IEP_CLASS_0
Offset: 0xb94

INTR_CTLR_CHANNEL1_SLICE15_IEP_CLASS_0
Offset: 0xbd4

INTR_CTLR_CHANNEL1_SLICE16_IEP_CLASS_0
Offset: 0xc14

INTR_CTLR_CHANNEL2_SLICE0_IEP_CLASS_0
Offset: 0x1014

INTR_CTLR_CHANNEL2_SLICE1_IEP_CLASS_0
Offset: 0x1054

INTR_CTLR_CHANNEL2_SLICE2_IEP_CLASS_0
Offset: 0x1094

INTR_CTLR_CHANNEL2_SLICE3_IEP_CLASS_0
Offset: 0x10d4

INTR_CTLR_CHANNEL2_SLICE4_IEP_CLASS_0
Offset: 0x1114

INTR_CTLR_CHANNEL2_SLICE5_IEP_CLASS_0
Offset: 0x1154

INTR_CTLR_CHANNEL2_SLICE6_IEP_CLASS_0
Offset: 0x1194

INTR_CTLR_CHANNEL2_SLICE7_IEP_CLASS_0
Offset: 0x11d4

INTR_CTLR_CHANNEL2_SLICE8_IEP_CLASS_0
Offset: 0x1214

INTR_CTLR_CHANNEL2_SLICE9_IEP_CLASS_0
Offset: 0x1254

INTR_CTLR_CHANNEL2_SLICE10_IEP_CLASS_0
Offset: 0x1294

INTR_CTLR_CHANNEL2_SLICE11_IEP_CLASS_0
Offset: 0x12d4

INTR_CTLR_CHANNEL2_SLICE12_IEP_CLASS_0
Offset: 0x1314

INTR_CTLR_CHANNEL2_SLICE13_IEP_CLASS_0
Offset: 0x1354

INTR_CTLR_CHANNEL2_SLICE14_IEP_CLASS_0
Offset: 0x1394

INTR_CTLR_CHANNEL2_SLICE15_IEP_CLASS_0
Offset: 0x13d4

INTR_CTLR_CHANNEL2_SLICE16_IEP_CLASS_0
Offset: 0x1414

INTR_CTLR_CHANNEL3_SLICE0_IEP_CLASS_0
Offset: 0x1814

INTR_CTLR_CHANNEL3_SLICE1_IEP_CLASS_0
Offset: 0x1854

INTR_CTLR_CHANNEL3_SLICE2_IEP_CLASS_0
Offset: 0x1894

INTR_CTLR_CHANNEL3_SLICE3_IEP_CLASS_0
Offset: 0x18d4

INTR_CTLR_CHANNEL3_SLICE4_IEP_CLASS_0
Offset: 0x1914

INTR_CTLR_CHANNEL3_SLICE5_IEP_CLASS_0
Offset: 0x1954

INTR_CTLR_CHANNEL3_SLICE6_IEP_CLASS_0
Offset: 0x1994

INTR_CTLR_CHANNEL3_SLICE7_IEP_CLASS_0
Offset: 0x19d4

INTR_CTLR_CHANNEL3_SLICE8_IEP_CLASS_0
Offset: 0x1a14

INTR_CTLR_CHANNEL3_SLICE9_IEP_CLASS_0
Offset: 0x1a54

INTR_CTLR_CHANNEL3_SLICE10_IEP_CLASS_0
Offset: 0x1a94

INTR_CTLR_CHANNEL3_SLICE11_IEP_CLASS_0
Offset: 0x1ad4

INTR_CTLR_CHANNEL3_SLICE12_IEP_CLASS_0
Offset: 0x1b14

INTR_CTLR_CHANNEL3_SLICE13_IEP_CLASS_0
Offset: 0x1b54

INTR_CTLR_CHANNEL3_SLICE14_IEP_CLASS_0
Offset: 0x1b94

INTR_CTLR_CHANNEL3_SLICE15_IEP_CLASS_0
Offset: 0x1bd4

INTR_CTLR_CHANNEL3_SLICE16_IEP_CLASS_0
Offset: 0x1c14

INTR_CTLR_CHANNEL4_SLICE0_IEP_CLASS_0
Offset: 0x2014

INTR_CTLR_CHANNEL4_SLICE1_IEP_CLASS_0
Offset: 0x2054

INTR_CTLR_CHANNEL4_SLICE2_IEP_CLASS_0
Offset: 0x2094

INTR_CTLR_CHANNEL4_SLICE3_IEP_CLASS_0
Offset: 0x20d4

INTR_CTLR_CHANNEL4_SLICE4_IEP_CLASS_0
Offset: 0x2114

INTR_CTLR_CHANNEL4_SLICE5_IEP_CLASS_0
Offset: 0x2154

INTR_CTLR_CHANNEL4_SLICE6_IEP_CLASS_0
Offset: 0x2194

INTR_CTLR_CHANNEL4_SLICE7_IEP_CLASS_0
Offset: 0x21d4

INTR_CTLR_CHANNEL4_SLICE8_IEP_CLASS_0
Offset: 0x2214

INTR_CTLR_CHANNEL4_SLICE9_IEP_CLASS_0
Offset: 0x2254

INTR_CTLR_CHANNEL4_SLICE10_IEP_CLASS_0
Offset: 0x2294

INTR_CTLR_CHANNEL4_SLICE11_IEP_CLASS_0
Offset: 0x22d4

INTR_CTLR_CHANNEL4_SLICE12_IEP_CLASS_0
Offset: 0x2314

INTR_CTLR_CHANNEL4_SLICE13_IEP_CLASS_0
Offset: 0x2354

INTR_CTLR_CHANNEL4_SLICE14_IEP_CLASS_0
Offset: 0x2394

INTR_CTLR_CHANNEL4_SLICE15_IEP_CLASS_0
Offset: 0x23d4

INTR_CTLR_CHANNEL4_SLICE16_IEP_CLASS_0
Offset: 0x2414

INTR_CTLR_CHANNEL5_SLICE0_IEP_CLASS_0
Offset: 0x2814

INTR_CTLR_CHANNEL5_SLICE1_IEP_CLASS_0
Offset: 0x2854

INTR_CTLR_CHANNEL5_SLICE2_IEP_CLASS_0
Offset: 0x2894

INTR_CTLR_CHANNEL5_SLICE3_IEP_CLASS_0
Offset: 0x28d4

INTR_CTLR_CHANNEL5_SLICE4_IEP_CLASS_0
Offset: 0x2914

INTR_CTLR_CHANNEL5_SLICE5_IEP_CLASS_0
Offset: 0x2954

INTR_CTLR_CHANNEL5_SLICE6_IEP_CLASS_0
Offset: 0x2994

INTR_CTLR_CHANNEL5_SLICE7_IEP_CLASS_0
Offset: 0x29d4

INTR_CTLR_CHANNEL5_SLICE8_IEP_CLASS_0
Offset: 0x2a14

INTR_CTLR_CHANNEL5_SLICE9_IEP_CLASS_0
Offset: 0x2a54

INTR_CTLR_CHANNEL5_SLICE10_IEP_CLASS_0
Offset: 0x2a94

INTR_CTLR_CHANNEL5_SLICE11_IEP_CLASS_0
Offset: 0x2ad4

INTR_CTLR_CHANNEL5_SLICE12_IEP_CLASS_0
Offset: 0x2b14

INTR_CTLR_CHANNEL5_SLICE13_IEP_CLASS_0
Offset: 0x2b54

INTR_CTLR_CHANNEL5_SLICE14_IEP_CLASS_0
Offset: 0x2b94

INTR_CTLR_CHANNEL5_SLICE15_IEP_CLASS_0
Offset: 0x2bd4

INTR_CTLR_CHANNEL5_SLICE16_IEP_CLASS_0
Offset: 0x2c14

INTR_CTLR_CHANNEL6_SLICE0_IEP_CLASS_0
Offset: 0x3014

INTR_CTLR_CHANNEL6_SLICE1_IEP_CLASS_0
Offset: 0x3054

INTR_CTLR_CHANNEL6_SLICE2_IEP_CLASS_0
Offset: 0x3094

INTR_CTLR_CHANNEL6_SLICE3_IEP_CLASS_0
Offset: 0x30d4

INTR_CTLR_CHANNEL6_SLICE4_IEP_CLASS_0
Offset: 0x3114

INTR_CTLR_CHANNEL6_SLICE5_IEP_CLASS_0
Offset: 0x3154

INTR_CTLR_CHANNEL6_SLICE6_IEP_CLASS_0
Offset: 0x3194

INTR_CTLR_CHANNEL6_SLICE7_IEP_CLASS_0
Offset: 0x31d4

INTR_CTLR_CHANNEL6_SLICE8_IEP_CLASS_0
Offset: 0x3214

INTR_CTLR_CHANNEL6_SLICE9_IEP_CLASS_0
Offset: 0x3254

INTR_CTLR_CHANNEL6_SLICE10_IEP_CLASS_0
Offset: 0x3294

INTR_CTLR_CHANNEL6_SLICE11_IEP_CLASS_0
Offset: 0x32d4

INTR_CTLR_CHANNEL6_SLICE12_IEP_CLASS_0
Offset: 0x3314

INTR_CTLR_CHANNEL6_SLICE13_IEP_CLASS_0
Offset: 0x3354

INTR_CTLR_CHANNEL6_SLICE14_IEP_CLASS_0
Offset: 0x3394

INTR_CTLR_CHANNEL6_SLICE15_IEP_CLASS_0
Offset: 0x33d4

INTR_CTLR_CHANNEL6_SLICE16_IEP_CLASS_0
Offset: 0x3414

INTR_CTLR_CHANNEL7_SLICE0_IEP_CLASS_0
Offset: 0x3814

INTR_CTLR_CHANNEL7_SLICE1_IEP_CLASS_0
Offset: 0x3854

INTR_CTLR_CHANNEL7_SLICE2_IEP_CLASS_0
Offset: 0x3894

INTR_CTLR_CHANNEL7_SLICE3_IEP_CLASS_0
Offset: 0x38d4

INTR_CTLR_CHANNEL7_SLICE4_IEP_CLASS_0
Offset: 0x3914

INTR_CTLR_CHANNEL7_SLICE5_IEP_CLASS_0
Offset: 0x3954

INTR_CTLR_CHANNEL7_SLICE6_IEP_CLASS_0
Offset: 0x3994

INTR_CTLR_CHANNEL7_SLICE7_IEP_CLASS_0
Offset: 0x39d4

INTR_CTLR_CHANNEL7_SLICE8_IEP_CLASS_0
Offset: 0x3a14

INTR_CTLR_CHANNEL7_SLICE9_IEP_CLASS_0
Offset: 0x3a54

INTR_CTLR_CHANNEL7_SLICE10_IEP_CLASS_0
Offset: 0x3a94

INTR_CTLR_CHANNEL7_SLICE11_IEP_CLASS_0
Offset: 0x3ad4

INTR_CTLR_CHANNEL7_SLICE12_IEP_CLASS_0
Offset: 0x3b14

INTR_CTLR_CHANNEL7_SLICE13_IEP_CLASS_0
Offset: 0x3b54

INTR_CTLR_CHANNEL7_SLICE14_IEP_CLASS_0
Offset: 0x3b94

INTR_CTLR_CHANNEL7_SLICE15_IEP_CLASS_0
Offset: 0x3bd4

INTR_CTLR_CHANNEL7_SLICE16_IEP_CLASS_0
Offset: 0x3c14

INTR_CTLR_CHANNEL8_SLICE0_IEP_CLASS_0
Offset: 0x4014

INTR_CTLR_CHANNEL8_SLICE1_IEP_CLASS_0
Offset: 0x4054

INTR_CTLR_CHANNEL8_SLICE2_IEP_CLASS_0
Offset: 0x4094

INTR_CTLR_CHANNEL8_SLICE3_IEP_CLASS_0
Offset: 0x40d4

INTR_CTLR_CHANNEL8_SLICE4_IEP_CLASS_0
Offset: 0x4114

INTR_CTLR_CHANNEL8_SLICE5_IEP_CLASS_0
Offset: 0x4154

INTR_CTLR_CHANNEL8_SLICE6_IEP_CLASS_0
Offset: 0x4194

INTR_CTLR_CHANNEL8_SLICE7_IEP_CLASS_0
Offset: 0x41d4

INTR_CTLR_CHANNEL8_SLICE8_IEP_CLASS_0
Offset: 0x4214

INTR_CTLRCHANNEL8_SLICE9_IEP_CLASS_0
Offset: 0x4254

INTR_CTLR_CHANNEL8_SLICE10_IEP_CLASS_0
Offset: 0x4294

INTR_CTLR_CHANNEL8_SLICE11_IEP_CLASS_0
Offset: 0x42d4

INTR_CTLR_CHANNEL8_SLICE12_IEP_CLASS_0
Offset: 0x4314

INTR_CTLR_CHANNEL8_SLICE13_IEP_CLASS_0
Offset: 0x4354

INTR_CTLR_CHANNEL8_SLICE14_IEP_CLASS_0
Offset: 0x4394

INTR_CTLR_CHANNEL8_SLICE15_IEP_CLASS_0
Offset: 0x43d4

INTR_CTLR_CHANNEL8_SLICE16_IEP_CLASS_0
Offset: 0x4414

INTR_CTLR_CHANNEL9_SLICE0_IEP_CLASS_0
Offset: 0x4814

INTR_CTLR_CHANNEL9_SLICE1_IEP_CLASS_0
Offset: 0x4854

INTR_CTLR_CHANNEL9_SLICE2_IEP_CLASS_0
Offset: 0x4894

INTR_CTLR_CHANNEL9_SLICE3_IEP_CLASS_0
Offset: 0x48d4

INTR_CTLR_CHANNEL9_SLICE4_IEP_CLASS_0
Offset: 0x4914

INTR_CTLR_CHANNEL9_SLICE5_IEP_CLASS_0
Offset: 0x4954

INTR_CTLR_CHANNEL9_SLICE6_IEP_CLASS_0
Offset: 0x4994

INTR_CTLR_CHANNEL9_SLICE7_IEP_CLASS_0
Offset: 0x49d4

INTR_CTLR_CHANNEL9_SLICE8_IEP_CLASS_0
Offset: 0x4a14

INTR_CTLR_CHANNEL9_SLICE9_IEP_CLASS_0
Offset: 0x4a54

INTR_CTLR_CHANNEL9_SLICE10_IEP_CLASS_0
Offset: 0x4a94

INTR_CTLR_CHANNEL9_SLICE11_IEP_CLASS_0
Offset: 0x4ad4

INTR_CTLR_CHANNEL9_SLICE12_IEP_CLASS_0
Offset: 0x4b14

INTR_CTLR_CHANNEL9_SLICE13_IEP_CLASS_0
Offset: 0x4b54

INTR_CTLR_CHANNEL9_SLICE14_IEP_CLASS_0
Offset: 0x4b94

INTR_CTLR_CHANNEL9_SLICE15_IEP_CLASS_0
Offset: 0x4bd4

INTR_CTLR_CHANNEL9_SLICE16_IEP_CLASS_0
Offset: 0x4c14

INTR_CTLR_CHANNEL10_SLICE0_IEP_CLASS_0
Offset: 0x5014

INTR_CTLR_CHANNEL10_SLICE1_IEP_CLASS_0
Offset: 0x5054

INTR_CTLR_CHANNEL10_SLICE2_IEP_CLASS_0
Offset: 0x5094

INTR_CTLR_CHANNEL10_SLICE3_IEP_CLASS_0
Offset: 0x50d4

INTR_CTLR_CHANNEL10_SLICE4_IEP_CLASS_0
Offset: 0x5114

INTR_CTLR_CHANNEL10_SLICE5_IEP_CLASS_0
Offset: 0x5154

INTR_CTLR_CHANNEL10_SLICE6_IEP_CLASS_0
Offset: 0x5194

INTR_CTLR_CHANNEL10_SLICE7_IEP_CLASS_0
Offset: 0x51d4

INTR_CTLR_CHANNEL10_SLICE8_IEP_CLASS_0
Offset: 0x5214

INTR_CTLR_CHANNEL10_SLICE9_IEP_CLASS_0
Offset: 0x5254

INTR_CTLR_CHANNEL10_SLICE10_IEP_CLASS_0
Offset: 0x5294

INTR_CTLR_CHANNEL10_SLICE11_IEP_CLASS_0
Offset: 0x52d4

INTR_CTLR_CHANNEL10_SLICE12_IEP_CLASS_0
Offset: 0x5314

INTR_CTLR_CHANNEL10_SLICE13_IEP_CLASS_0
Offset: 0x5354

INTR_CTLR_CHANNEL10_SLICE14_IEP_CLASS_0
Offset: 0x5394

INTR_CTLR_CHANNEL10_SLICE15_IEP_CLASS_0
Offset: 0x53d4

INTR_CTLR_CHANNEL10_SLICE16_IEP_CLASS_0
Offset: 0x5414

INTR_CTLR_CHANNEL11_SLICE0_IEP_CLASS_0
Offset: 0x5814

INTR_CTLR_CHANNEL11_SLICE1_IEP_CLASS_0
Offset: 0x5854

INTR_CTLR_CHANNEL11_SLICE2_IEP_CLASS_0
Offset: 0x5894

INTR_CTLR_CHANNEL11_SLICE3_IEP_CLASS_0
Offset: 0x58d4

INTR_CTLR_CHANNEL11_SLICE4_IEP_CLASS_0
Offset: 0x5914

INTR_CTLR_CHANNEL11_SLICE5_IEP_CLASS_0
Offset: 0x5954

INTR_CTLR_CHANNEL11_SLICE6_IEP_CLASS_0
Offset: 0x5994

INTR_CTLR_CHANNEL11_SLICE7_IEP_CLASS_0
Offset: 0x59d4

INTR_CTLR_CHANNEL11_SLICE8_IEP_CLASS_0
Offset: 0x5a14

INTR_CTLR_CHANNEL11_SLICE9_IEP_CLASS_0
Offset: 0x5a54

INTR_CTLR_CHANNEL11_SLICE10_IEP_CLASS_0
Offset: 0x5a94

INTR_CTLR_CHANNEL11_SLICE11_IEP_CLASS_0
Offset: 0x5ad4

INTR_CTLR_CHANNEL11_SLICE12_IEP_CLASS_0
Offset: 0x5b14

INTR_CTLR_CHANNEL11_SLICE13_IEP_CLASS_0
Offset: 0x5b54

INTR_CTLR_CHANNEL11_SLICE14_IEP_CLASS_0
Offset: 0x5b94

INTR_CTLR_CHANNEL11_SLICE15_IEP_CLASS_0
Offset: 0x5bd4

INTR_CTLR_CHANNEL11_SLICE16_IEP_CLASS_0
Offset: 0x5c14

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: INTR_CTLR_CHANNEL<*i*>_SCR_CHANNEL<*i*>_SCRE_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**IEP_CLASS** Each bit for the corresponding Interrupt. 0: IRQ 1: FIQ

INTR_CTLR_CHANNEL<*i*>_SLICE<*j*>_ISR_0,

where <*i*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, and

<*j*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16.

This register holds the current value of input to the specific Channel/Slice after IDR.

INTR_CTLR_CHANNEL0_SLICE0_ISR_0
Offset: 0x18

INTR_CTLR_CHANNEL0_SLICE1_ISR_0
Offset: 0x58

INTR_CTLR_CHANNEL0_SLICE2_ISR_0
Offset: 0x98

INTR_CTLR_CHANNEL0_SLICE3_ISR_0
Offset: 0xd8

INTR_CTLR_CHANNEL0_SLICE4_ISR_0
Offset: 0x118

INTR_CTLR_CHANNEL0_SLICE5_ISR_0
Offset: 0x158

INTR_CTLR_CHANNEL0_SLICE6_ISR_0
Offset: 0x198

INTR_CTLR_CHANNEL0_SLICE7_ISR_0
Offset: 0x1d8

INTR_CTLR_CHANNEL0_SLICE8_ISR_0
Offset: 0x218

INTR_CTLR_CHANNEL0_SLICE9_ISR_0
Offset: 0x258

INTR_CTLR_CHANNEL0_SLICE10_ISR_0
Offset: 0x298

INTR_CTLR_CHANNEL0_SLICE11_ISR_0
Offset: 0x2d8

INTR_CTLR_CHANNEL0_SLICE12_ISR_0
Offset: 0x318

INTR_CTLR_CHANNEL0_SLICE13_ISR_0
Offset: 0x358

INTR_CTLR_CHANNEL0_SLICE14_ISR_0
Offset: 0x398

INTR_CTLR_CHANNEL0_SLICE15_ISR_0
Offset: 0x3d8

INTR_CTLR_CHANNEL0_SLICE16_ISR_0
Offset: 0x418

INTR_CTLR_CHANNEL1_SLICE0_ISR_0
Offset: 0x818

INTR_CTLR_CHANNEL1_SLICE1_ISR_0
Offset: 0x858

INTR_CTLR_CHANNEL1_SLICE2_ISR_0
Offset: 0x898

INTR_CTLR_CHANNEL1_SLICE3_ISR_0
Offset: 0x8d8

INTR_CTLR_CHANNEL1_SLICE4_ISR_0
Offset: 0x918

INTR_CTLR_CHANNEL1_SLICE5_ISR_0
Offset: 0x958

INTR_CTLR_CHANNEL1_SLICE6_ISR_0
Offset: 0x998

INTR_CTLR_CHANNEL1_SLICE7_ISR_0
Offset: 0x9d8

INTR_CTLR_CHANNEL1_SLICE8_ISR_0
Offset: 0xa18

INTR_CTLR_CHANNEL1_SLICE9_ISR_0
Offset: 0xa58

INTR_CTLR_CHANNEL1_SLICE10_ISR_0
Offset: 0xa98

INTR_CTLR_CHANNEL1_SLICE11_ISR_0
Offset: 0xad8

INTR_CTLR_CHANNEL1_SLICE12_ISR_0
Offset: 0xb18

INTR_CTLR_CHANNEL1_SLICE13_ISR_0
Offset: 0xb58

INTR_CTLR_CHANNEL1_SLICE14_ISR_0
Offset: 0xb98

INTR_CTLR_CHANNEL1_SLICE15_ISR_0
Offset: 0xbd8

INTR_CTLR_CHANNEL1_SLICE16_ISR_0
Offset: 0xc18

INTR_CTLR_CHANNEL2_SLICE0_ISR_0
Offset: 0x1018

INTR_CTLR_CHANNEL2_SLICE1_ISR_0
Offset: 0x1058

INTR_CTLR_CHANNEL2_SLICE2_ISR_0
Offset: 0x1098

INTR_CTLR_CHANNEL2_SLICE3_ISR_0
Offset: 0x10d8

INTR_CTLR_CHANNEL2_SLICE4_ISR_0
Offset: 0x1118

INTR_CTLR_CHANNEL2_SLICE5_ISR_0
Offset: 0x1158

INTR_CTLR_CHANNEL2_SLICE6_ISR_0
Offset: 0x1198

INTR_CTLR_CHANNEL2_SLICE7_ISR_0
Offset: 0x11d8

INTR_CTLR_CHANNEL2_SLICE8_ISR_0
Offset: 0x1218

INTR_CTLR_CHANNEL2_SLICE9_ISR_0
Offset: 0x1258

INTR_CTLR_CHANNEL2_SLICE10_ISR_0
Offset: 0x1298

INTR_CTLR_CHANNEL2_SLICE11_ISR_0
Offset: 0x12d8

INTR_CTLR_CHANNEL2_SLICE12_ISR_0
Offset: 0x1318

INTR_CTLR_CHANNEL2_SLICE13_ISR_0
Offset: 0x1358

INTR_CTLR_CHANNEL2_SLICE14_ISR_0
Offset: 0x1398

INTR_CTLR_CHANNEL2_SLICE15_ISR_0
Offset: 0x13d8

INTR_CTLR_CHANNEL2_SLICE16_ISR_0
Offset: 0x1418

INTR_CTLR_CHANNEL3_SLICE0_ISR_0
Offset: 0x1818

INTR_CTLR_CHANNEL3_SLICE1_ISR_0
Offset: 0x1858

INTR_CTLR_CHANNEL3_SLICE2_ISR_0
Offset: 0x1898

INTR_CTLR_CHANNEL3_SLICE3_ISR_0
Offset: 0x18d8

INTR_CTLR_CHANNEL3_SLICE4_ISR_0
Offset: 0x1918

INTR_CTLR_CHANNEL3_SLICE5_ISR_0
Offset: 0x1958

INTR_CTLR_CHANNEL3_SLICE6_ISR_0
Offset: 0x1998

INTR_CTLR_CHANNEL3_SLICE7_ISR_0
Offset: 0x19d8

INTR_CTLR_CHANNEL3_SLICE8_ISR_0
Offset: 0x1a18

INTR_CTLR_CHANNEL3_SLICE9_ISR_0
Offset: 0x1a58

INTR_CTLR_CHANNEL3_SLICE10_ISR_0
Offset: 0x1a98

INTR_CTLR_CHANNEL3_SLICE11_ISR_0
Offset: 0x1ad8

INTR_CTLR_CHANNEL3_SLICE12_ISR_0
Offset: 0x1b18

INTR_CTLR_CHANNEL3_SLICE13_ISR_0
Offset: 0x1b58

INTR_CTLR_CHANNEL3_SLICE14_ISR_0
Offset: 0x1b98

INTR_CTLR_CHANNEL3_SLICE15_ISR_0
Offset: 0x1bd8

INTR_CTLR_CHANNEL3_SLICE16_ISR_0
Offset: 0x1c18

INTR_CTLR_CHANNEL4_SLICE0_ISR_0
Offset: 0x2018

INTR_CTLR_CHANNEL4_SLICE1_ISR_0
Offset: 0x2058

INTR_CTLR_CHANNEL4_SLICE2_ISR_0
Offset: 0x2098

INTR_CTLR_CHANNEL4_SLICE3_ISR_0
Offset: 0x20d8

INTR_CTLR_CHANNEL4_SLICE4_ISR_0
Offset: 0x2118

INTR_CTLR_CHANNEL4_SLICE5_ISR_0
Offset: 0x2158

INTR_CTLR_CHANNEL4_SLICE6_ISR_0
Offset: 0x2198

INTR_CTLR_CHANNEL4_SLICE7_ISR_0
Offset: 0x21d8

INTR_CTLR_CHANNEL4_SLICE8_ISR_0
Offset: 0x2218

INTR_CTLR_CHANNEL4_SLICE9_ISR_0
Offset: 0x2258

INTR_CTLR_CHANNEL4_SLICE10_ISR_0
Offset: 0x2298

INTR_CTLR_CHANNEL4_SLICE11_ISR_0
Offset: 0x22d8

INTR_CTLR_CHANNEL4_SLICE12_ISR_0
Offset: 0x2318

INTR_CTLR_CHANNEL4_SLICE13_ISR_0
Offset: 0x2358

INTR_CTLR_CHANNEL4_SLICE14_ISR_0
Offset: 0x2398

INTR_CTLR_CHANNEL4_SLICE15_ISR_0
Offset: 0x23d8

INTR_CTLR_CHANNEL4_SLICE16_ISR_0
Offset: 0x2418

INTR_CTLR_CHANNEL5_SLICE0_ISR_0
Offset: 0x2818

INTR_CTLR_CHANNEL5_SLICE1_ISR_0
Offset: 0x2858

INTR_CTLR_CHANNEL5_SLICE2_ISR_0
Offset: 0x2898

INTR_CTLR_CHANNEL5_SLICE3_ISR_0
Offset: 0x28d8

INTR_CTLR_CHANNEL5_SLICE4_ISR_0
Offset: 0x2918

INTR_CTLR_CHANNEL5_SLICE5_ISR_0
Offset: 0x2958

INTR_CTLR_CHANNEL5_SLICE6_ISR_0
Offset: 0x2998

INTR_CTLR_CHANNEL5_SLICE7_ISR_0
Offset: 0x29d8

INTR_CTLR_CHANNEL5_SLICE8_ISR_0
Offset: 0x2a18

INTR_CTLR_CHANNEL5_SLICE9_ISR_0
Offset: 0x2a58

INTR_CTLR_CHANNEL5_SLICE10_ISR_0
Offset: 0x2a98

INTR_CTLR_CHANNEL5_SLICE11_ISR_0
Offset: 0x2ad8

INTR_CTLR_CHANNEL5_SLICE12_ISR_0
Offset: 0x2b18

INTR_CTLR_CHANNEL5_SLICE13_ISR_0
Offset: 0x2b58

INTR_CTLR_CHANNEL5_SLICE14_ISR_0
Offset: 0x2b98

INTR_CTLR_CHANNEL5_SLICE15_ISR_0
Offset: 0x2bd8

INTR_CTLR_CHANNEL5_SLICE16_ISR_0
Offset: 0x2c18

INTR_CTLR_CHANNEL6_SLICE0_ISR_0
Offset: 0x3018

INTR_CTLR_CHANNEL6_SLICE1_ISR_0
Offset: 0x3058

INTR_CTLR_CHANNEL6_SLICE2_ISR_0
Offset: 0x3098

INTR_CTLR_CHANNEL6_SLICE3_ISR_0
Offset: 0x30d8

INTR_CTLR_CHANNEL6_SLICE4_ISR_0
Offset: 0x3118

INTR_CTLR_CHANNEL6_SLICE5_ISR_0
Offset: 0x3158

INTR_CTLR_CHANNEL6_SLICE6_ISR_0
Offset: 0x3198

INTR_CTLR_CHANNEL6_SLICE7_ISR_0
Offset: 0x31d8

INTR_CTLR_CHANNEL6_SLICE8_ISR_0
Offset: 0x3218

INTR_CTLR_CHANNEL6_SLICE9_ISR_0
Offset: 0x3258

INTR_CTLR_CHANNEL6_SLICE10_ISR_0
Offset: 0x3298

INTR_CTLR_CHANNEL6_SLICE11_ISR_0
Offset: 0x32d8

INTR_CTLR_CHANNEL6_SLICE12_ISR_0
Offset: 0x3318

INTR_CTLR_CHANNEL6_SLICE13_ISR_0
Offset: 0x3358

INTR_CTLR_CHANNEL6_SLICE14_ISR_0
Offset: 0x3398

INTR_CTLR_CHANNEL6_SLICE15_ISR_0
Offset: 0x33d8

INTR_CTLR_CHANNEL6_SLICE16_ISR_0
Offset: 0x3418

INTR_CTLR_CHANNEL7_SLICE0_ISR_0
Offset: 0x3818

INTR_CTLR_CHANNEL7_SLICE1_ISR_0
Offset: 0x3858

INTR_CTLR_CHANNEL7_SLICE2_ISR_0
Offset: 0x3898

INTR_CTLR_CHANNEL7_SLICE3_ISR_0
Offset: 0x38d8

INTR_CTLR_CHANNEL7_SLICE4_ISR_0
Offset: 0x3918

INTR_CTLR_CHANNEL7_SLICE5_ISR_0
Offset: 0x3958

INTR_CTLR_CHANNEL7_SLICE6_ISR_0
Offset: 0x3998

INTR_CTLR_CHANNEL7_SLICE7_ISR_0
Offset: 0x39d8

INTR_CTLR_CHANNEL7_SLICE8_ISR_0
Offset: 0x3a18

INTR_CTLR_CHANNEL7_SLICE9_ISR_0
Offset: 0x3a58

INTR_CTLR_CHANNEL7_SLICE10_ISR_0
Offset: 0x3a98

INTR_CTLR_CHANNEL7_SLICE11_ISR_0
Offset: 0x3ad8

INTR_CTLR_CHANNEL7_SLICE12_ISR_0
Offset: 0x3b18

INTR_CTLR_CHANNEL7_SLICE13_ISR_0
Offset: 0x3b58

INTR_CTLR_CHANNEL7_SLICE14_ISR_0
Offset: 0x3b98

INTR_CTLR_CHANNEL7_SLICE15_ISR_0
Offset: 0x3bd8

INTR_CTLR_CHANNEL7_SLICE16_ISR_0
Offset: 0x3c18

INTR_CTLR_CHANNEL8_SLICE0_ISR_0
Offset: 0x4018

INTR_CTLR_CHANNEL8_SLICE1_ISR_0
Offset: 0x4058

INTR_CTLR_CHANNEL8_SLICE2_ISR_0
Offset: 0x4098

INTR_CTLR_CHANNEL8_SLICE3_ISR_0
Offset: 0x40d8

INTR_CTLR_CHANNEL8_SLICE4_ISR_0
Offset: 0x4118

INTR_CTLR_CHANNEL8_SLICE5_ISR_0
Offset: 0x4158

INTR_CTLR_CHANNEL8_SLICE6_ISR_0
Offset: 0x4198

INTR_CTLR_CHANNEL8_SLICE7_ISR_0
Offset: 0x41d8

INTR_CTLR_CHANNEL8_SLICE8_ISR_0
Offset: 0x4218

INTR_CTLR_CHANNEL8_SLICE9_ISR_0
Offset: 0x4258

INTR_CTLR_CHANNEL8_SLICE10_ISR_0
Offset: 0x4298

INTR_CTLR_CHANNEL8_SLICE11_ISR_0
Offset: 0x42d8

INTR_CTLR_CHANNEL8_SLICE12_ISR_0
Offset: 0x4318

INTR_CTLR_CHANNEL8_SLICE13_ISR_0
Offset: 0x4358

INTR_CTLR_CHANNEL8_SLICE14_ISR_0
Offset: 0x4398

INTR_CTLR_CHANNEL8_SLICE15_ISR_0
Offset: 0x43d8

INTR_CTLR_CHANNEL8_SLICE16_ISR_0
Offset: 0x4418

INTR_CTLR_CHANNEL9_SLICE0_ISR_0
Offset: 0x4818

INTR_CTLR_CHANNEL9_SLICE1_ISR_0
Offset: 0x4858

INTR_CTLR_CHANNEL9_SLICE2_ISR_0
Offset: 0x4898

INTR_CTLR_CHANNEL9_SLICE3_ISR_0
Offset: 0x48d8

INTR_CTLR_CHANNEL9_SLICE4_ISR_0
Offset: 0x4918

INTR_CTLR_CHANNEL9_SLICE5_ISR_0
Offset: 0x4958

INTR_CTLR_CHANNEL9_SLICE6_ISR_0
Offset: 0x4998

INTR_CTLR_CHANNEL9_SLICE7_ISR_0
Offset: 0x49d8

INTR_CTLR_CHANNEL9_SLICE8_ISR_0
Offset: 0x4a18

INTR_CTLR_CHANNEL9_SLICE9_ISR_0
Offset: 0x4a58

INTR_CTLR_CHANNEL9_SLICE10_ISR_0
Offset: 0x4a98

INTR_CTLR_CHANNEL9_SLICE11_ISR_0
Offset: 0x4ad8

INTR_CTLR_CHANNEL9_SLICE12_ISR_0
Offset: 0x4b18

INTR_CTLR_CHANNEL9_SLICE13_ISR_0
Offset: 0x4b58

INTR_CTLR_CHANNEL9_SLICE14_ISR_0
Offset: 0x4b98

INTR_CTLR_CHANNEL9_SLICE15_ISR_0
Offset: 0x4bd8

INTR_CTLR_CHANNEL9_SLICE16_ISR_0
Offset: 0x4c18

INTR_CTLR_CHANNEL10_SLICE0_ISR_0
Offset: 0x5018

INTR_CTLR_CHANNEL10_SLICE1_ISR_0
Offset: 0x5058

INTR_CTLR_CHANNEL10_SLICE2_ISR_0
Offset: 0x5098

INTR_CTLR_CHANNEL10_SLICE3_ISR_0
Offset: 0x50d8

INTR_CTLR_CHANNEL10_SLICE4_ISR_0
Offset: 0x5118

INTR_CTLR_CHANNEL10_SLICE5_ISR_0
Offset: 0x5158

INTR_CTLR_CHANNEL10_SLICE6_ISR_0
Offset: 0x5198

INTR_CTLR_CHANNEL10_SLICE7_ISR_0
Offset: 0x51d8

INTR_CTLR_CHANNEL10_SLICE8_ISR_0
Offset: 0x5218

INTR_CTLR_CHANNEL10_SLICE9_ISR_0
Offset: 0x5258

INTR_CTLR_CHANNEL10_SLICE10_ISR_0
Offset: 0x5298

INTR_CTLR_CHANNEL10_SLICE11_ISR_0
Offset: 0x52d8

INTR_CTLR_CHANNEL10_SLICE12_ISR_0
Offset: 0x5318

INTR_CTLR_CHANNEL10_SLICE13_ISR_0
Offset: 0x5358

INTR_CTLR_CHANNEL10_SLICE14_ISR_0
Offset: 0x5398

INTR_CTLR_CHANNEL10_SLICE15_ISR_0
Offset: 0x53d8

INTR_CTLR_CHANNEL10_SLICE16_ISR_0
Offset: 0x5418

INTR_CTLR_CHANNEL11_SLICE0_ISR_0
Offset: 0x5818

INTR_CTLR_CHANNEL11_SLICE1_ISR_0
Offset: 0x5858

INTR_CTLR_CHANNEL11_SLICE2_ISR_0
Offset: 0x5898

INTR_CTLR_CHANNEL11_SLICE3_ISR_0
Offset: 0x58d

INTR_CTLR_CHANNEL11_SLICE4_ISR_0
Offset: 0x5918

INTR_CTLR_CHANNEL11_SLICE5_ISR_0
Offset: 0x5958

INTR_CTLR_CHANNEL11_SLICE6_ISR_0
Offset: 0x5998

INTR_CTLR_CHANNEL11_SLICE7_ISR_0
Offset: 0x59d8

INTR_CTLR_CHANNEL11_SLICE8_ISR_0
Offset: 0x5a18

INTR_CTLR_CHANNEL11_SLICE9_ISR_0
Offset: 0x5a58

INTR_CTLR_CHANNEL11_SLICE10_ISR_0
Offset: 0x5a98

INTR_CTLR_CHANNEL11_SLICE11_ISR_0
Offset: 0x5ad8

INTR_CTLR_CHANNEL11_SLICE12_ISR_0
Offset: 0x5b18

INTR_CTLR_CHANNEL11_SLICE13_ISR_0
Offset: 0x5b58

INTR_CTLR_CHANNEL11_SLICE14_ISR_0
Offset: 0x5b98

INTR_CTLR_CHANNEL11_SLICE15_ISR_0
Offset: 0x5bd8

INTR_CTLR_CHANNEL11_SLICE16_ISR_0
Offset: 0x5c18

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: INTR_CTLR_CHANNEL<*i*>_SCR_CHANNEL<*i*>_SCRE_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ISR**

INTR_CTLR_CHANNEL<*i*>_SLICE<*j*>_IDR_0,

where <*i*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, and

<*j*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16.

Each bit of this Interrupt Disable Register (IDR) indicates whether the corresponding Interrupt in thespecific Channel/Slice is disabled.

INTR_CTLR_CHANNEL0_SLICE0_IDR_0
Offset: 0x1c

INTR_CTLR_CHANNEL0_SLICE1_IDR_0
Offset: 0x5c

INTR_CTLR_CHANNEL0_SLICE2_IDR_0
Offset: 0x9c

INTR_CTLR_CHANNEL0_SLICE3_IDR_0
Offset: 0xdc

INTR_CTLR_CHANNEL0_SLICE4_IDR_0
Offset: 0x11c

INTR_CTLR_CHANNEL0_SLICE5_IDR_0
Offset: 0x15c

INTR_CTLR_CHANNEL0_SLICE6_IDR_0
Offset: 0x19c

INTR_CTLR_CHANNEL0_SLICE7_IDR_0
Offset: 0x1dc

INTR_CTLR_CHANNEL0_SLICE8_IDR_0
Offset: 0x21c

INTR_CTLR_CHANNEL0_SLICE9_IDR_0
Offset: 0x25c

INTR_CTLR_CHANNEL0_SLICE10_IDR_0
Offset: 0x29c

INTR_CTLR_CHANNEL0_SLICE11_IDR_0
Offset: 0x2dc

INTR_CTLR_CHANNEL0_SLICE12_IDR_0
Offset: 0x31c

INTR_CTLR_CHANNEL0_SLICE13_IDR_0
Offset: 0x35c

INTR_CTLR_CHANNEL0_SLICE14_IDR_0
Offset: 0x39c

INTR_CTLR_CHANNEL0_SLICE15_IDR_0
Offset: 0x3dc

INTR_CTLR_CHANNEL0_SLICE16_IDR_0
Offset: 0x41c

INTR_CTLR_CHANNEL1_SLICE0_IDR_0
Offset: 0x81c

INTR_CTLR_CHANNEL1_SLICE1_IDR_0
Offset: 0x85c

INTR_CTLR_CHANNEL1_SLICE2_IDR_0
Offset: 0x89c

INTR_CTLR_CHANNEL1_SLICE3_IDR_0
Offset: 0x8dc

INTR_CTLR_CHANNEL1_SLICE4_IDR_0
Offset: 0x91c

INTR_CTLR_CHANNEL1_SLICE5_IDR_0
Offset: 0x95c

INTR_CTLR_CHANNEL1_SLICE6_IDR_0
Offset: 0x99c

INTR_CTLR_CHANNEL1_SLICE7_IDR_0
Offset: 0x9dc

INTR_CTLR_CHANNEL1_SLICE8_IDR_0
Offset: 0xa1c

INTR_CTLR_CHANNEL1_SLICE9_IDR_0
Offset: 0xa5c

INTR_CTLR_CHANNEL1_SLICE10_IDR_0
Offset: 0xa9c

INTR_CTLR_CHANNEL1_SLICE11_IDR_0
Offset: 0xadc

INTR_CTLR_CHANNEL1_SLICE12_IDR_0
Offset: 0xb1c

INTR_CTLR_CHANNEL1_SLICE13_IDR_0
Offset: 0xb5c

INTR_CTLR_CHANNEL1_SLICE14_IDR_0
Offset: 0xb9c

INTR_CTLR_CHANNEL1_SLICE15_IDR_0
Offset: 0xbdc

INTR_CTLR_CHANNEL1_SLICE16_IDR_0
Offset: 0xc1c

INTR_CTLR_CHANNEL2_SLICE0_IDR_0
Offset: 0x101c

INTR_CTLR_CHANNEL2_SLICE1_IDR_0
Offset: 0x105c

INTR_CTLR_CHANNEL2_SLICE2_IDR_0
Offset: 0x109c

INTR_CTLR_CHANNEL2_SLICE3_IDR_0
Offset: 0x10dc

INTR_CTLR_CHANNEL2_SLICE4_IDR_0
Offset: 0x111c

INTR_CTLR_CHANNEL2_SLICE5_IDR_0
Offset: 0x115c

INTR_CTLR_CHANNEL2_SLICE6_IDR_0
Offset: 0x119c

INTR_CTLR_CHANNEL2_SLICE7_IDR_0
Offset: 0x11dc

INTR_CTLR_CHANNEL2_SLICE8_IDR_0
Offset: 0x121c

INTR_CTLR_CHANNEL2_SLICE9_IDR_0
Offset: 0x125c

INTR_CTLR_CHANNEL2_SLICE10_IDR_0
Offset: 0x129c

INTR_CTLR_CHANNEL2_SLICE11_IDR_0
Offset: 0x12dc

INTR_CTLR_CHANNEL2_SLICE12_IDR_0
Offset: 0x131c

INTR_CTLR_CHANNEL2_SLICE13_IDR_0
Offset: 0x135c

INTR_CTLR_CHANNEL2_SLICE14_IDR_0
Offset: 0x139c

INTR_CTLR_CHANNEL2_SLICE15_IDR_0
Offset: 0x13dc

INTR_CTLR_CHANNEL2_SLICE16_IDR_0
Offset: 0x141c

INTR_CTLR_CHANNEL3_SLICE0_IDR_0
Offset: 0x181c

INTR_CTLR_CHANNEL3_SLICE1_IDR_0
Offset: 0x185c

INTR_CTLR_CHANNEL3_SLICE2_IDR_0
Offset: 0x189c

INTR_CTLR_CHANNEL3_SLICE3_IDR_0
Offset: 0x18dc

INTR_CTLR_CHANNEL3_SLICE4_IDR_0
Offset: 0x191c

INTR_CTLR_CHANNEL3_SLICE5_IDR_0
Offset: 0x195c

INTR_CTLR_CHANNEL3_SLICE6_IDR_0
Offset: 0x199c

INTR_CTLR_CHANNEL3_SLICE7_IDR_0
Offset: 0x19dc

INTR_CTLR_CHANNEL3_SLICE8_IDR_0
Offset: 0x1a1c

INTR_CTLR_CHANNEL3_SLICE9_IDR_0
Offset: 0x1a5c

INTR_CTLR_CHANNEL3_SLICE10_IDR_0
Offset: 0x1a9c

INTR_CTLR_CHANNEL3_SLICE11_IDR_0
Offset: 0x1adc

INTR_CTLR_CHANNEL3_SLICE12_IDR_0
Offset: 0x1b1c

INTR_CTLR_CHANNEL3_SLICE13_IDR_0
Offset: 0x1b5c

INTR_CTLR_CHANNEL3_SLICE14_IDR_0
Offset: 0x1b9c

INTR_CTLR_CHANNEL3_SLICE15_IDR_0
Offset: 0x1bdc

INTR_CTLR_CHANNEL3_SLICE16_IDR_0
Offset: 0x1c1c

INTR_CTLR_CHANNEL4_SLICE0_IDR_0
Offset: 0x201c

INTR_CTLR_CHANNEL4_SLICE1_IDR_0
Offset: 0x205c

INTR_CTLR_CHANNEL4_SLICE2_IDR_0
Offset: 0x209c

INTR_CTLR_CHANNEL4_SLICE3_IDR_0
Offset: 0x20dc

INTR_CTLR_CHANNEL4_SLICE4_IDR_0
Offset: 0x211c

INTR_CTLR_CHANNEL4_SLICE5_IDR_0
Offset: 0x215c

INTR_CTLR_CHANNEL4_SLICE6_IDR_0
Offset: 0x219c

INTR_CTLR_CHANNEL4_SLICE7_IDR_0
Offset: 0x21dc

INTR_CTLR_CHANNEL4_SLICE8_IDR_0
Offset: 0x221c

INTR_CTLR_CHANNEL4_SLICE9_IDR_0
Offset: 0x225c

INTR_CTLR_CHANNEL4_SLICE10_IDR_0
Offset: 0x229c

INTR_CTLR_CHANNEL4_SLICE11_IDR_0
Offset: 0x22dc

INTR_CTLR_CHANNEL4_SLICE12_IDR_0
Offset: 0x231c

INTR_CTLR_CHANNEL4_SLICE13_IDR_0
Offset: 0x235c

INTR_CTLR_CHANNEL4_SLICE14_IDR_0
Offset: 0x239c

INTR_CTLR_CHANNEL4_SLICE15_IDR_0
Offset: 0x23dc

INTR_CTLR_CHANNEL4_SLICE16_IDR_0
Offset: 0x241c

INTR_CTLR_CHANNEL5_SLICE0_IDR_0
Offset: 0x281c

INTR_CTLR_CHANNEL5_SLICE1_IDR_0
Offset: 0x285c

INTR_CTLR_CHANNEL5_SLICE2_IDR_0
Offset: 0x289c

INTR_CTLR_CHANNEL5_SLICE3_IDR_0
Offset: 0x28dc

INTR_CTLR_CHANNEL5_SLICE4_IDR_0
Offset: 0x291c

INTR_CTLR_CHANNEL5_SLICE5_IDR_0
Offset: 0x295c

INTR_CTLR_CHANNEL5_SLICE6_IDR_0
Offset: 0x299c

INTR_CTLR_CHANNEL5_SLICE7_IDR_0
Offset: 0x29dc

INTR_CTLR_CHANNEL5_SLICE8_IDR_0
Offset: 0x2a1c

INTR_CTLR_CHANNEL5_SLICE9_IDR_0
Offset: 0x2a5c

INTR_CTLR_CHANNEL5_SLICE10_IDR_0
Offset: 0x2a9c

INTR_CTLR_CHANNEL5_SLICE11_IDR_0
Offset: 0x2adc

INTR_CTLR_CHANNEL5_SLICE12_IDR_0
Offset: 0x2b1c

INTR_CTLR_CHANNEL5_SLICE13_IDR_0
Offset: 0x2b5c

INTR_CTLR_CHANNEL5_SLICE14_IDR_0
Offset: 0x2b9c

INTR_CTLR_CHANNEL5_SLICE15_IDR_0
Offset: 0x2bdc

INTR_CTLR_CHANNEL5_SLICE16_IDR_0
Offset: 0x2c1c

INTR_CTLR_CHANNEL6_SLICE0_IDR_0
Offset: 0x301c

INTR_CTLR_CHANNEL6_SLICE1_IDR_0
Offset: 0x305c

INTR_CTLR_CHANNEL6_SLICE2_IDR_0
Offset: 0x309c

INTR_CTLR_CHANNEL6_SLICE3_IDR_0
Offset: 0x30dc

INTR_CTLR_CHANNEL6_SLICE4_IDR_0
Offset: 0x311c

INTR_CTLR_CHANNEL6_SLICE5_IDR_0
Offset: 0x315c

INTR_CTLR_CHANNEL6_SLICE6_IDR_0
Offset: 0x319c

INTR_CTLR_CHANNEL6_SLICE7_IDR_0
Offset: 0x31dc

INTR_CTLR_CHANNEL6_SLICE8_IDR_0
Offset: 0x321c

INTR_CTLR_CHANNEL6_SLICE9_IDR_0
Offset: 0x325c

INTR_CTLR_CHANNEL6_SLICE10_IDR_0
Offset: 0x329c

INTR_CTLR_CHANNEL6_SLICE11_IDR_0
Offset: 0x32dc

INTR_CTLR_CHANNEL6_SLICE12_IDR_0
Offset: 0x331c

INTR_CTLR_CHANNEL6_SLICE13_IDR_0
Offset: 0x335c

INTR_CTLR_CHANNEL6_SLICE14_IDR_0
Offset: 0x339c

INTR_CTLR_CHANNEL6_SLICE15_IDR_0
Offset: 0x33dc

INTR_CTLR_CHANNEL6_SLICE16_IDR_0
Offset: 0x341c

INTR_CTLR_CHANNEL7_SLICE0_IDR_0
Offset: 0x381c

INTR_CTLR_CHANNEL7_SLICE1_IDR_0
Offset: 0x385c

INTR_CTLR_CHANNEL7_SLICE2_IDR_0
Offset: 0x389c

INTR_CTLR_CHANNEL7_SLICE3_IDR_0
Offset: 0x38dc

INTR_CTLR_CHANNEL7_SLICE4_IDR_0
Offset: 0x391c

INTR_CTLR_CHANNEL7_SLICE5_IDR_0
Offset: 0x395c

INTR_CTLR_CHANNEL7_SLICE6_IDR_0
Offset: 0x399c

INTR_CTLR_CHANNEL7_SLICE7_IDR_0
Offset: 0x39dc

INTR_CTLR_CHANNEL7_SLICE8_IDR_0
Offset: 0x3a1c

INTR_CTLR_CHANNEL7_SLICE9_IDR_0
Offset: 0x3a5c

INTR_CTLR_CHANNEL7_SLICE10_IDR_0
Offset: 0x3a9c

INTR_CTLR_CHANNEL7_SLICE11_IDR_0
Offset: 0x3adc

INTR_CTLR_CHANNEL7_SLICE12_IDR_0
Offset: 0x3b1c

INTR_CTLR_CHANNEL7_SLICE13_IDR_0
Offset: 0x3b5c

INTR_CTLR_CHANNEL7_SLICE14_IDR_0
Offset: 0x3b9c

INTR_CTLR_CHANNEL7_SLICE15_IDR_0
Offset: 0x3bdc

INTR_CTLR_CHANNEL7_SLICE16_IDR_0
Offset: 0x3c1c

INTR_CTLR_CHANNEL8_SLICE0_IDR_0
Offset: 0x401c

INTR_CTLR_CHANNEL8_SLICE1_IDR_0
Offset: 0x405c

INTR_CTLR_CHANNEL8_SLICE2_IDR_0
Offset: 0x409c

INTR_CTLR_CHANNEL8_SLICE3_IDR_0
Offset: 0x40dc

INTR_CTLR_CHANNEL8_SLICE4_IDR_0
Offset: 0x411c

INTR_CTLR_CHANNEL8_SLICE5_IDR_0
Offset: 0x415c

INTR_CTLR_CHANNEL8_SLICE6_IDR_0
Offset: 0x419c

INTR_CTLR_CHANNEL8_SLICE7_IDR_0
Offset: 0x41dc

INTR_CTLR_CHANNEL8_SLICE8_IDR_0
Offset: 0x421c

INTR_CTLR_CHANNEL8_SLICE9_IDR_0
Offset: 0x425c

INTR_CTLR_CHANNEL8_SLICE10_IDR_0
Offset: 0x429c

INTR_CTLR_CHANNEL8_SLICE11_IDR_0
Offset: 0x42dc

INTR_CTLR_CHANNEL8_SLICE12_IDR_0
Offset: 0x431c

INTR_CTLR_CHANNEL8_SLICE13_IDR_0
Offset: 0x435c

INTR_CTLR_CHANNEL8_SLICE14_IDR_0
Offset: 0x439c

INTR_CTLR_CHANNEL8_SLICE15_IDR_0
Offset: 0x43dc

INTR_CTLR_CHANNEL8_SLICE16_IDR_0
Offset: 0x441c

INTR_CTLR_CHANNEL9_SLICE0_IDR_0
Offset: 0x481c

INTR_CTLR_CHANNEL9_SLICE1_IDR_0
Offset: 0x485c

INTR_CTLR_CHANNEL9_SLICE2_IDR_0
Offset: 0x489c

INTR_CTLR_CHANNEL9_SLICE3_IDR_0
Offset: 0x48dc

INTR_CTLR_CHANNEL9_SLICE4_IDR_0
Offset: 0x491c

INTR_CTLR_CHANNEL9_SLICE5_IDR_0
Offset: 0x495c

INTR_CTLR_CHANNEL9_SLICE6_IDR_0
Offset: 0x499c

INTR_CTLR_CHANNEL9_SLICE7_IDR_0
Offset: 0x49dc

INTR_CTLR_CHANNEL9_SLICE8_IDR_0
Offset: 0x4a1c

INTR_CTLR_CHANNEL9_SLICE9_IDR_0
Offset: 0x4a5c

INTR_CTLR_CHANNEL9_SLICE10_IDR_0
Offset: 0x4a9c

INTR_CTLR_CHANNEL9_SLICE11_IDR_0
Offset: 0x4adc

INTR_CTLR_CHANNEL9_SLICE12_IDR_0
Offset: 0x4b1c

INTR_CTLR_CHANNEL9_SLICE13_IDR_0
Offset: 0x4b5c

INTR_CTLR_CHANNEL9_SLICE14_IDR_0
Offset: 0x4b9c

INTR_CTLR_CHANNEL9_SLICE15_IDR_0
Offset: 0x4bdc

INTR_CTLR_CHANNEL9_SLICE16_IDR_0
Offset: 0x4c1c

INTR_CTLR_CHANNEL10_SLICE0_IDR_0
Offset: 0x501c

INTR_CTLR_CHANNEL10_SLICE1_IDR_0
Offset: 0x505c

INTR_CTLR_CHANNEL10_SLICE2_IDR_0
Offset: 0x509c

INTR_CTLR_CHANNEL10_SLICE3_IDR_0
Offset: 0x50dc

INTR_CTLR_CHANNEL10_SLICE4_IDR_0
Offset: 0x511c

INTR_CTLR_CHANNEL10_SLICE5_IDR_0
Offset: 0x515c

INTR_CTLR_CHANNEL10_SLICE6_IDR_0
Offset: 0x519c

INTR_CTLR_CHANNEL10_SLICE7_IDR_0
Offset: 0x51dc

INTR_CTLR_CHANNEL10_SLICE8_IDR_0
Offset: 0x521c

INTR_CTLR_CHANNEL10_SLICE9_IDR_0
Offset: 0x525c

INTR_CTLR_CHANNEL10_SLICE10_IDR_0
Offset: 0x529c

INTR_CTLR_CHANNEL10_SLICE11_IDR_0
Offset: 0x52dc

INTR_CTLR_CHANNEL10_SLICE12_IDR_0
Offset: 0x531c

INTR_CTLR_CHANNEL10_SLICE13_IDR_0
Offset: 0x535c

INTR_CTLR_CHANNEL10_SLICE14_IDR_0
Offset: 0x539c

INTR_CTLR_CHANNEL10_SLICE15_IDR_0
Offset: 0x53dc

INTR_CTLR_CHANNEL10_SLICE16_IDR_0
Offset: 0x541c

INTR_CTLR_CHANNEL11_SLICE0_IDR_0
Offset: 0x581c

INTR_CTLR_CHANNEL11_SLICE1_IDR_0
Offset: 0x585c

INTR_CTLR_CHANNEL11_SLICE2_IDR_0
Offset: 0x589c

INTR_CTLR_CHANNEL11_SLICE3_IDR_0
Offset: 0x58dc

INTR_CTLR_CHANNEL11_SLICE4_IDR_0
Offset: 0x591c

INTR_CTLR_CHANNEL11_SLICE5_IDR_0
Offset: 0x595c

INTR_CTLR_CHANNEL11_SLICE6_IDR_0
Offset: 0x599c

INTR_CTLR_CHANNEL11_SLICE7_IDR_0
Offset: 0x59dc

INTR_CTLR_CHANNEL11_SLICE8_IDR_0
Offset: 0x5a1c

INTR_CTLR_CHANNEL11_SLICE9_IDR_0
Offset: 0x5a5c

INTR_CTLR_CHANNEL11_SLICE10_IDR_0
Offset: 0x5a9c

INTR_CTLR_CHANNEL11_SLICE11_IDR_0
Offset: 0x5adc

INTR_CTLR_CHANNEL11_SLICE12_IDR_0
Offset: 0x5b1c

INTR_CTLR_CHANNEL11_SLICE13_IDR_0
Offset: 0x5b5c

INTR_CTLR_CHANNEL11_SLICE14_IDR_0
Offset: 0x5b9c

INTR_CTLR_CHANNEL11_SLICE15_IDR_0
Offset: 0x5bdc

INTR_CTLR_CHANNEL11_SLICE16_IDR_0
Offset: 0x5c1c

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: INTR_CTLR_CHANNEL<*i*>_SCR_CHANNEL<*i*>_SCRD_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**IDR** Each bit for the corresponding bit in IER. 0: Not disabled. 1: Disabled.

INTR_CTLR_CHANNEL<*i*>_SLICE<*j*>_IDR_SET_0,

where <*i*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, and

<*j*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16.

Each bit of this register is used to set the corresponding bit in the Interrupt Disable Register (IDR) for the specific Channel/Slice.

INTR_CTLR_CHANNEL0_SLICE0_IDR_SET_0
Offset: 0x20

INTR_CTLR_CHANNEL0_SLICE1_IDR_SET_0
Offset: 0x60

INTR_CTLR_CHANNEL0_SLICE2_IDR_SET_0
Offset: 0xa0

INTR_CTLR_CHANNEL0_SLICE3_IDR_SET_0
Offset: 0xe0

INTR_CTLR_CHANNEL0_SLICE4_IDR_SET_0
Offset: 0x120

INTR_CTLR_CHANNEL0_SLICE5_IDR_SET_0
Offset: 0x160

INTR_CTLR_CHANNEL0_SLICE6_IDR_SET_0
Offset: 0x1a0

INTR_CTLR_CHANNEL0_SLICE7_IDR_SET_0
Offset: 0x1e0

INTR_CTLR_CHANNEL0_SLICE8_IDR_SET_0
Offset: 0x220

INTR_CTLR_CHANNEL0_SLICE9_IDR_SET_0
Offset: 0x260

INTR_CTLR_CHANNEL0_SLICE10_IDR_SET_0
Offset: 0x2a0

INTR_CTLR_CHANNEL0_SLICE11_IDR_SET_0
Offset: 0x2e0

INTR_CTLR_CHANNEL0_SLICE12_IDR_SET_0
Offset: 0x320

INTR_CTLR_CHANNEL0_SLICE13_IDR_SET_0
Offset: 0x360

INTR_CTLR_CHANNEL0_SLICE14_IDR_SET_0
Offset: 0x3a0

INTR_CTLR_CHANNEL0_SLICE15_IDR_SET_0
Offset: 0x3e0

INTR_CTLR_CHANNEL0_SLICE16_IDR_SET_0
Offset: 0x420

INTR_CTLR_CHANNEL1_SLICE0_IDR_SET_0
Offset: 0x820

INTR_CTLR_CHANNEL1_SLICE1_IDR_SET_0
Offset: 0x860

INTR_CTLR_CHANNEL1_SLICE2_IDR_SET_0
Offset: 0x8a0

INTR_CTLR_CHANNEL1_SLICE3_IDR_SET_0
Offset: 0x8e0

INTR_CTLR_CHANNEL1_SLICE4_IDR_SET_0
Offset: 0x920

INTR_CTLR_CHANNEL1_SLICE5_IDR_SET_0
Offset: 0x960

INTR_CTLR_CHANNEL1_SLICE6_IDR_SET_0
Offset: 0x9a0

INTR_CTLR_CHANNEL1_SLICE7_IDR_SET_0
Offset: 0x9e0

INTR_CTLR_CHANNEL1_SLICE8_IDR_SET_0
Offset: 0xa20

INTR_CTLR_CHANNEL1_SLICE9_IDR_SET_0
Offset: 0xa60

INTR_CTLR_CHANNEL1_SLICE10_IDR_SET_0
Offset: 0xaa0

INTR_CTLR_CHANNEL1_SLICE11_IDR_SET_0
Offset: 0xae0

INTR_CTLR_CHANNEL1_SLICE12_IDR_SET_0
Offset: 0xb20

INTR_CTLR_CHANNEL1_SLICE13_IDR_SET_0
Offset: 0xb60

INTR_CTLR_CHANNEL1_SLICE14_IDR_SET_0
Offset: 0xba0

INTR_CTLR_CHANNEL1_SLICE15_IDR_SET_0
Offset: 0xbe0

INTR_CTLR_CHANNEL1_SLICE16_IDR_SET_0
Offset: 0xc20

INTR_CTLR_CHANNEL2_SLICE0_IDR_SET_0
Offset: 0x1020

INTR_CTLR_CHANNEL2_SLICE1_IDR_SET_0
Offset: 0x1060

INTR_CTLR_CHANNEL2_SLICE2_IDR_SET_0
Offset: 0x10a0

INTR_CTLR_CHANNEL2_SLICE3_IDR_SET_0
Offset: 0x10e0

INTR_CTLR_CHANNEL2_SLICE4_IDR_SET_0
Offset: 0x1120

INTR_CTLR_CHANNEL2_SLICE5_IDR_SET_0
Offset: 0x1160

INTR_CTLR_CHANNEL2_SLICE6_IDR_SET_0
Offset: 0x11a0

INTR_CTLR_CHANNEL2_SLICE7_IDR_SET_0
Offset: 0x11e0

INTR_CTLR_CHANNEL2_SLICE8_IDR_SET_0
Offset: 0x1220

INTR_CTLR_CHANNEL2_SLICE9_IDR_SET_0
Offset: 0x1260

INTR_CTLR_CHANNEL2_SLICE10_IDR_SET_0
Offset: 0x12a0

INTR_CTLR_CHANNEL2_SLICE11_IDR_SET_0
Offset: 0x12e0

INTR_CTLR_CHANNEL2_SLICE12_IDR_SET_0
Offset: 0x1320

INTR_CTLR_CHANNEL2_SLICE13_IDR_SET_0
Offset: 0x1360

INTR_CTLR_CHANNEL2_SLICE14_IDR_SET_0
Offset: 0x13a0

INTR_CTLR_CHANNEL2_SLICE15_IDR_SET_0
Offset: 0x13e0

INTR_CTLR_CHANNEL2_SLICE16_IDR_SET_0
Offset: 0x1420

INTR_CTLR_CHANNEL3_SLICE0_IDR_SET_0
Offset: 0x1820

INTR_CTLR_CHANNEL3_SLICE1_IDR_SET_0
Offset: 0x1860

INTR_CTLR_CHANNEL3_SLICE2_IDR_SET_0
Offset: 0x18a0

INTR_CTLR_CHANNEL3_SLICE3_IDR_SET_0
Offset: 0x18e0

INTR_CTLR_CHANNEL3_SLICE4_IDR_SET_0
Offset: 0x1920

INTR_CTLR_CHANNEL3_SLICE5_IDR_SET_0
Offset: 0x1960

INTR_CTLR_CHANNEL3_SLICE6_IDR_SET_0
Offset: 0x19a0

INTR_CTLR_CHANNEL3_SLICE7_IDR_SET_0
Offset: 0x19e0

INTR_CTLR_CHANNEL3_SLICE8_IDR_SET_0
Offset: 0x1a20

INTR_CTLR_CHANNEL3_SLICE9_IDR_SET_0
Offset: 0x1a60

INTR_CTLR_CHANNEL3_SLICE10_IDR_SET_0
Offset: 0x1aa0

INTR_CTLR_CHANNEL3_SLICE11_IDR_SET_0
Offset: 0x1ae0

INTR_CTLR_CHANNEL3_SLICE12_IDR_SET_0
Offset: 0x1b20

INTR_CTLR_CHANNEL3_SLICE13_IDR_SET_0
Offset: 0x1b60

INTR_CTLR_CHANNEL3_SLICE14_IDR_SET_0
Offset: 0x1ba0

INTR_CTLR_CHANNEL3_SLICE15_IDR_SET_0
Offset: 0x1be0

INTR_CTLR_CHANNEL3_SLICE16_IDR_SET_0
Offset: 0x1c20

INTR_CTLR_CHANNEL4_SLICE0_IDR_SET_0
Offset: 0x2020

INTR_CTLR_CHANNEL4_SLICE1_IDR_SET_0
Offset: 0x2060

INTR_CTLR_CHANNEL4_SLICE2_IDR_SET_0
Offset: 0x20a0

INTR_CTLR_CHANNEL4_SLICE3_IDR_SET_0
Offset: 0x20e0

INTR_CTLR_CHANNEL4_SLICE4_IDR_SET_0
Offset: 0x2120

INTR_CTLR_CHANNEL4_SLICE5_IDR_SET_0
Offset: 0x2160

INTR_CTLR_CHANNEL4_SLICE6_IDR_SET_0
Offset: 0x21a0

INTR_CTLR_CHANNEL4_SLICE7_IDR_SET_0
Offset: 0x21e0

INTR_CTLR_CHANNEL4_SLICE8_IDR_SET_0
Offset: 0x2220

INTR_CTLR_CHANNEL4_SLICE9_IDR_SET_0
Offset: 0x2260

INTR_CTLR_CHANNEL4_SLICE10_IDR_SET_0
Offset: 0x22a0

INTR_CTLR_CHANNEL4_SLICE11_IDR_SET_0
Offset: 0x22e0

INTR_CTLR_CHANNEL4_SLICE12_IDR_SET_0
Offset: 0x2320

INTR_CTLR_CHANNEL4_SLICE13_IDR_SET_0
Offset: 0x2360

INTR_CTLR_CHANNEL4_SLICE14_IDR_SET_0
Offset: 0x23a0

INTR_CTLR_CHANNEL4_SLICE15_IDR_SET_0
Offset: 0x23e0

INTR_CTLR_CHANNEL4_SLICE16_IDR_SET_0
Offset: 0x2420

INTR_CTLR_CHANNEL5_SLICE0_IDR_SET_0
Offset: 0x2820

INTR_CTLR_CHANNEL5_SLICE1_IDR_SET_0
Offset: 0x2860

INTR_CTLR_CHANNEL5_SLICE2_IDR_SET_0
Offset: 0x28a0

INTR_CTLR_CHANNEL5_SLICE3_IDR_SET_0
Offset: 0x28e0

INTR_CTLR_CHANNEL5_SLICE4_IDR_SET_0
Offset: 0x2920

INTR_CTLR_CHANNEL5_SLICE5_IDR_SET_0
Offset: 0x2960

INTR_CTLR_CHANNEL5_SLICE6_IDR_SET_0
Offset: 0x29a0

INTR_CTLR_CHANNEL5_SLICE7_IDR_SET_0
Offset: 0x29e0

INTR_CTLR_CHANNEL5_SLICE8_IDR_SET_0
Offset: 0x2a20

INTR_CTLR_CHANNEL5_SLICE9_IDR_SET_0
Offset: 0x2a60

INTR_CTLR_CHANNEL5_SLICE10_IDR_SET_0
Offset: 0x2aa0

INTR_CTLR_CHANNEL5_SLICE11_IDR_SET_0
Offset: 0x2ae0

INTR_CTLR_CHANNEL5_SLICE12_IDR_SET_0
Offset: 0x2b20

INTR_CTLR_CHANNEL5_SLICE13_IDR_SET_0
Offset: 0x2b60

INTR_CTLR_CHANNEL5_SLICE14_IDR_SET_0
Offset: 0x2ba0

INTR_CTLR_CHANNEL5_SLICE15_IDR_SET_0
Offset: 0x2be0

INTR_CTLR_CHANNEL5_SLICE16_IDR_SET_0
Offset: 0x2c20

INTR_CTLR_CHANNEL6_SLICE0_IDR_SET_0
Offset: 0x3020

INTR_CTLR_CHANNEL6_SLICE1_IDR_SET_0
Offset: 0x3060

INTR_CTLR_CHANNEL6_SLICE2_IDR_SET_0
Offset: 0x30a0

INTR_CTLR_CHANNEL6_SLICE3_IDR_SET_0
Offset: 0x30e0

INTR_CTLR_CHANNEL6_SLICE4_IDR_SET_0
Offset: 0x3120

INTR_CTLR_CHANNEL6_SLICE5_IDR_SET_0
Offset: 0x3160

INTR_CTLR_CHANNEL6_SLICE6_IDR_SET_0
Offset: 0x31a0

INTR_CTLR_CHANNEL6_SLICE7_IDR_SET_0
Offset: 0x31e0

INTR_CTLR_CHANNEL6_SLICE8_IDR_SET_0
Offset: 0x3220

INTR_CTLR_CHANNEL6_SLICE9_IDR_SET_0
Offset: 0x3260

INTR_CTLR_CHANNEL6_SLICE10_IDR_SET_0
Offset: 0x32a0

INTR_CTLR_CHANNEL6_SLICE11_IDR_SET_0
Offset: 0x32e0

INTR_CTLR_CHANNEL6_SLICE12_IDR_SET_0
Offset: 0x3320

INTR_CTLR_CHANNEL6_SLICE13_IDR_SET_0
Offset: 0x3360

INTR_CTLR_CHANNEL6_SLICE14_IDR_SET_0
Offset: 0x33a0

INTR_CTLR_CHANNEL6_SLICE15_IDR_SET_0
Offset: 0x33e0

INTR_CTLR_CHANNEL6_SLICE16_IDR_SET_0
Offset: 0x3420

INTR_CTLR_CHANNEL7_SLICE0_IDR_SET_0
Offset: 0x3820

INTR_CTLR_CHANNEL7_SLICE1_IDR_SET_0
Offset: 0x3860

INTR_CTLR_CHANNEL7_SLICE2_IDR_SET_0
Offset: 0x38a0

INTR_CTLR_CHANNEL7_SLICE3_IDR_SET_0
Offset: 0x38e0

INTR_CTLR_CHANNEL7_SLICE4_IDR_SET_0
Offset: 0x3920

INTR_CTLR_CHANNEL7_SLICE5_IDR_SET_0
Offset: 0x3960

INTR_CTLR_CHANNEL7_SLICE6_IDR_SET_0
Offset: 0x39a0

INTR_CTLR_CHANNEL7_SLICE7_IDR_SET_0
Offset: 0x39e0

INTR_CTLR_CHANNEL7_SLICE8_IDR_SET_0
Offset: 0x3a20

INTR_CTLR_CHANNEL7_SLICE9_IDR_SET_0
Offset: 0x3a60

INTR_CTLR_CHANNEL7_SLICE10_IDR_SET_0
Offset: 0x3aa0

INTR_CTLR_CHANNEL7_SLICE11_IDR_SET_0
Offset: 0x3ae0

INTR_CTLR_CHANNEL7_SLICE12_IDR_SET_0
Offset: 0x3b20

INTR_CTLR_CHANNEL7_SLICE13_IDR_SET_0
Offset: 0x3b60

INTR_CTLR_CHANNEL7_SLICE14_IDR_SET_0
Offset: 0x3ba0

INTR_CTLR_CHANNEL7_SLICE15_IDR_SET_0
Offset: 0x3be0

INTR_CTLR_CHANNEL7_SLICE16_IDR_SET_0
Offset: 0x3c20

INTR_CTLR_CHANNEL8_SLICE0_IDR_SET_0
Offset: 0x4020

INTR_CTLR_CHANNEL8_SLICE1_IDR_SET_0
Offset: 0x4060

INTR_CTLR_CHANNEL8_SLICE2_IDR_SET_0
Offset: 0x40a0

INTR_CTLR_CHANNEL8_SLICE3_IDR_SET_0
Offset: 0x40e0

INTR_CTLR_CHANNEL8_SLICE4_IDR_SET_0
Offset: 0x4120

INTR_CTLR_CHANNEL8_SLICE5_IDR_SET_0
Offset: 0x4160

INTR_CTLR_CHANNEL8_SLICE6_IDR_SET_0
Offset: 0x41a0

INTR_CTLR_CHANNEL8_SLICE7_IDR_SET_0
Offset: 0x41e0

INTR_CTLR_CHANNEL8_SLICE8_IDR_SET_0
Offset: 0x4220

INTR_CTLR_CHANNEL8_SLICE9_IDR_SET_0
Offset: 0x4260

INTR_CTLR_CHANNEL8_SLICE10_IDR_SET_0
Offset: 0x42a0

INTR_CTLR_CHANNEL8_SLICE11_IDR_SET_0
Offset: 0x42e0

INTR_CTLR_CHANNEL8_SLICE12_IDR_SET_0
Offset: 0x4320

INTR_CTLR_CHANNEL8_SLICE13_IDR_SET_0
Offset: 0x4360

INTR_CTLR_CHANNEL8_SLICE14_IDR_SET_0
Offset: 0x43a0

INTR_CTLR_CHANNEL8_SLICE15_IDR_SET_0
Offset: 0x43e0

INTR_CTLR_CHANNEL8_SLICE16_IDR_SET_0
Offset: 0x4420

INTR_CTLR_CHANNEL9_SLICE0_IDR_SET_0
Offset: 0x4820

INTR_CTLR_CHANNEL9_SLICE1_IDR_SET_0
Offset: 0x4860

INTR_CTLR_CHANNEL9_SLICE2_IDR_SET_0
Offset: 0x48a0

INTR_CTLR_CHANNEL9_SLICE3_IDR_SET_0
Offset: 0x48e0

INTR_CTLR_CHANNEL9_SLICE4_IDR_SET_0
Offset: 0x4920

INTR_CTLR_CHANNEL9_SLICE5_IDR_SET_0
Offset: 0x4960

INTR_CTLR_CHANNEL9_SLICE6_IDR_SET_0
Offset: 0x49a0

INTR_CTLR_CHANNEL9_SLICE7_IDR_SET_0
Offset: 0x49e0

INTR_CTLR_CHANNEL9_SLICE8_IDR_SET_0
Offset: 0x4a20

INTR_CTLR_CHANNEL9_SLICE9_IDR_SET_0
Offset: 0x4a60

INTR_CTLR_CHANNEL9_SLICE10_IDR_SET_0
Offset: 0x4aa0

INTR_CTLR_CHANNEL9_SLICE11_IDR_SET_0
Offset: 0x4ae0

INTR_CTLR_CHANNEL9_SLICE12_IDR_SET_0
Offset: 0x4b20

INTR_CTLR_CHANNEL9_SLICE13_IDR_SET_0
Offset: 0x4b60

INTR_CTLR_CHANNEL9_SLICE14_IDR_SET_0
Offset: 0x4ba0

INTR_CTLR_CHANNEL9_SLICE15_IDR_SET_0
Offset: 0x4be0

INTR_CTLR_CHANNEL9_SLICE16_IDR_SET_0
Offset: 0x4c20

INTR_CTLR_CHANNEL10_SLICE0_IDR_SET_0
Offset: 0x5020

INTR_CTLR_CHANNEL10_SLICE1_IDR_SET_0
Offset: 0x5060

INTR_CTLR_CHANNEL10_SLICE2_IDR_SET_0
Offset: 0x50a0

INTR_CTLR_CHANNEL10_SLICE3_IDR_SET_0
Offset: 0x50e0

INTR_CTLR_CHANNEL10_SLICE4_IDR_SET_0
Offset: 0x5120

INTR_CTLR_CHANNEL10_SLICE5_IDR_SET_0
Offset: 0x5160

INTR_CTLR_CHANNEL10_SLICE6_IDR_SET_0
Offset: 0x51a0

INTR_CTLR_CHANNEL10_SLICE7_IDR_SET_0
Offset: 0x51e0

INTR_CTLR_CHANNEL10_SLICE8_IDR_SET_0
Offset: 0x5220

INTR_CTLR_CHANNEL10_SLICE9_IDR_SET_0
Offset: 0x5260

INTR_CTLR_CHANNEL10_SLICE10_IDR_SET_0
Offset: 0x52a0

INTR_CTLR_CHANNEL10_SLICE11_IDR_SET_0
Offset: 0x52e0

INTR_CTLR_CHANNEL10_SLICE12_IDR_SET_0
Offset: 0x5320

INTR_CTLR_CHANNEL10_SLICE13_IDR_SET_0
Offset: 0x5360

INTR_CTLR_CHANNEL10_SLICE14_IDR_SET_0
Offset: 0x53a0

INTR_CTLR_CHANNEL10_SLICE15_IDR_SET_0
Offset: 0x53e0

INTR_CTLR_CHANNEL10_SLICE16_IDR_SET_0
Offset: 0x5420

INTR_CTLR_CHANNEL11_SLICE0_IDR_SET_0
Offset: 0x5820

INTR_CTLR_CHANNEL11_SLICE1_IDR_SET_0
Offset: 0x5860

INTR_CTLR_CHANNEL11_SLICE2_IDR_SET_0
Offset: 0x58a0

INTR_CTLR_CHANNEL11_SLICE3_IDR_SET_0
Offset: 0x58e0

INTR_CTLR_CHANNEL11_SLICE4_IDR_SET_0
Offset: 0x5920

INTR_CTLR_CHANNEL11_SLICE5_IDR_SET_0
Offset: 0x5960

INTR_CTLR_CHANNEL11_SLICE6_IDR_SET_0
Offset: 0x59a0

INTR_CTLR_CHANNEL11_SLICE7_IDR_SET_0
Offset: 0x59e0

INTR_CTLR_CHANNEL11_SLICE8_IDR_SET_0
Offset: 0x5a20

INTR_CTLR_CHANNEL11_SLICE9_IDR_SET_0
Offset: 0x5a60

INTR_CTLR_CHANNEL11_SLICE10_IDR_SET_0
Offset: 0x5aa0

INTR_CTLR_CHANNEL11_SLICE11_IDR_SET_0
Offset: 0x5ae0

INTR_CTLR_CHANNEL11_SLICE12_IDR_SET_0
Offset: 0x5b20

INTR_CTLR_CHANNEL11_SLICE13_IDR_SET_0
Offset: 0x5b60

INTR_CTLR_CHANNEL11_SLICE14_IDR_SET_0
Offset: 0x5ba0

INTR_CTLR_CHANNEL11_SLICE15_IDR_SET_0
Offset: 0x5be0

INTR_CTLR_CHANNEL11_SLICE16_IDR_SET_0
Offset: 0x5c20

Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: INTR_CTLR_CHANNEL<*i*>_SCR_CHANNEL0_SCRD_0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**IDR_SET** Each bit for the corresponding bit in IDR. 0: No action. 1: Set the bit.

INTR_CTLR_CHANNEL<*i*>_SLICE<*j*>_IDR_CLR_0,

where <*i*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, and

<*j*> = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16.

Each bit of this register is used to clear the corresponding bit in the Interrupt Disable Register (IDR) for the specific Channel/Slice.

INTR_CTLR_CHANNEL0_SLICE0_IDR_CLR_0
Offset: 0x24

INTR_CTLR_CHANNEL0_SLICE1_IDR_CLR_0
Offset: 0x64

INTR_CTLR_CHANNEL0_SLICE2_IDR_CLR_0
Offset: 0xa4

INTR_CTLR_CHANNEL0_SLICE3_IDR_CLR_0
Offset: 0xe4

INTR_CTLR_CHANNEL0_SLICE4_IDR_CLR_0
Offset: 0x124

INTR_CTLR_CHANNEL0_SLICE5_IDR_CLR_0
Offset: 0x164

INTR_CTLR_CHANNEL0_SLICE6_IDR_CLR_0
Offset: 0x1a4

INTR_CTLR_CHANNEL0_SLICE7_IDR_CLR_0
Offset: 0x1e4

INTR_CTLR_CHANNEL0_SLICE8_IDR_CLR_0
Offset: 0x224

INTR_CTLR_CHANNEL0_SLICE9_IDR_CLR_0
Offset: 0x264

INTR_CTLR_CHANNEL0_SLICE10_IDR_CLR_0
Offset: 0x2a4

INTR_CTLR_CHANNEL0_SLICE11_IDR_CLR_0
Offset: 0x2e4

INTR_CTLR_CHANNEL0_SLICE12_IDR_CLR_0
Offset: 0x324

INTR_CTLR_CHANNEL0_SLICE13_IDR_CLR_0
Offset: 0x364

INTR_CTLR_CHANNEL0_SLICE14_IDR_CLR_0
Offset: 0x3a4

INTR_CTLR_CHANNEL0_SLICE15_IDR_CLR_0
Offset: 0x3e4

INTR_CTLR_CHANNEL0_SLICE16_IDR_CLR_0
Offset: 0x424

INTR_CTLR_CHANNEL1_SLICE0_IDR_CLR_0
Offset: 0x824

INTR_CTLR_CHANNEL1_SLICE1_IDR_CLR_0
Offset: 0x864

INTR_CTLR_CHANNEL1_SLICE2_IDR_CLR_0
Offset: 0x8a4

INTR_CTLR_CHANNEL1_SLICE3_IDR_CLR_0
Offset: 0x8e4

INTR_CTLR_CHANNEL1_SLICE4_IDR_CLR_0
Offset: 0x924

INTR_CTLR_CHANNEL1_SLICE5_IDR_CLR_0
Offset: 0x964

INTR_CTLR_CHANNEL1_SLICE6_IDR_CLR_0
Offset: 0x9a4

INTR_CTLR_CHANNEL1_SLICE7_IDR_CLR_0
Offset: 0x9e4

INTR_CTLR_CHANNEL1_SLICE8_IDR_CLR_0
Offset: 0xa24

INTR_CTLR_CHANNEL1_SLICE9_IDR_CLR_0
Offset: 0xa64

INTR_CTLR_CHANNEL1_SLICE10_IDR_CLR_0
Offset: 0xaa4

INTR_CTLR_CHANNEL1_SLICE11_IDR_CLR_0
Offset: 0xae4

INTR_CTLR_CHANNEL1_SLICE12_IDR_CLR_0
Offset: 0xb24

INTR_CTLR_CHANNEL1_SLICE13_IDR_CLR_0
Offset: 0xb64

INTR_CTLR_CHANNEL1_SLICE14_IDR_CLR_0
Offset: 0xba4

INTR_CTLR_CHANNEL1_SLICE15_IDR_CLR_0
Offset: 0xbe4

INTR_CTLR_CHANNEL1_SLICE16_IDR_CLR_0
Offset: 0xc24

INTR_CTLR_CHANNEL2_SLICE0_IDR_CLR_0
Offset: 0x1024

INTR_CTLR_CHANNEL2_SLICE1_IDR_CLR_0
Offset: 0x1064

INTR_CTLR_CHANNEL2_SLICE2_IDR_CLR_0
Offset: 0x10a4

INTR_CTLR_CHANNEL2_SLICE3_IDR_CLR_0
Offset: 0x10e4

INTR_CTLR_CHANNEL2_SLICE4_IDR_CLR_0
Offset: 0x1124

INTR_CTLR_CHANNEL2_SLICE5_IDR_CLR_0
Offset: 0x1164

INTR_CTLR_CHANNEL2_SLICE6_IDR_CLR_0
Offset: 0x11a4

INTR_CTLR_CHANNEL2_SLICE7_IDR_CLR_0
Offset: 0x11e4

INTR_CTLR_CHANNEL2_SLICE8_IDR_CLR_0
Offset: 0x1224

INTR_CTLR_CHANNEL2_SLICE9_IDR_CLR_0
Offset: 0x1264

INTR_CTLR_CHANNEL2_SLICE10_IDR_CLR_0
Offset: 0x12a4

INTR_CTLR_CHANNEL2_SLICE11_IDR_CLR_0
Offset: 0x12e4

INTR_CTLR_CHANNEL2_SLICE12_IDR_CLR_0
Offset: 0x1324

INTR_CTLR_CHANNEL2_SLICE13_IDR_CLR_0
Offset: 0x1364

INTR_CTLR_CHANNEL2_SLICE14_IDR_CLR_0
Offset: 0x13a4

INTR_CTLR_CHANNEL2_SLICE15_IDR_CLR_0
Offset: 0x13e4

INTR_CTLR_CHANNEL2_SLICE16_IDR_CLR_0
Offset: 0x1424

INTR_CTLR_CHANNEL3_SLICE0_IDR_CLR_0
Offset: 0x1824

INTR_CTLR_CHANNEL3_SLICE1_IDR_CLR_0
Offset: 0x1864

INTR_CTLR_CHANNEL3_SLICE2_IDR_CLR_0
Offset: 0x18a4

INTR_CTLR_CHANNEL3_SLICE3_IDR_CLR_0
Offset: 0x18e4

INTR_CTLR_CHANNEL3_SLICE4_IDR_CLR_0
Offset: 0x1924

INTR_CTLR_CHANNEL3_SLICE5_IDR_CLR_0
Offset: 0x1964

INTR_CTLR_CHANNEL3_SLICE6_IDR_CLR_0
Offset: 0x19a4

INTR_CTLR_CHANNEL3_SLICE7_IDR_CLR_0
Offset: 0x19e4

INTR_CTLR_CHANNEL3_SLICE8_IDR_CLR_0
Offset: 0x1a24

INTR_CTLR_CHANNEL3_SLICE9_IDR_CLR_0
Offset: 0x1a64

INTR_CTLR_CHANNEL3_SLICE10_IDR_CLR_0
Offset: 0x1aa4

INTR_CTLR_CHANNEL3_SLICE11_IDR_CLR_0
Offset: 0x1ae4

INTR_CTLR_CHANNEL3_SLICE12_IDR_CLR_0
Offset: 0x1b24

INTR_CTLR_CHANNEL3_SLICE13_IDR_CLR_0
Offset: 0x1b64

INTR_CTLR_CHANNEL3_SLICE14_IDR_CLR_0
Offset: 0x1ba4

INTR_CTLR_CHANNEL3_SLICE15_IDR_CLR_0
Offset: 0x1be4

INTR_CTLR_CHANNEL3_SLICE16_IDR_CLR_0
Offset: 0x1c24

INTR_CTLR_CHANNEL4_SLICE0_IDR_CLR_0
Offset: 0x2024

INTR_CTLR_CHANNEL4_SLICE1_IDR_CLR_0
Offset: 0x2064

INTR_CTLR_CHANNEL4_SLICE2_IDR_CLR_0
Offset: 0x20a4

INTR_CTLR_CHANNEL4_SLICE3_IDR_CLR_0
Offset: 0x20e4

INTR_CTLR_CHANNEL4_SLICE4_IDR_CLR_0
Offset: 0x2124

INTR_CTLR_CHANNEL4_SLICE5_IDR_CLR_0
Offset: 0x2164

INTR_CTLR_CHANNEL4_SLICE6_IDR_CLR_0
Offset: 0x21a4

INTR_CTLR_CHANNEL4_SLICE7_IDR_CLR_0
Offset: 0x21e4

INTR_CTLR_CHANNEL4_SLICE8_IDR_CLR_0
Offset: 0x2224

INTR_CTLR_CHANNEL4_SLICE9_IDR_CLR_0
Offset: 0x2264

INTR_CTLR_CHANNEL4_SLICE10_IDR_CLR_0
Offset: 0x22a4

INTR_CTLR_CHANNEL4_SLICE11_IDR_CLR_0
Offset: 0x22e4

INTR_CTLR_CHANNEL4_SLICE12_IDR_CLR_0
Offset: 0x2324

INTR_CTLR_CHANNEL4_SLICE13_IDR_CLR_0
Offset: 0x2364

INTR_CTLR_CHANNEL4_SLICE14_IDR_CLR_0
Offset: 0x23a4

INTR_CTLR_CHANNEL4_SLICE15_IDR_CLR_0
Offset: 0x23e4

INTR_CTLR_CHANNEL4_SLICE16_IDR_CLR_0
Offset: 0x2424

INTR_CTLR_CHANNEL5_SLICE0_IDR_CLR_0
Offset: 0x2824

INTR_CTLR_CHANNEL5_SLICE1_IDR_CLR_0
Offset: 0x2864

INTR_CTLR_CHANNEL5_SLICE2_IDR_CLR_0
Offset: 0x28a4

INTR_CTLR_CHANNEL5_SLICE3_IDR_CLR_0
Offset: 0x28e4

INTR_CTLR_CHANNEL5_SLICE4_IDR_CLR_0
Offset: 0x2924

INTR_CTLR_CHANNEL5_SLICE5_IDR_CLR_0
Offset: 0x2964

INTR_CTLR_CHANNEL5_SLICE6_IDR_CLR_0
Offset: 0x29a4

INTR_CTLR_CHANNEL5_SLICE7_IDR_CLR_0
Offset: 0x29e4

INTR_CTLR_CHANNEL5_SLICE8_IDR_CLR_0
Offset: 0x2a24

INTR_CTLR_CHANNEL5_SLICE9_IDR_CLR_0
Offset: 0x2a64

INTR_CTLR_CHANNEL5_SLICE10_IDR_CLR_0
Offset: 0x2aa4

INTR_CTLR_CHANNEL5_SLICE11_IDR_CLR_0
Offset: 0x2ae4

INTR_CTLR_CHANNEL5_SLICE12_IDR_CLR_0
Offset: 0x2b24

INTR_CTLR_CHANNEL5_SLICE13_IDR_CLR_0
Offset: 0x2b64

INTR_CTLR_CHANNEL5_SLICE14_IDR_CLR_0
Offset: 0x2ba4

INTR_CTLR_CHANNEL5_SLICE15_IDR_CLR_0
Offset: 0x2be4

INTR_CTLR_CHANNEL5_SLICE16_IDR_CLR_0
Offset: 0x2c24

INTR_CTLR_CHANNEL6_SLICE0_IDR_CLR_0
Offset: 0x3024

INTR_CTLR_CHANNEL6_SLICE1_IDR_CLR_0
Offset: 0x3064

INTR_CTLR_CHANNEL6_SLICE2_IDR_CLR_0
Offset: 0x30a4

INTR_CTLR_CHANNEL6_SLICE3_IDR_CLR_0
Offset: 0x30e4

INTR_CTLR_CHANNEL6_SLICE4_IDR_CLR_0
Offset: 0x3124

INTR_CTLR_CHANNEL6_SLICE5_IDR_CLR_0
Offset: 0x3164

INTR_CTLR_CHANNEL6_SLICE6_IDR_CLR_0
Offset: 0x31a4

INTR_CTLR_CHANNEL6_SLICE7_IDR_CLR_0
Offset: 0x31e4

INTR_CTLR_CHANNEL6_SLICE8_IDR_CLR_0
Offset: 0x3224

INTR_CTLR_CHANNEL6_SLICE9_IDR_CLR_0
Offset: 0x3264

INTR_CTLR_CHANNEL6_SLICE10_IDR_CLR_0
Offset: 0x32a4

INTR_CTLR_CHANNEL6_SLICE11_IDR_CLR_0
Offset: 0x32e4

INTR_CTLR_CHANNEL6_SLICE12_IDR_CLR_0
Offset: 0x3324

INTR_CTLR_CHANNEL6_SLICE13_IDR_CLR_0
Offset: 0x3364

INTR_CTLR_CHANNEL6_SLICE14_IDR_CLR_0
Offset: 0x33a4

INTR_CTLR_CHANNEL6_SLICE15_IDR_CLR_0
Offset: 0x33e4

INTR_CTLR_CHANNEL6_SLICE16_IDR_CLR_0
Offset: 0x3424

INTR_CTLR_CHANNEL7_SLICE0_IDR_CLR_0
Offset: 0x3824

INTR_CTLR_CHANNEL7_SLICE1_IDR_CLR_0
Offset: 0x3864

INTR_CTLR_CHANNEL7_SLICE2_IDR_CLR_0
Offset: 0x38a4

INTR_CTLR_CHANNEL7_SLICE3_IDR_CLR_0
Offset: 0x38e4

INTR_CTLR_CHANNEL7_SLICE4_IDR_CLR_0
Offset: 0x3924

INTR_CTLR_CHANNEL7_SLICE5_IDR_CLR_0
Offset: 0x3964

INTR_CTLR_CHANNEL7_SLICE6_IDR_CLR_0
Offset: 0x39a4

INTR_CTLR_CHANNEL7_SLICE7_IDR_CLR_0
Offset: 0x39e4

INTR_CTLR_CHANNEL7_SLICE8_IDR_CLR_0
Offset: 0x3a24

INTR_CTLR_CHANNEL7_SLICE9_IDR_CLR_0
Offset: 0x3a64

INTR_CTLR_CHANNEL7_SLICE10_IDR_CLR_0
Offset: 0x3aa4

INTR_CTLR_CHANNEL7_SLICE11_IDR_CLR_0
Offset: 0x3ae4

INTR_CTLR_CHANNEL7_SLICE12_IDR_CLR_0
Offset: 0x3b24

INTR_CTLR_CHANNEL7_SLICE13_IDR_CLR_0
Offset: 0x3b64

INTR_CTLR_CHANNEL7_SLICE14_IDR_CLR_0
Offset: 0x3ba4

INTR_CTLR_CHANNEL7_SLICE15_IDR_CLR_0
Offset: 0x3be4

INTR_CTLR_CHANNEL7_SLICE16_IDR_CLR_0
Offset: 0x3c24

INTR_CTLR_CHANNEL8_SLICE0_IDR_CLR_0
Offset: 0x4024

INTR_CTLR_CHANNEL8_SLICE1_IDR_CLR_0
Offset: 0x4064

INTR_CTLR_CHANNEL8_SLICE2_IDR_CLR_0
Offset: 0x40a4

INTR_CTLR_CHANNEL8_SLICE3_IDR_CLR_0
Offset: 0x40e4

INTR_CTLR_CHANNEL8_SLICE4_IDR_CLR_0
Offset: 0x4124

INTR_CTLR_CHANNEL8_SLICE5_IDR_CLR_0
Offset: 0x4164

INTR_CTLR_CHANNEL8_SLICE6_IDR_CLR_0
Offset: 0x41a4

INTR_CTLR_CHANNEL8_SLICE7_IDR_CLR_0
Offset: 0x41e4

INTR_CTLR_CHANNEL8_SLICE8_IDR_CLR_0
Offset: 0x4224

INTR_CTLR_CHANNEL8_SLICE9_IDR_CLR_0
Offset: 0x4264

INTR_CTLR_CHANNEL8_SLICE10_IDR_CLR_0
Offset: 0x42a4

INTR_CTLR_CHANNEL8_SLICE11_IDR_CLR_0
Offset: 0x42e4

INTR_CTLR_CHANNEL8_SLICE12_IDR_CLR_0
Offset: 0x4324

INTR_CTLR_CHANNEL8_SLICE13_IDR_CLR_0
Offset: 0x4364

INTR_CTLR_CHANNEL8_SLICE14_IDR_CLR_0
Offset: 0x43a4

INTR_CTLR_CHANNEL8_SLICE15_IDR_CLR_0
Offset: 0x43e4

INTR_CTLR_CHANNEL8_SLICE16_IDR_CLR_0
Offset: 0x4424

INTR_CTLR_CHANNEL9_SLICE0_IDR_CLR_0
Offset: 0x4824

INTR_CTLR_CHANNEL9_SLICE1_IDR_CLR_0
Offset: 0x4864

INTR_CTLR_CHANNEL9_SLICE2_IDR_CLR_0
Offset: 0x48a4

INTR_CTLR_CHANNEL9_SLICE3_IDR_CLR_0
Offset: 0x48e4

INTR_CTLR_CHANNEL9_SLICE4_IDR_CLR_0
Offset: 0x4924

INTR_CTLR_CHANNEL9_SLICE5_IDR_CLR_0
Offset: 0x4964

INTR_CTLR_CHANNEL9_SLICE6_IDR_CLR_0
Offset: 0x49a4

INTR_CTLR_CHANNEL9_SLICE7_IDR_CLR_0
Offset: 0x49e4

INTR_CTLR_CHANNEL9_SLICE8_IDR_CLR_0
Offset: 0x4a24

INTR_CTLR_CHANNEL9_SLICE9_IDR_CLR_0
Offset: 0x4a64

INTR_CTLR_CHANNEL9_SLICE10_IDR_CLR_0
Offset: 0x4aa4

INTR_CTLR_CHANNEL9_SLICE11_IDR_CLR_0
Offset: 0x4ae4

INTR_CTLR_CHANNEL9_SLICE12_IDR_CLR_0
Offset: 0x4b24

INTR_CTLR_CHANNEL9_SLICE13_IDR_CLR_0
Offset: 0x4b64

INTR_CTLR_CHANNEL9_SLICE14_IDR_CLR_0
Offset: 0x4ba4

INTR_CTLR_CHANNEL9_SLICE15_IDR_CLR_0
Offset: 0x4be4

INTR_CTLR_CHANNEL9_SLICE16_IDR_CLR_0
Offset: 0x4c24

INTR_CTLR_CHANNEL10_SLICE0_IDR_CLR_0
Offset: 0x5024

INTR_CTLR_CHANNEL10_SLICE1_IDR_CLR_0
Offset: 0x5064

INTR_CTLR_CHANNEL10_SLICE2_IDR_CLR_0
Offset: 0x50a4

INTR_CTLR_CHANNEL10_SLICE3_IDR_CLR_0
Offset: 0x50e4

INTR_CTLR_CHANNEL10_SLICE4_IDR_CLR_0
Offset: 0x5124

INTR_CTLR_CHANNEL10_SLICE5_IDR_CLR_0
Offset: 0x5164

INTR_CTLR_CHANNEL10_SLICE6_IDR_CLR_0
Offset: 0x51a4

INTR_CTLR_CHANNEL10_SLICE7_IDR_CLR_0
Offset: 0x51e4

INTR_CTLR_CHANNEL10_SLICE8_IDR_CLR_0
Offset: 0x5224

INTR_CTLR_CHANNEL10_SLICE9_IDR_CLR_0
Offset: 0x5264

INTR_CTLR_CHANNEL10_SLICE10_IDR_CLR_0
Offset: 0x52a4

INTR_CTLR_CHANNEL10_SLICE11_IDR_CLR_0
Offset: 0x52e4

INTR_CTLR_CHANNEL10_SLICE12_IDR_CLR_0
Offset: 0x5324

INTR_CTLR_CHANNEL10_SLICE13_IDR_CLR_0
Offset: 0x5364

INTR_CTLR_CHANNEL10_SLICE14_IDR_CLR_0
Offset: 0x53a4

INTR_CTLR_CHANNEL10_SLICE15_IDR_CLR_0
Offset: 0x53e4

INTR_CTLR_CHANNEL10_SLICE16_IDR_CLR_0
Offset: 0x5424

INTR_CTLR_CHANNEL11_SLICE0_IDR_CLR_0
Offset: 0x5824

INTR_CTLR_CHANNEL11_SLICE1_IDR_CLR_0
Offset: 0x5864

INTR_CTLR_CHANNEL11_SLICE2_IDR_CLR_0
Offset: 0x58a4

INTR_CTLR_CHANNEL11_SLICE3_IDR_CLR_0
Offset: 0x58e4

INTR_CTLR_CHANNEL11_SLICE4_IDR_CLR_0
Offset: 0x5924

INTR_CTLR_CHANNEL11_SLICE5_IDR_CLR_0
Offset: 0x5964

INTR_CTLR_CHANNEL11_SLICE6_IDR_CLR_0
Offset: 0x59a4

INTR_CTLR_CHANNEL11_SLICE7_IDR_CLR_0
Offset: 0x59e4

INTR_CTLR_CHANNEL11_SLICE8_IDR_CLR_0
Offset: 0x5a24

INTR_CTLR_CHANNEL11_SLICE9_IDR_CLR_0
Offset: 0x5a64

INTR_CTLR_CHANNEL11_SLICE10_IDR_CLR_0
Offset: 0x5aa4

INTR_CTLR_CHANNEL11_SLICE11_IDR_CLR_0
Offset: 0x5ae4

INTR_CTLR_CHANNEL11_SLICE12_IDR_CLR_0
Offset: 0x5b24

INTR_CTLR_CHANNEL11_SLICE13_IDR_CLR_0
Offset: 0x5b64

INTR_CTLR_CHANNEL11_SLICE14_IDR_CLR_0
Offset: 0x5ba4

INTR_CTLR_CHANNEL11_SLICE15_IDR_CLR_0
Offset: 0x5be4

INTR_CTLR_CHANNEL11_SLICE16_IDR_CLR_0
Offset: 0x5c24

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: INTR_CTLR_CHANNEL<*i*>_SCR_CHANNEL<*i*>_SCRD_0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**IDR_CLR** Each bit for the corresponding bit in IDR. 0: No action. 1: Clear the bit.

8.3 Timers

8.3.1 Overview

This chapter documents the various timers available to software in a NVIDIA® Orin™ series System-on-Chip (SoC) system. The following table summarizes the different types of timer elements.

Table 8.16 List of Timers

Name	Primary Use	Interrupt (1)	Security	Frequency Range	Notes
RTC	Wall Clock	RTC	Configurable	1 kHz	Seconds counter lockable.
AOWDT	SC7 watchdog	AOWDT	Configurable	1 kHz	Part of RTC, last resort internal watchdog.
USEC	Real-time source	-	Configurable	1 MHz	Also generates time reference pulses.

Name	Primary Use	Interrupt (1)	Security	Frequency Range	Notes
TMR	NV Generic	LIC for Top TMR15-0 + local interrupts in processor clusters	Configurable	1 MHz	TMR is part of TKE, which has multiple instances: ▪ For Top TKE, the TMR interrupts are routed to LIC and there are 16 such interrupts. ▪ For other TKE, the TMR interrupts are routed to the local interrupt controller of the processor cluster that instances the TKE, i.e., BPMP. TKE TMR interrupts are routed to the BPMP VIC attached to the BPMP Cortex-R5.
WDT	Top and local watchdog timers	LIC for Top WDT2-0 + local interrupts in processor clusters	Configurable	1 MHz	WDT is also part of TKE (see also TMR). ▪ Top TKE WDT interrupts are routed to LIC. ▪ Other TKE WDT are routed to the local interrupt controller (two first level), plus there is one remote interrupt going to LIC (third level).
TSC	Reference for GT	LIC	TrustZone/ Configurable	30 MHz	Counter value can only be updated in secure mode.
GT	Arm Generic	PPI(2)	TrustZone	TSC	These timers use TSC as reference.
A9T	Cortex®-A9 Timers	PPI	TrustZone	SCU	Operates on the CPU clock; so can be problematic if frequency is not fixed.

(1) See the Interrupt Controller chapter for the authoritative mapping of RTC/TMR/WDT interrupts.
(2) PPIs are per CPU Private Peripheral Interrupts.

All timer modules support fine grain access rights. The security model is TrustZone® for modules coming from Arm or following the Arm architecture prescriptions: TSC, GT, and A9T, and the configurable NV security models based on Security Control Register (SCR) for all other modules, including the NV-specific TSC implementation registers.

All time-keeping modules can be configured to stop during debug. This is useful to suspend the operation of WDT (to avoid resetting the chip) and the CCPLEX GT (suspending the OS tick). The only exception is the safety reference counter in RTC.

8.3.1.1 Glossary and Acronyms

Note that different names may end up in the same acronym when the entire TRM is put in perspective. The acronyms listed here are within the context of the Timers chapter. Refer also to the Glossary in the Introduction section of the TRM.

Note: NVIDIA is transitioning from the use of Master/Slave terminology in our documentation. During the transition period, there may be inconsistent use of the terminology. In this chapter, "Main" can be interpreted as "Master" and "Secondary" can be interpreted as "Slave."

Term	Definition
AO	Always On (sometimes also shown as AON)
Arm Processor Complex	Arm processor core itself plus some closely associated peripherals, including ▪ CCPLEX, the main CPU complex of eight Carmel cores ▪ Multiple Cortex-R5 clusters ▪ The Cortex-A9 ADSP inside APE
AVIC	Arm Vectored Interrupt Controller
GT	Generic Timer
GTE	Generic Timestamping Engine
LFSR	Linear Feedback Shift Register
LIC	Legacy Interrupt Controller
LNCO	Lockable NCO
MTSC	Main TSC
NCO	Numerically Controlled Oscillator
OSC	Oscillator
PCR	Present Count value Register
PTV	Present Trigger Value, the value loaded at start into an NV timer
RTC	Real-Time Clock
SMP	Symmetrical Multi-Processing
SPE	Sensor Processing Engine
TKE	Time-Keeping Element, the module instantiating NV timers and WDTs
TMR	TMR refers to NV timers
TSC	Timer's System Counter, also known as TimeStamp Counter
WDT	WatchDog Timer

8.3.1.2 Relevant Chapters in the TRM

- Address Map
- Clock Controller and Reset (CAR)
- Hardware Safety Manager (HSM)

8.3.1.3 Address Map

The address spaces (in 64 KiB pages) assigned to the different timers modules are as follows:

- Each TKE
  - One page for shared registers
  - One page per NV Timer, nT of them where nT is a configuration parameter
  - One page per WDT, nW of them where nW is a configuration parameter
- Each GTE requires one page
- The RTC requires two pages:
  - One page for the RTC proper
  - One page for the AOWDT
- The TSC requires four pages total
  - One page for each of the three Arm specified frames
  - One page for the TSC implementation registers
  - One page for the microsecond (µs) counter

8.3.1.4 Clocks

The TSC RefClk (also known as tsc_ref_clk) is specific to timers and generated inside CAR using dedicated circuitry. The base logic is a fractional divider, but with an additional adjust signal to allow for fine control of the generated frequency. The fractional divider is part of CAR with the adjust signal coming from TSC.

TKE uses the following clocks:

- APB clock
- TSC RefClk, used for the TSC input and for general operation
- TKE also uses one of these two clocks
  - OSC for the TKE instance inside the Always On partition
  - clk_m for all other TKE instances

RTC uses the following clocks:

- APB clock
- The 32,768 Hz reference clock

TSC uses the following clocks:

- APB clock
- 32,768 Hz reference clock
- TSC RefClk
- OSC clock (for the microsecond and reference pulse generation)

GTE uses the following clocks:

- APB clock

- TSC RefClk

FSI-TSC and FSI-TKE use the following clocks:

- FSI crystal clock for operation
- FSI APB clock for register access

For safety reasons, the FSI crystal clock must be protected by an FMON that acts as an independent clock source.

8.3.1.5 Resets

Most modules have an associated software reset control in CAR. FSI-TSC and FSI-TKE use the FSI system reset.

The following time-related modules do not have a software reset:

- TSC and RTC
  - As they maintain the notion of secure time and may be used to generate corresponding secure interrupts, their operations should not be impeded by non-secure software. This requirement applies to the microsecond counter (USEC) inside TSC as well.
- Top TKE
  - Top TKE can also be used to expose secure time, so has no software reset. Other TKEs, however, still have software resets.

The TSC terminal nodes are always associated with a logic block that consumes the TSC value. The TSC terminal node may share their reset with other logic block it is tightly coupled with. The following table summarizes the TSC terminal nodes and their reset logic.

Table 8.17 Terminal Nodes Reset Logic

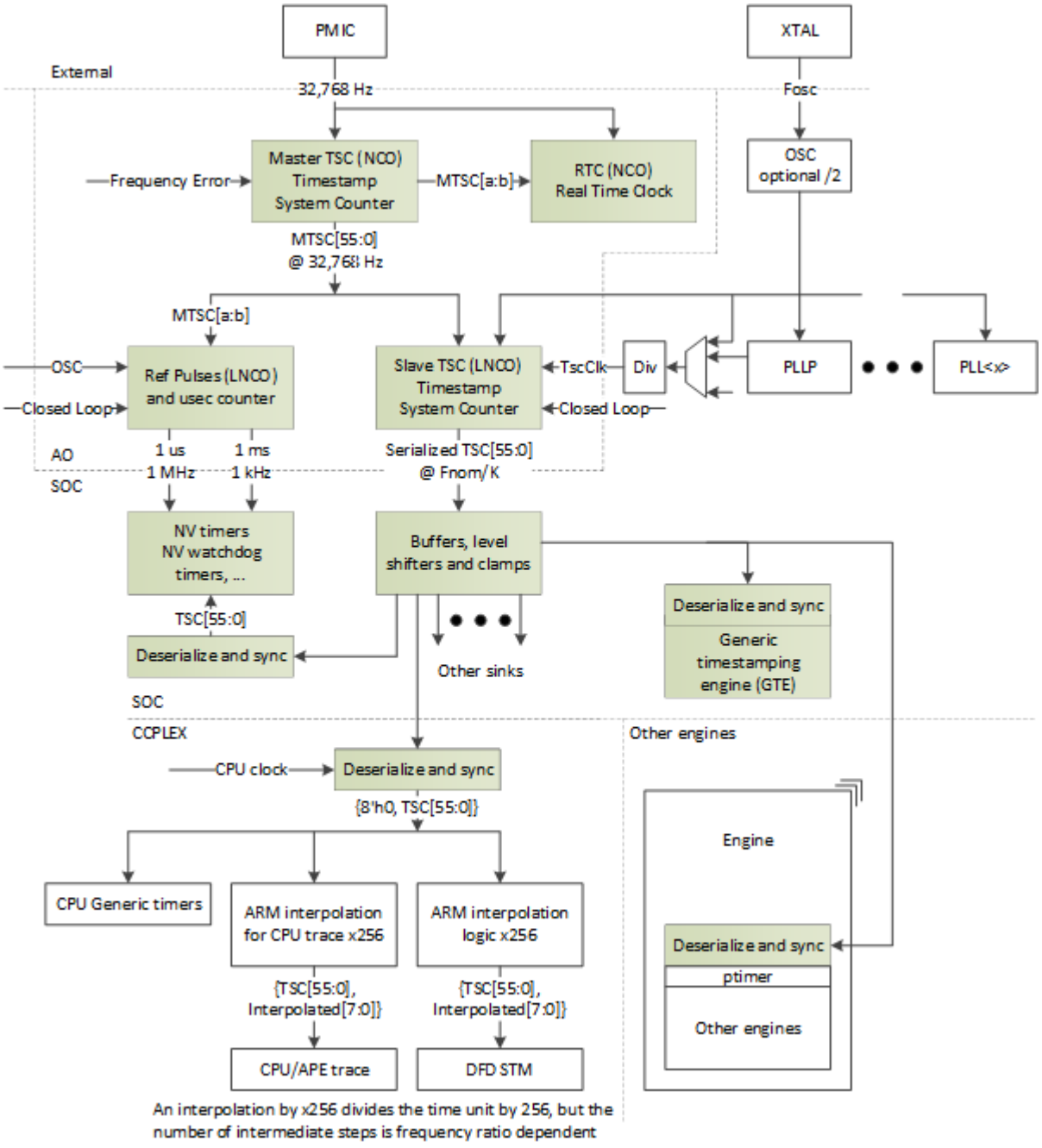
Terminal Node	Reset Logic
In AON (used for AON TKE)	By software reset, CLK_RST_CONTROLLER_RST_DEV_TSCTNAON; must be enabled by software at boot time for cold boot. Disabled and restarted by software when terminal node is power gated.
ADSP and ADMA TSC in APE	On same reset as ADSP and ADMA, no special action needed.
BPMP, SCE, RCE, and DCE	By software reset, CLK_RST_CONTROLLER_RST_DEV_TSCTN<BPMP, SCE, RCE, DCE>; must be enabled by software during respectively BPMP, SCE, RCE, and DCE startup sequence.
GPU	On same reset as GPU, no special action needed.
CCPLEX	On CCPLEX reset, no special action needed.
Backbone (Top TKE and GTE)	On system reset, no special action needed.

Terminal Node	Reset Logic
Coresight	On same reset as Coresight, no special action needed.
VI	By software reset, CLK_RST_CONTROLLER_RST_DEV_TSCTNVI; must be enabled by software during VI startup sequence.

8.3.2 Functional Description

The common reference for all timers is a frequency corrected version of the 32,768 Hz input clock. When operated in locked mode, the timers operate as a hierarchy shown below. When locking is not enabled, the different timers operate independently.

Figure 8.8 Timer Hierarchy



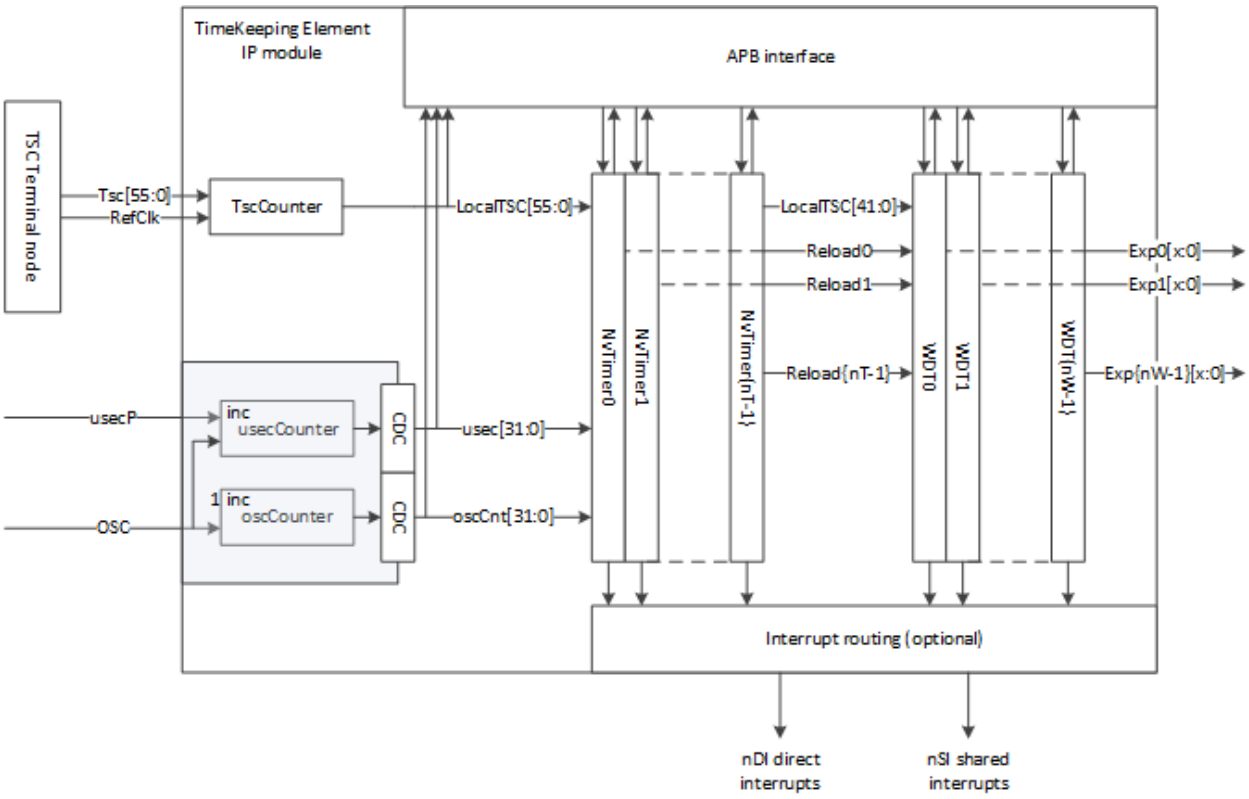
All Time-Keeping Elements (TKEs) in the SoC can be traced to the OSC or 32,768 Hz clocks. All TKEs, except TSC, can optionally take TSC as their timing reference to ensure a universal timebase. This, however, is not a mandatory mode of operation (see figure above).

Locking the reference happens via a Numerically Controlled Oscillator (NCO), i.e., a phase accumulator. An NCO can be coupled with a very simple loop filter to lock its frequency to a reference.

8.3.2.1 Hierarchical Grouping

For ease of use, NV timers and WDTs are provided in a packaged module called Time-Keeping Element (TKE). The general TKE structure is shown below.

Figure 8.9 Timekeeping Element IP Module



A TKE contains the following elements:

- A TSC value synchronized to the TKE clock. The TSC value is used both for the operation of the timers inside the TKE and exposed in software readable registers (providing a low-latency path for TSC access by a local processor). The TSC value comes in parallel format, normally from a TSC terminal node. The terminal node could be used by other blocks and is not instantiated inside the TKE itself.
- Local microsecond and OSC counters, also synchronized to the TKE clock. These counters are used for the operation of the timers and exposed in software readable registers. Contrary to the TSC, the microsecond and OSC counters in different TKE modules are not aligned to a common Main via hardware mechanisms.

- A programmable number of NV timers, nT, at maximum 16.
- A programmable number of watchdog timers, nW, at maximum four, when using shared interrupts.
- An optional interrupt routing block, with nSI shared interrupts.
- An APB interface

A TKE module exposes as outputs:

- A set of interrupts
  - The complete set of raw interrupts are always present. But only a smaller number, nDI, may be connected to an interrupt controller
  - A set of aggregated and routed interrupts when the interrupt routing block is present
- A set of WDT reset request interface signals, connected only when required.

TKE has the following configuration parameters:

- The number of NV timers, nT
- The number of watchdog timers, nW
- The number of directly connected interrupts, nDI, range[0, nT+3nW]
- The number of shared interrupts, nSI

Note: nSI > 0 implies TKE instantiates interrupt routing logic in the TKE common logic.

When nSI is > 0 and the interrupt routing logic is present, nT must be in the range [0,16] and nW in the range [0,4] to fit into the interrupt status register. If nSI == 0, the interrupt status register is not present and there is no a priori limit on the TKE configuration.

Only three different configurations (three flavors) are used across all instances, one for top, one for FSI and one for everything else. All flavors have nT an even number to allow for software consistency when needed. Each TKE also contains at least one WDT, so that the SoC contains a lot of WDTs.

The next table shows the characteristics of the different TKE instances.

Table 8.18 TKE Instances

Module	nT	nW	nDI	nSI	Remark
Top	16	3	7	16	System-level use cases, direct interrupts are from the WDTs only ▪ WDT0 IRQ/FIQ directly connected to CCPLEX GIC (bypassing LIC) ▪ WDT1 can be used as redundant to WDT0 or for software-defined use, IRQ/FIQ not connected ▪ WDT2 is the APE WDT for legacy reasons, IRQ/FIQ not connected ▪ One remote interrupt that aggregates the level 2 of all WDTs
FSI	4	5	15	0	Local timekeeping for the Cortex-R5 inside FSI and one WDT per CPU core present in FSI (four Cortex-R52 and one Cortex-R5). Note that FSI TKE does not use tsc_ref_clk but the local OSC clock.

Module	nT	nW	nDI	nSI	Remark
AON (aka SPE)	4	1	7	0	Local timekeeping, direct interrupts connected to local interrupt controller except remote WDT interrupt. PVA{0} WDT does not connect its reset request signals as PVA is considered a more deeply embedded processor whose failure does not warrant to ever automatically trigger a system reset.
SCE					
BPMP					
APE					
PVA{0}					
RCE					
DCE					

Software indicated a preference for only one WDT directly connected to CCPLEX as software treats the CCPLEX as a single coherent system. Software level techniques must be used to aggregate the status of cores currently managed by the coherent OS and only clear the WDT when all managed cores are operating correctly.

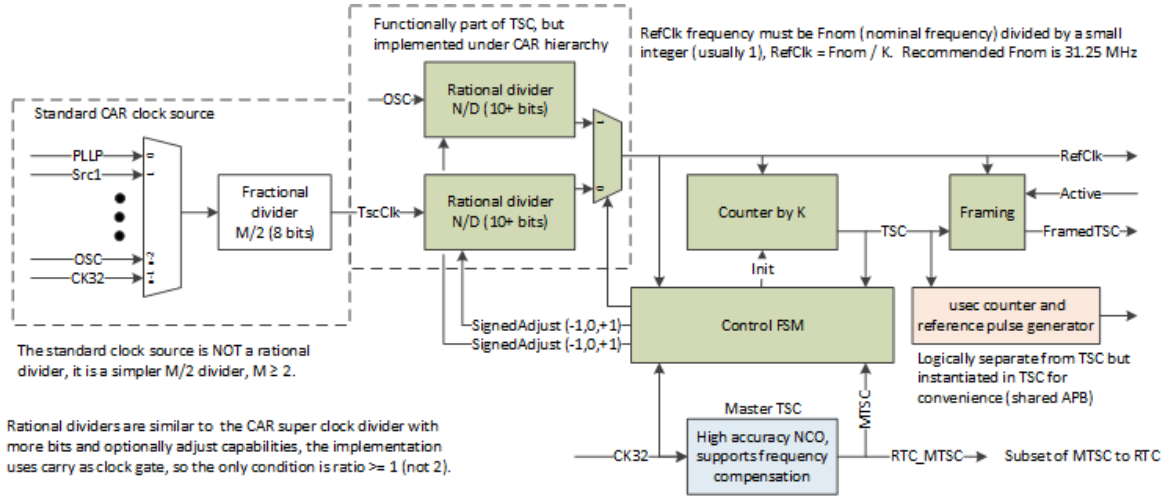
Care must be taken to make sure that the AON instance can still operate correctly when VDD_SOC is off and OSC is present, possibly only for a subset of internal configurations. All instances of TKE stop operating when the OSC is off and only the 32,768 Hz clock is active. This is true even for TKEs on the AO rail that remain powered.

The NV timers inside Top TKE are used as wake timers for the CPU cores in some CCPLEX low power states. Software uses one wake timer per CPU core, so the number of NV timers and interrupts is increased to at least 12 for Top TKE.

8.3.2.2 TSC

The TSC logic operates in the AO domain.

Figure 8.10 TSC Generation



A subset of MTSC is sent to RTC, allowing RTC to be optionally locked in frequency with TSC.

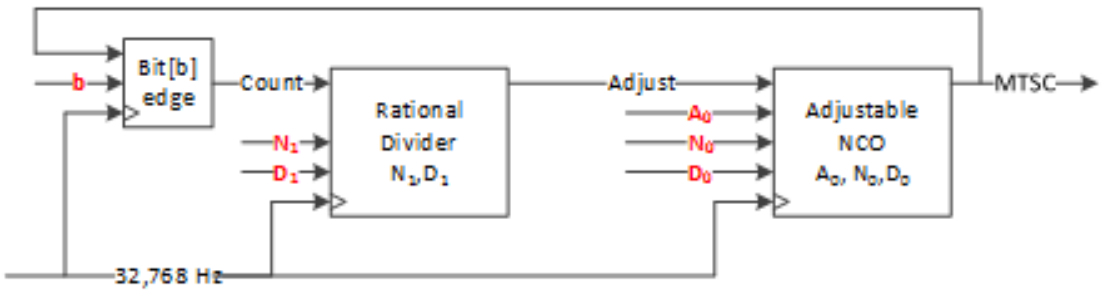
For convenience, the microsecond counter with associated reference pulses generation is instantiated inside the TSC logic, to share the same APB interface.

8.3.2.3 Main TSC Generation

The main TSC is a high accuracy NCO operating on the 32,768 Hz clock. It supports frequency error correction by allowing the main TSC operation to track an accurate external reference if wanted. As the external reference is not hardware visible at the main TSC boundary, the tracking loop needs to be done in software.

To support a very fine granularity for the frequency error correction, the main TSC uses a programmable bit of the MTSC as the count enable for the rational divider controlling the adjust signal, this is equivalent to scaling the divider by powers of 2.

Figure 8.11 Main TSC Generation



The main TSC's nominal frequency is programmable. The recommended value is 31.25 MHz so that the period (32 ns) is a power of 2.

The MTSC nominal frequency is given by the formula below

$$\frac{F_{out}}{F_{in}} = \frac{N_0}{D_0}\left(1 - \frac{1}{2^b}\frac{A_0}{D_0}\frac{N_1}{D_1}\right)^{-1} \approx \frac{N_0}{D_0}\left(1 + \frac{1}{2^b}\frac{A_0}{D_0}\frac{N_1}{D_1}\right) \; when \; \frac{1}{2^b}\frac{A_0}{D_0}\frac{N_1}{D_1} \ll 1$$

The selected bit of MTSC must be correctly sampled (no frequency aliasing), bit[b] of MTSC can be interpreted as a square wave with frequency (F_{out} / 2^{b+1}) sampled by the 32,768 Hz clock. Correct detection of the edges require the sampling frequency to be at least twice the frequency of the sampled square wave, so:

$$\frac{F_{out}}{2^b} \leq F_{in}$$

This can be solved for the value of b:

$$b \geq \left\lceil \log_2\left(\frac{F_{out}}{F_{in}}\right)\right\rceil$$

Using the recommended target frequency of 31.25 MHz and the 32,768 Hz input clock frequency gives:

$$b \geq \left\lceil \log_2\left(\frac{31{,}250{,}000}{32{,}768}\right)\right\rceil = 10$$

The supported range for b is [9,16].

The width of the parameters of the adjustable rational divider with some extra margin, giving 24 bits for N_0 and 12 bits for D_0, but the exact structure of the NCO needs to take into account the fact that the ratio is larger than 1. The corresponding structure splits the ratio into an integer and fractional part, that is:

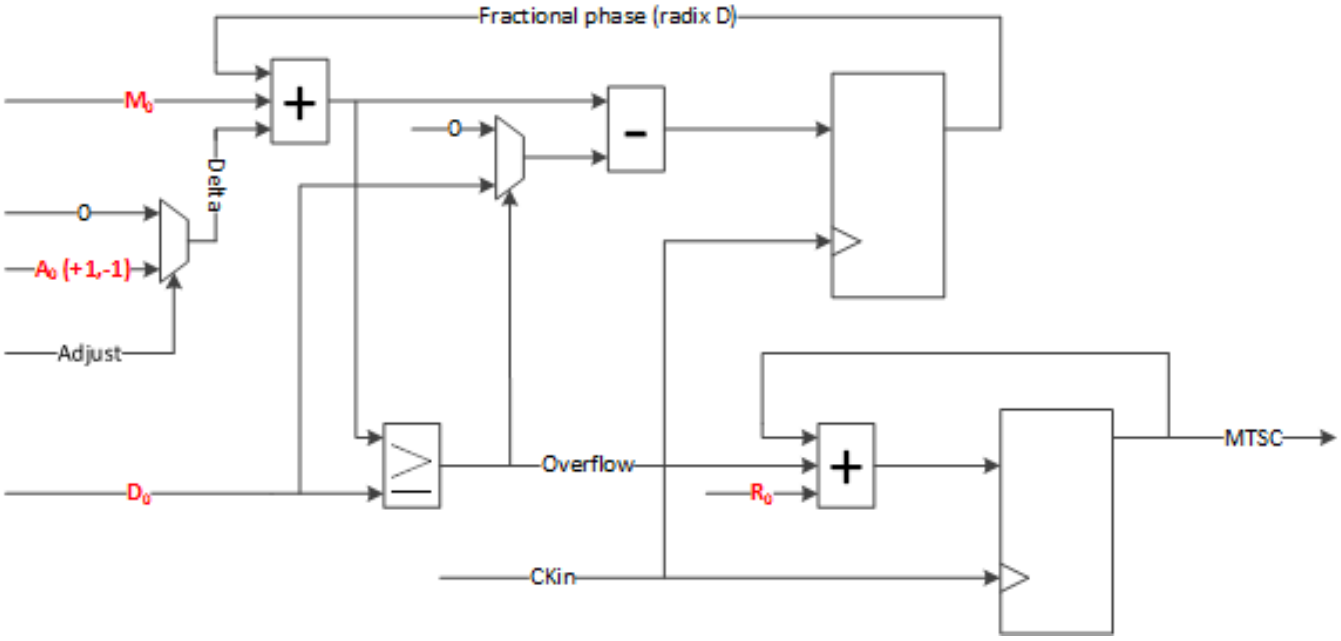
$$\frac{N_0}{D_0} = \left\lfloor \frac{N_0}{D_0} \right\rfloor + \frac{N_0 \bmod D_0}{D_0} = R_0 + \frac{M_0}{D_0}$$

The hardware directly uses R_0, D_0, and M_0, with software responsible to calculate their correct value.

The corresponding structure of the NCO is shown below, including the adjust capability, with a total of four software controlled parameters. The implementation further restricts A_0 to be +1 or -1,

A_0 is the direction of the frequency correction while the magnitude of the correction is controlled by the parameters of the rational divider controlling the adjust frequency: N_1, D_1, and b.

Figure 8.12 NCO Structure for MTSC Generation



8.3.2.4 Secondary TSC

Only OSC at 38.4 MHz with K = 1 is supported in the SoC.

The secondary TSC is essentially a rational divider with adjust capabilities followed by a counter. In the normal operation, the counter increments by one for each rising edge of the divided clock, similar to previous chips.

The secondary TSC always takes its initial value from the main TSC.

8.3.2.5 External Reference Tracking

The secondary TSC can track an external reference instead of tracking the main TSC, in particular, the external reference can be an external input, typically the GPS PPS signal or the PTP signal received on an Ethernet port.

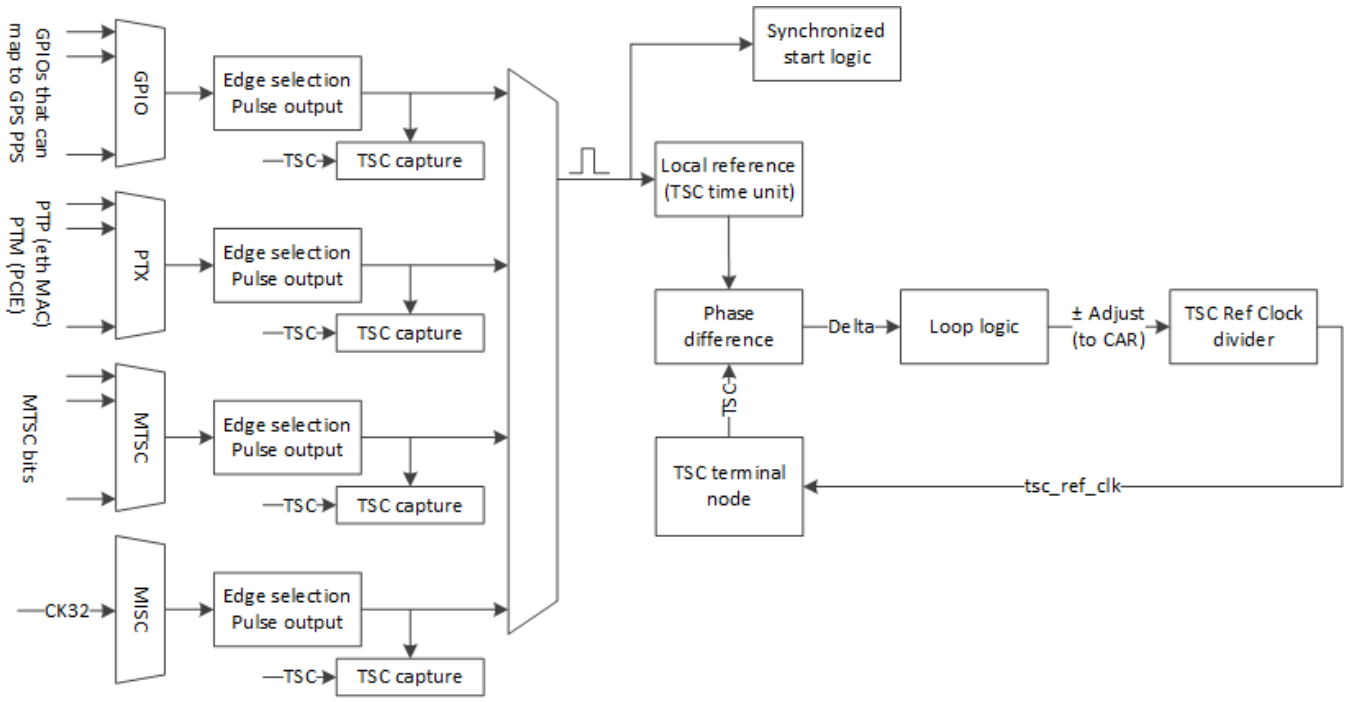
Because the time interval between edges of the reference becomes much larger, the approach for adjusting must be modified to provide a finer control of the frequency of adjust signals sent to the CAR fractional divider.

The approach is to use a fractional divided to generate the adjust frequency, with the hardware loop controlling the numerator. The denominator is sized to get a fine enough granularity based on the assumed maximum frequency drift rate to compensate.

The tracking can only be done on one selected signal, but the tracking logic also provides the current difference between STSC and multiple references, as shown in the figure below. Note that one of the paths uses bits of MTSC for reference and replaces the tracking mechanism found in previous devices.

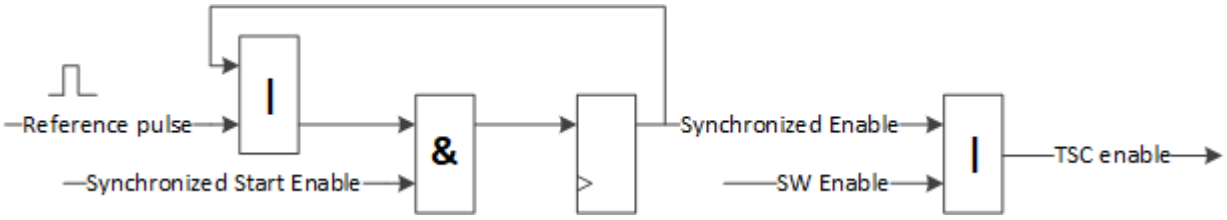
The hardware locking mechanism also allows to start the TSC counter synchronously with the external reference. This may be used as part of a hardware locked loop or to align the TSC value across different SoC devices. This is achieved by combining a hardware enable synchronized to the reference edge with the software controlled TSC enable bit.

Figure 8.13 TSC Capture and Hardware Tracking



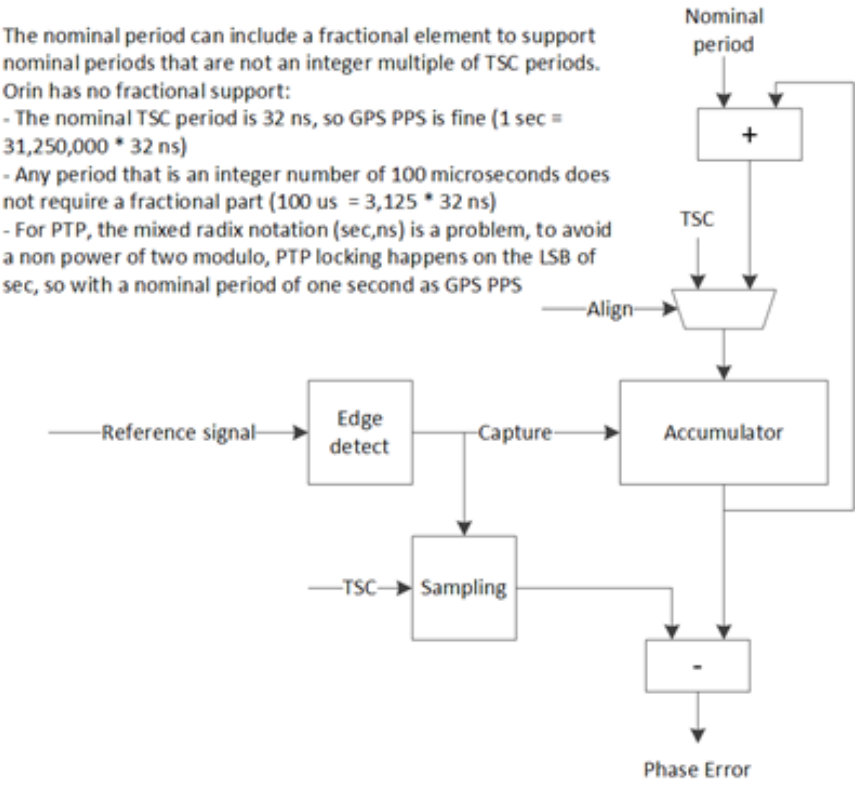
The synchronized start logic enables TSC counting at the first reference edge after software sets the synchronized start enable bit. The same source selection is used for both the hardware locking logic and the synchronized start feature.

Figure 8.14 Synchronized Start Logic



The phase difference measurement circuitry is shown below, it starts measuring at an arbitrary phase value, the current TSC value when the first edge after software sets the ALIGN bit.

Figure 8.15 Measuring Phase Error for hardware Locking



The following lists the signals that are used as reference signals, per branch:

- Branch GPS

  A selected subset of the input signals to the AO cluster GTE. It is assumed that one of these GPIO is the GPS PPS, normally carried as a short pulse, and the rising edge must be used as reference.

- Branch PTX

    - The LSB of the PTP second counter in each Ethernet MAC that supports PTP. The PTP standard requires to present the time as a mixed radix value of seconds and nanoseconds. This unfortunately means that bits of the nanoseconds counter are not all periodic, so each edge of the PTP second counter is used. This incidentally also matches the period (one second) of the PPS signal.
    - A defined counter bit of the PTM reference counter of each PCIe controller (a priori PTM capture is only needed for PCIe controllers that can carry Ethernet data). The PTM counter is not mixed radix so there are no restriction on which bit of the counter to select, the bit must be selected to provide a toggle frequency in the range 1 to 100 Hz in nominal conditions
- Branch MTSC

  A bit of MTSC, to support the legacy mechanism where TSC is locked on MTSC
- Branch MISC
    - CK32 (used as a data signal)

The loop logic controls the sign and frequency of the adjust signal in the following way:

- The frequency of the adjust signal is controlled by a rational divider, with the numerator of the divider being the magnitude of the variable controlled by hardware, while the denominator is fixed to 2. Fixing the denominator to a power of 2 simplifies the logic.
- The sign of the variable controlled by hardware (possibly inverted for test purposes) is the sign of the adjust signal sent to CAR

The adjustment of the numerator is done in this way

- If the magnitude of the phase difference is decreasing, do nothing
- If the magnitude of the phase difference is increasing, increase or decrease the numerator by a programmable delta based on the current sign of the phase difference.

The direction of the magnitude change is found by comparing two consecutive measurements of the phase difference.

The above approach is stable but converges slowly and is combined with a fast convergence mode that works by directly calculating change in the adjust values. This requires a more complex algorithm, that considers the value of consecutive differences. Note that the algorithm could also be done via software (possibly using more complex extrapolation algorithms).

The fast convergence operation is performed only when enabled and when the error or the extrapolated error is large:

- At each capture edge, calculate an extrapolated error that would be the error at the next capture edge if the adjust range was left unchanged, using the two last difference values measured $D_{extrapolated,i+1} = D_i + (D_i - D_{i-1}) = 2D_i - D_{i-1}$
- Calculates a delta to apply to the numerator as $D_{extrapolated,i+1} * K = dn$ if $abs(D_{extrapolated,i+1})$ is larger than a configured threshold. The new numerator $n_{i+1} = n_i + dn$ (with saturation). K is a (fractional) scale factor provided by software, the value is provided

as K = K_int * 2m with m configurable. To limit the complexity, K_int has not many bits as calculating the delta requires a multiplication followed by a left shift (the multiplication can be done serially bit per bit (speed is not important) or brute force parallel). The numerator must be saturated as the interface to CAR must maintain a distance of at least eight TSC cycles between adjust requests. The value dn is calculated to make the difference equal to 0 after one sample cycle.

The K factor is determined by calculating the change in number of generated TSC cycles for an increase of dn in the numerator, across one capture cycle.

- Number of nominal TSC cycles in one capture cycle, Ftsc * Tref = Nref
- Changes in number of adjust cycles requested in one capture cycle, Nref * dn / AdjustDen = NAdj
- Changes in the number of TSC cycles generated for adjust of A per request, A * NAdj / CarTscDen = NTsc

  Note: A is currently fixed at 1.
- dn is desired such that the extrapolated phase error after correction becomes 0, so
  - Nref * dn / AdjustDen / CarTscDen= NTsc
  - dn = (AdjustDen * CarTscDen) / (Ftsc * Tref) NTsc = K NTsc
  - K = (AdjustDen * CarTscDen) / (Ftsc * Tref)

The programmed value should be slightly smaller than the value calculated above, e.g., by a factor of 0.9 as the estimate is approximate, so finally Kint * 2^m = 0.9 * (AdjustDen * CarTscDen) / (Ftsc * Tref), with $2^{(w-1)}$ <= Kint < 2^w, where w is the bit width of Kint.

The fast convergence mode is normally much better but makes more assumptions and is more sensitive to misconfiguration up to becoming instable. The slow convergence makes almost no assumption, and is unconditionally stable if the step is small enough.

The algorithm requires one multiplication, Delta_NUM = -((Extrapolated * Kint) << m)). The variable Extrapolated represents the expected phase error in TSC unit and should never be large. Also, the operation of the algorithm does not require an extremely high precision of the results, so Extrapolated is saturated to 14 bits and Kint is using eight bits, possibly allowing for a shift and add multiplier if wanted. The dimensioning for Extrapolated assumes that the maximum absolute difference during fast convergence does not exceed twice the accumulated error during one reference period, i.e., is less than 2 * Ftsc * Tref * relative frequency error = 2 * 31.25 MHz * 1 s * 50 ppm = 3125. This requires 13 bits as Extrapolated is a signed quantity, the implementation has one extra bit for margin.

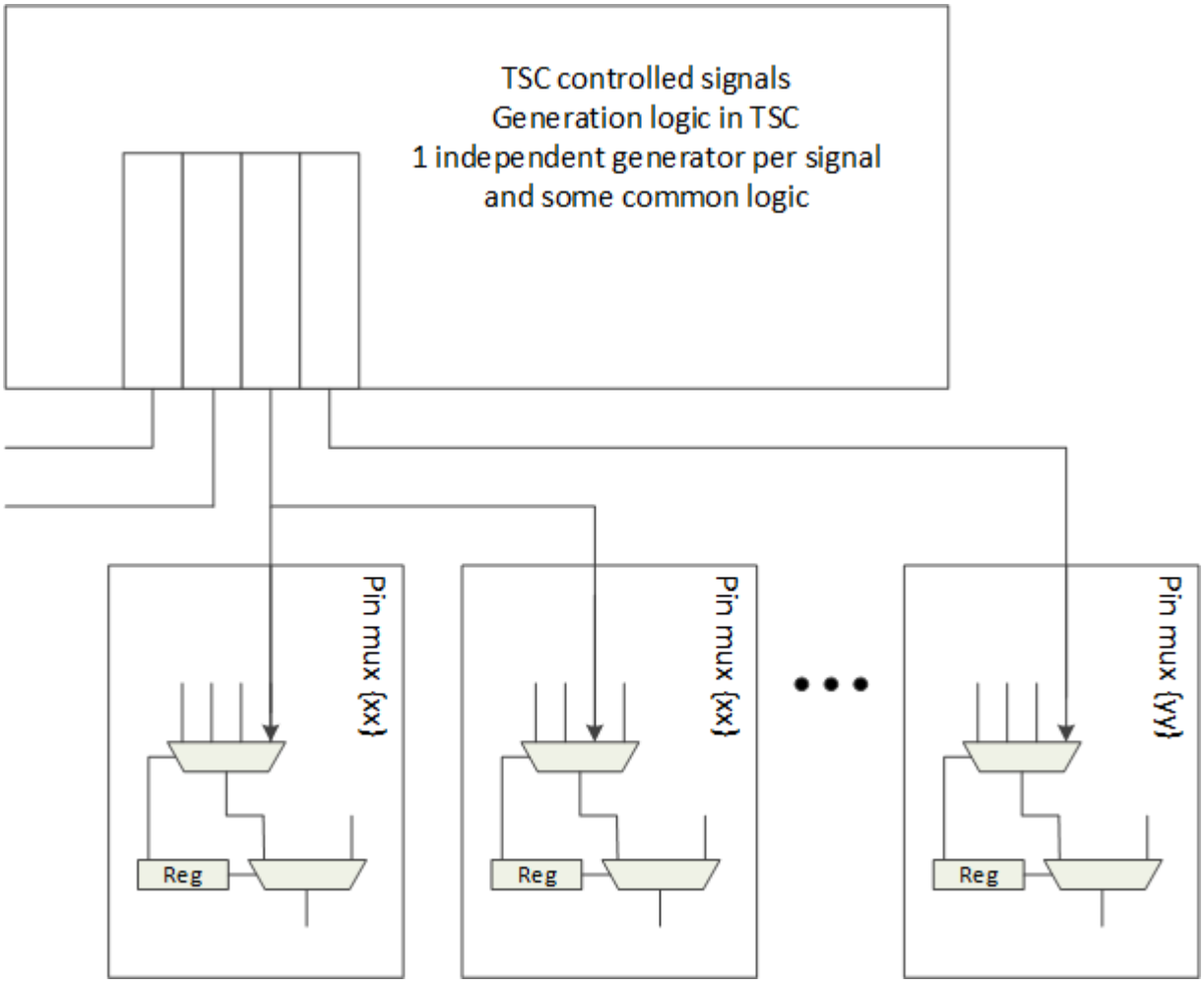
8.3.2.6 TSC Controlled Signals

TSC controlled signals extends the capability of placing interrupts at a known value of TSC and allows to place the edges of a limited number of signals at known values of TSC.

The canonical example for the need of such feature is controlling the time positions of a light flash relative to the shutter opening of a sensor. In general, this can help when synchronization of multiple external devices must happen at known time relative to the internal software time base.

In the SoC, the TSC block generates four independent signals, that are broadcast to a small number of pads. The target pads must use the standard pinmux logic, ideally with one available port or replacing an existing but not used port, as shown below. The first generator broadcasts its signal to four pins with stringent timing skew requirements, eliminating an external buffer. Other generators sent this signal to a single pin only.

Figure 8.16 TSC Controlled Signals



The register interface is defined to allow the generation of simple waveforms with minimal software complexity and make possible the generation of more complex sequences with more complex software:

- A generator always starts at a TSC value defined by a set of START registers. More than one register is needed because TSC values are 56 bits long.
  - The START registers also define the initial level
- A STATUS register allows software to follow the operation and to detect possible problem like the generator not starting because the START value is in the past.
  - WAITING, indicates that the generator has started and is waiting for TSC to reach the START TSC value
  - RUNNING, indicates that the generator is actively processing the EDGE registers
  - EDGE_ID, which EDGE register is processed by the generator
  - VALUE, current value present on the output
  - INTERRUPT, current interrupt level
- Once the generator is started, it changes the level based on a set of EDGE registers, each EDGE register defines
  - The OFFSET (in TSC unit) from the previous change at which to apply the actions defined by the register
  - A TOGGLE action bit, indicating if the signal must be toggled or not
  - A STOP action bit, indication if the generator must be stopped after this edge
  - A LOOP bit, indication that the next EDGE register to use should be back the first EDGE register, allowing to define a periodic pattern, STOP has precedence on LOOP.
    - If neither LOOP nor STOP are set, the next EDGE register is the next EDGE register (in circular fashion), i.e., (i + 1) % N, where N is the number of EDGE registers implemented.
  - A IRQ action bit, indicating if an interrupt must be reported to software after executing this step.

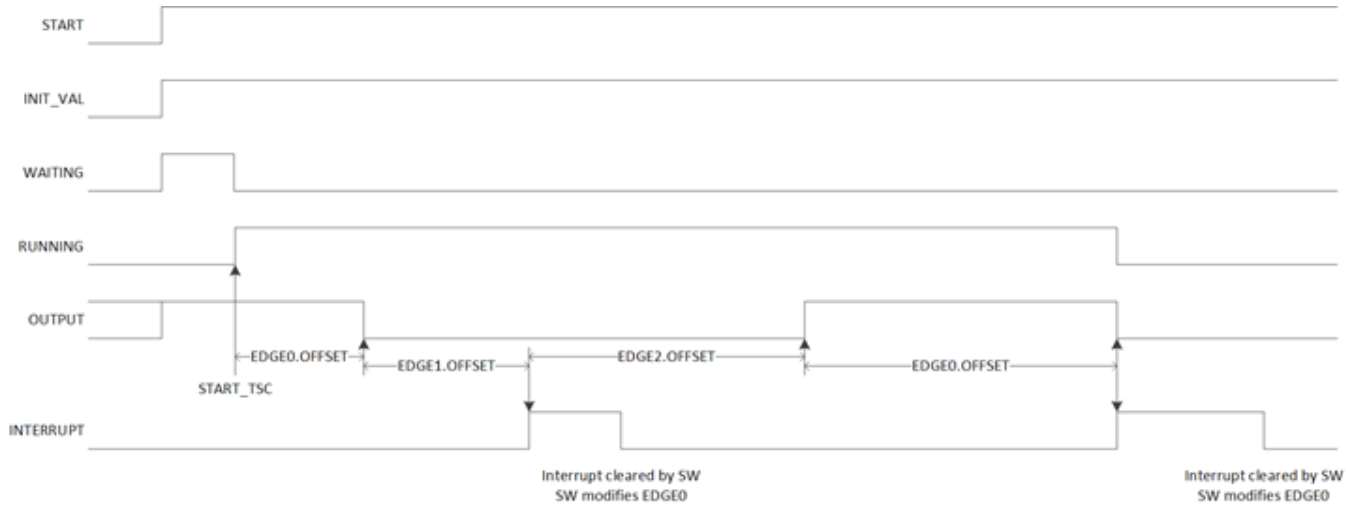
With two EDGE registers, the following waveforms can be generated with minimal software involvement:

- Single edge at a defined instant in the future, with optional interrupt
- Single pulse with defined width that starts at a defined instant in the future, with optional interrupt
- Periodic pulse with defined width and repetition period future that starts at a defined instant in the future, with optional interrupt.

Each generator implements eight EDGE registers, and the address space is structured to support easy extension to a much larger number. The structure of the EDGE registers allows them to be stored in a local memory if wanted.

Software can define arbitrary sequences if it can modify EDGE registers fast enough, possibly in response to an interrupt. The next figure is an example of use that includes such a dynamic modification.

Figure 8.17 Use of TSC Controlled Generation Example



index	LOOP	TOGGLE	STOP	IRQ
0	0	1	0	0
1	0	0	0	1
2	1	1	0	0
> 2	X	X	X	X

index	LOOP	TOGGLE	STOP	IRQ
0	0->X	1	0->1	0->1

8.3.2.7 TSC Distribution Logic

The TSC distribution logic operates by replicating the TSC value at terminal nodes. This is achieved by broadcasting both RefClk and serialized synchronization messages to all terminal nodes, plus a Halt indication used to stop TSC during debug. This is transparent to software except for the need to enable the distribution function.

8.3.2.8 Halting TSC During Debug

The Arm architecture supports halting TSC during debug, as a way to stop the timers dependent on TSC. This introduces problems in a complex SoC with multiple loosely coupled processors and TSC also used for timestamping operation.

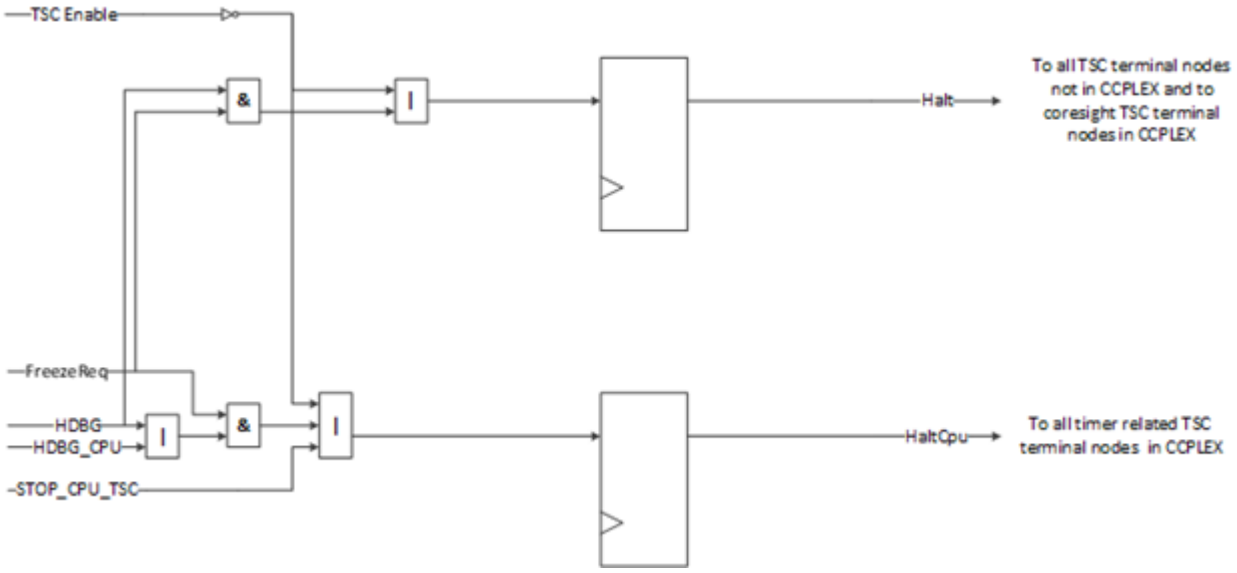
The following approaches tackle this issue:

- TSC terminal nodes support a third signal, Halt, so that they can be stopped without stopping the TSC reference clock. Two different Halt modes are supported, with the selection being done by a bit present in the SYNC message
  - The legacy mode is associated with value 0 of the selection bit (default), in that mode the TSC terminal node always honor SYNC messages when not Halted. This may trigger a jump at the end of a Halt period

    - The alternate mode, aka no jump mode, is associated with value 1 of the selection bit. In the no jump mode, the TSC terminal node enters a no jump state these two conditions are met
        - At least one SYNC message was honored since reset deassertion
        - Halt was asserted at least once since reset deassertion, with the no jump bit at 1

          In the no jump state, the TSC terminal node counts when Halt is deasserted, but does not synchronize to subsequent SYNC messages. The end result is that all TSC terminal nodes always increment at the same rate when they are not halted, and never experience any jump, but the TSC value can differ between different Halt domains.
- TSC and all TSC terminal nodes can be halted as before, using the existing HDBG configuration bit.
- The CCPLEX timers can be halted without stopping TSC itself by specifically stopping the TSC terminal node(s) in CCPLEX
    - All TSC terminal nodes in CCPLEX are hooked to a specific HaltCpu signal
    - A specific configuration to control the behavior: HDBG_CPU
    - The register interface includes a STOP_CPU_TSC field that operates independently of the debug mode as an alternate control method of stopping the CCPLEX TSC terminal nodes and thus, the CCPLEX timers.
- The TSC reference clock is never stopped
- The HALT signal is always asserted when TSC is disabled, otherwise the local time as shown by terminal nodes unexpectedly keep counting. This is not a useful use case but is nevertheless tested as part of Arm compliance testing.

Figure 8.18 Halt Generation Logic

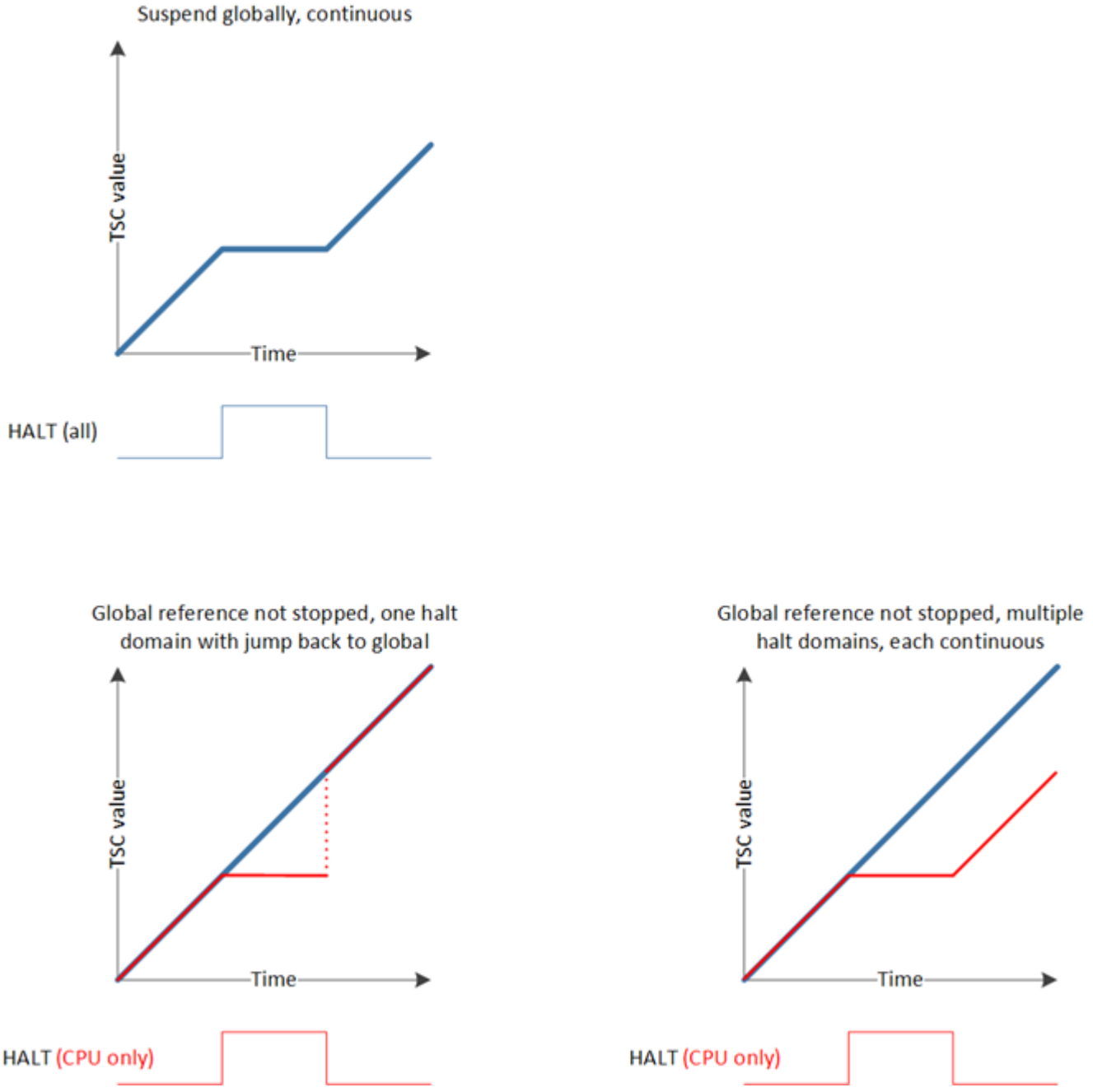


The TSC visible to the CPU via the Arm generic timers differs from the TSC visible to other engines when using the no jump halt mode after the first HALT_CPU only period.

Note: It is possible to define more independent controller Halt domains, but it is expected that only CCPLEX needs special handling, so only two modes of operations are defined, global Halt and CCPLEX only halt.

The behavior is illustrated below.

Figure 8.19 Halt Modes



The halt mode is in line with Arm recommendations on the use of TSC; Arm recommends two separate TSC, one which can be stopped during debug and one that never stops. Using the no jump mode with different HALT domains supports the Arm recommendation while insuring that any time

a TSC terminal node counts, it counts at the same rate as the global TSC. The following two halt domains are supported:

- The TSC terminal nodes feeding the Arm generic timers associated with the CPU cores inside CCPLEX. These terminal nodes are connected to the CPU halt domain, the corresponding DSU port is CNTVALUEB.
- All other TSC terminal nodes, including TSC terminal nodes inside CCPLEX used for program tracing, the corresponding DSU port is TSVALUEB[63:0].

8.3.2.9 Generic Timestamping Engine

The generic timestamping engine (GTE) is an IP block that can be instantiated where timestamping is performed by hardware instead of software. GTEs across the chip snoop a subset of signals:

- Interrupts, either close to LIC or close to AVIC inside a Cortex-R5 processor complex
- GPIO signals, especially for the sensor signals used by the SPE

The generic timestamping engine snoops a configurable set of signals, timestamps any change on them and presents the timestamped events to software via FIFOs and an APB interface. Care is taken to make the timestamping engine secure, i.e., to avoid security attacks through the use of timestamping.

The generic timestamping engine is built from a configurable number (nTS) of timestamping slices, each slice built from a configurable number (nC) of capture slices, each capture slice able to snoop up to 32 signals, and an arbitration stage when there are more than one capture slice per timestamping slice. Each timestamping slice is normally associated with a different Main in the SoC in particular, to provide differentiated security. All timestamping slices share a common APB interface.

The capture slice continuously monitors the set of enabled and not disabled inputs, if the current value is different from the last captured value and there is place in the capture FIFO:

- The captured value is updated to be the current value
- The current and last captured value is placed in the FIFO together with the slice ID and enough LSB of the TSC to allow a correct disambiguation of the timestamp later on.

If multiple capture slices are present, an arbitration stage merges the different capture slices into a common output FIFO.

Each output slice then presents the timestamping records to software via a set of access registers. Each timestamping slice has its own set of access registers to allow for differentiated security, but all timestamping slices share a common APB interface. Each timestamping slice can also generate an interrupt based on the FIFO occupancy; that interrupt may or not be connected to an interrupt controller.

The access registers can present the capture records either in raw form or as a set of individual bit changes.

8.3.2.9.1 Encoded Form for Captured Information

The raw information contains the current and previously captured value of the signals in the slice as a bitmap. Encoding transforms that into a sequence of changes. This is done by:

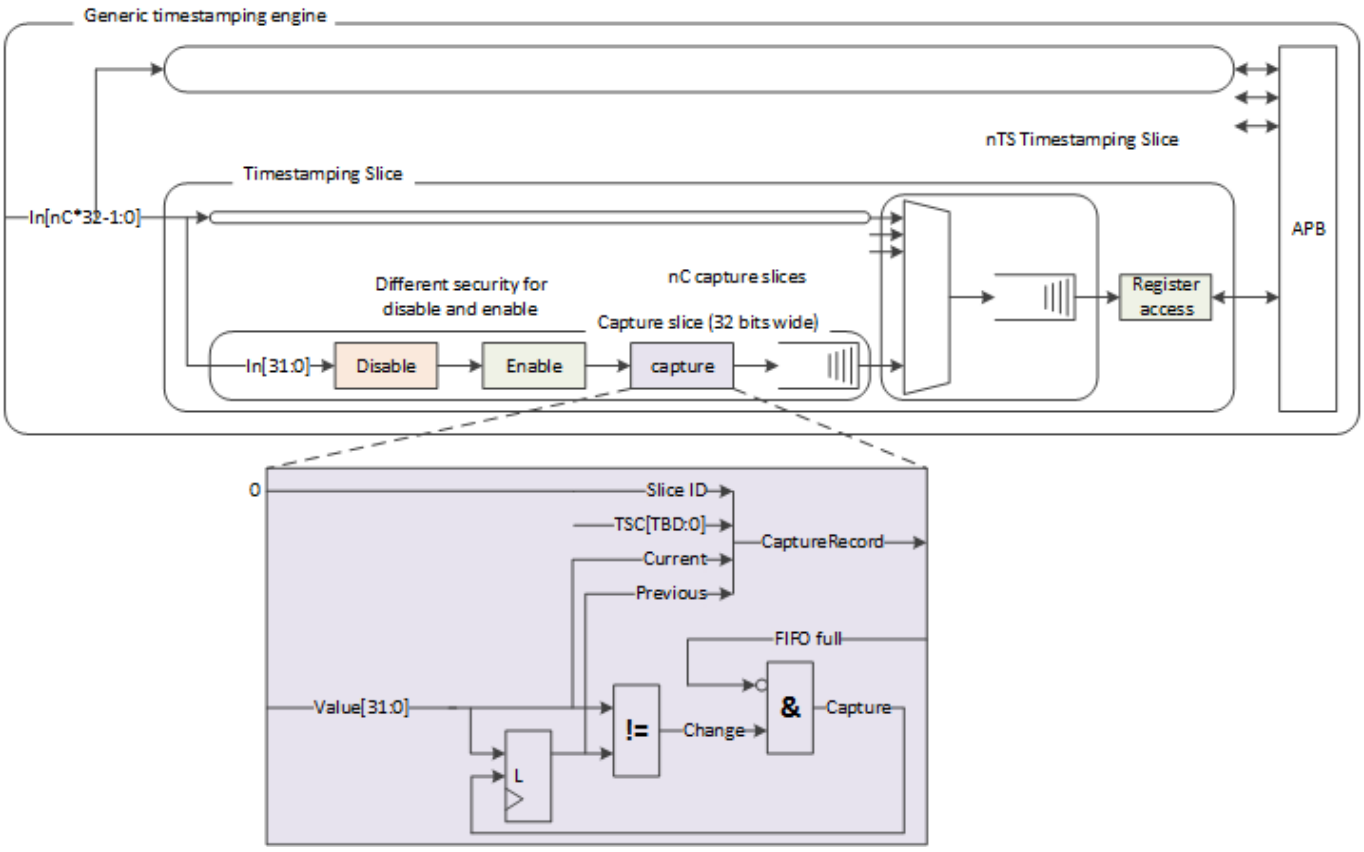
- XORing the previous and current values to calculate a change bitmap
- Iterating a procedure that:
- Finds the lowest index of a bit at 1 in the change bitmap, providing the index and the value of the bit in a register, plus the slice number.
- Clears the corresponding bit from the change bit map until the change bitmap is zero.

The index provides the full information about the signal captured; the register fields are defined so that the combination (slice number, bit index) gives the interrupt number.

8.3.2.9.2 General Structure of GTE

The general structure of the GTE is shown in the figure below.

Figure 8.20 Generic Timestamping Engine



There are multiple instances of the GTE. Each instance has a width defined by a number of 32-bit slices (nC). Some instances are paired to support multiple levels of security.

Table 8.19 GTE Instances

Module	Paired	nC	Remark
LIC	Yes	13+2	A pair of GTE for interrupt timestamping between LIC and GIC. Note that local GIC interrupts are not timestamped. Extra two slices for timestamping of top level peripherals or other top level signals.
BPMP	No	2	Timestamping the local interrupts, i.e., the inputs to the AVIC.
SCE	No		
PVA{0}	No		
RCE	No		
DCE	No		
AON	No	3	Timestamping the local interrupts, i.e., the inputs to the VIC, plus a small subset of GPIOs. The two lower slices are reserved for local interrupts, the third one for GPIOs.

In Orin, the following rules are followed to make the mapping between source signals and GTE input numbering more deterministic and stable:

- When timestamping interrupts, the GTE slice mapping is one to one with the corresponding interrupt slice, so that the interrupt numbering and GTE signal numbering are identical. In particular, reserved interrupt IDs become reserved GTE signal IDs and are not reused for other purpose, helping forward compatibility.
- Slices used for GPIO signals are mapped based on information contained in the GPIO configuration file; this indirectly also provides the mapping from GTE input number to a corresponding pin.

Note: All interrupt controllers are organized in slices, the GTE slice structure was patterned to reflect that approach.

8.3.2.10 RefClk Source Change Logic

The recommended usage model is to always use OSC (at 38.4 MHz) and with K=1 always.

8.3.2.11 Power Management

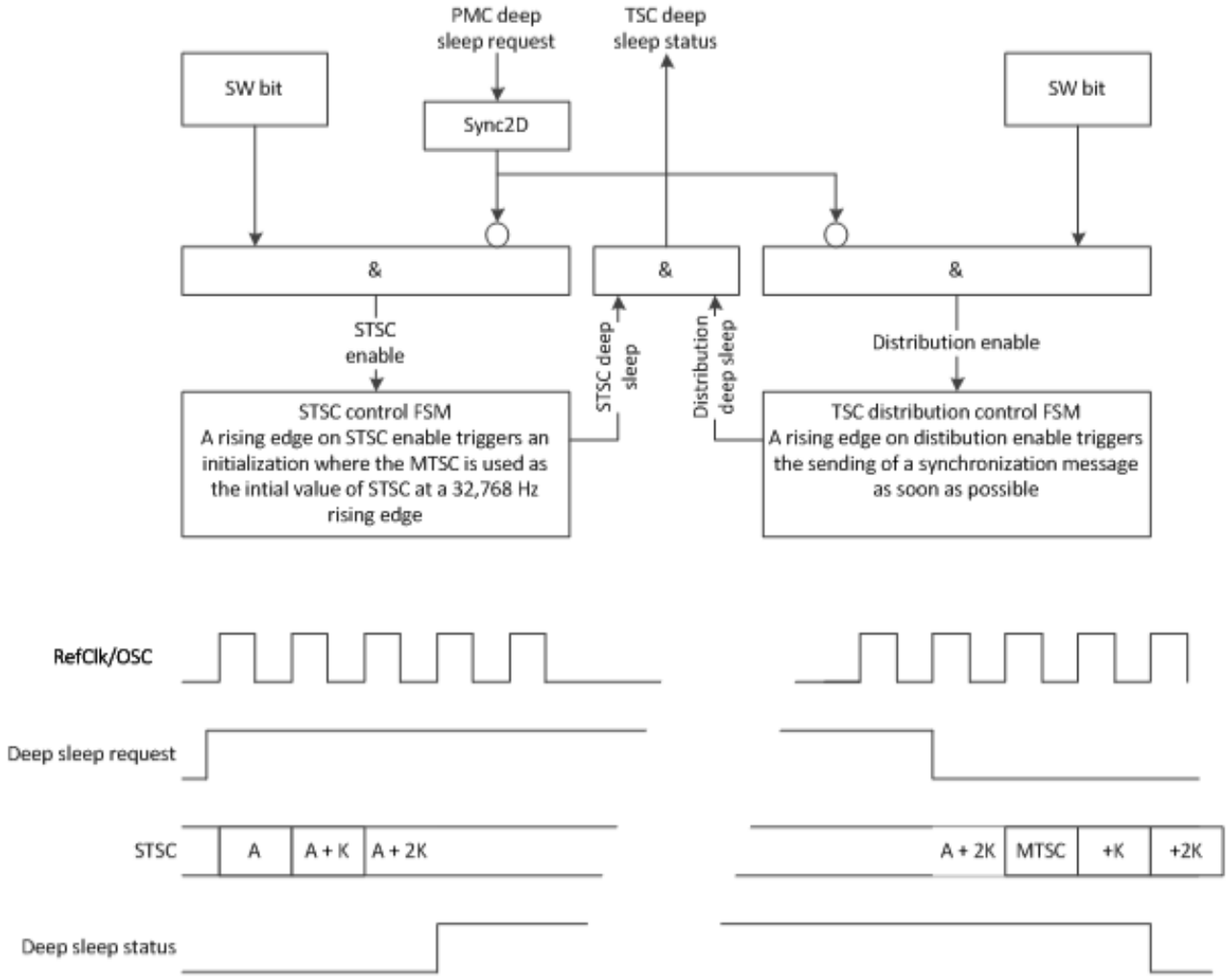
The TSC logic is not power gated but implements clock gating. Note that as it provides an SoC-wide timing reference, TSC is normally always enabled.

The secondary TSC and TSC distribution logic can not work when the OSC is disabled, as RefClk is not available then. OSC is disabled in some of the deep sleep states, with only 32,768 Hz present. This is true for some variants of SC7. The transitions to and from the deep sleep states that disable the OSC circuitry are controlled by the PMC logic. The PMC logic handshakes with TSC, to allow for a clean shutdown and restart of the secondary TSC as well as TSC distribution in case OSC is stopped in the target power state.

- Entering a deep sleep mode where OSC is to be stopped
  - PMC asserts a deep sleep request signal while OSC and RefClk are still running
  - TSC stops the secondary TSC and the distribution logic, then assert a deep sleep status signal to PMC
    Stopping is logically equivalent to disabling these blocks via the software control bits
  - PMC can now stop OSC and indirectly stops RefClk
- Exiting a deep sleep mode where OSC already stopped
  - PMC restarts OSC and indirectly restarts RefClk, and waits until OSC is stable
  - PMC deasserts the deep sleep request signal
  - TSC restarts the secondary TSC and the TSC distribution logic if their respective software enable bits is set
  - Restarting the secondary TSC requires an initialization, the main TSC value is copied to the secondary TSC counter at the next edge of the 32,768 Hz clock
  - Restarting the TSC distribution preferably includes sending a synchronization message as soon as possible
  - TSC deasserts the deep sleep status signal to PMC

The whole sequence is similar to the sequence used for Arm processors. The handshake must also work even when TSC itself is disabled, to avoid blocking the forward progress of the PMC state machine. This may require providing a non-gateable clock to the handshake logic.

Figure 8.21 Deep Sleep Handshake with PMC



8.3.2.12 Real Time Clock

The internal Real Time Clock (RTC) is a real-time clock in the AO partition that measures elapsed time since the last cold boot and is not affected by SC7 entry/exit transitions. Note that the PMIC also generally provides an external RTC with better robustness (not affected by the cold boot of the SoC).

The RTC accumulates real time in a mixed radix format of seconds (32 bits) and milliseconds (10 bits), it operates on the 32,768 Hz clock and can take its reference from the 32,768 Hz clock itself or TSC.

The seconds counter can be updated by software to provide an initial offset different from zero at initialization, and later updates sticky disabled for security.

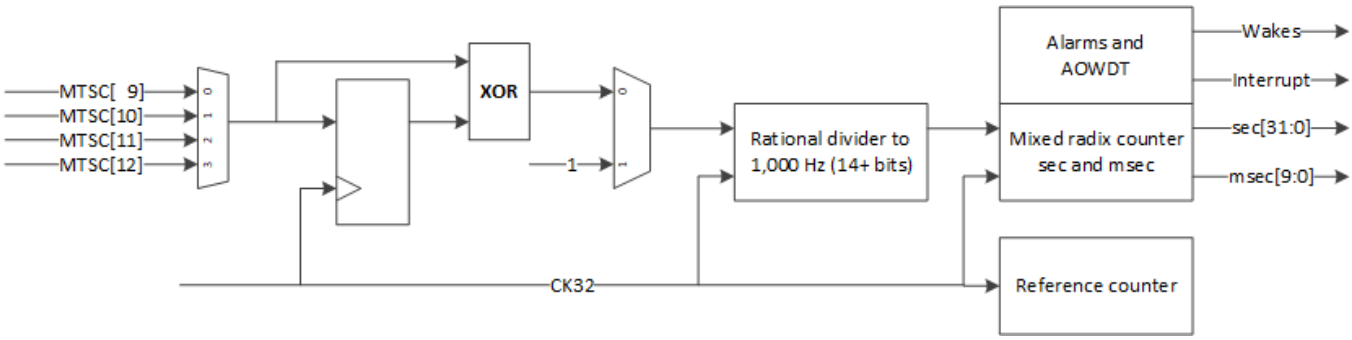
The RTC provides different alarms that can be used as interrupts or SC7 wake signals. The RTC also provides a safety reference counter, with no configuration whatsoever that can be used to check the correct operating frequency of other TKEs.

RTC logic operates in the 32,768 Hz clock domain, and software must take into account when accessing RTC registers. In particular, the RTC module maintains a 'busy' status register bit which is set (1'b1) while a register write is being transferred. Software must not send a write when this bit is set (this requires write accesses to the RTC to be protected by a resource semaphore if more than two agents are allowed to access the RTC).

The RTC maintains a count of seconds and milliseconds since the last cold reset. It can select between the 32,768 Hz clock and MTSC as time base. The only difference is the first NCO stage, the rest of the logic is independent of the selected reference.

The logic itself always operates at the 32,768 Hz clock.

Figure 8.22 RTC Generation



The RTC block also contains alarms implemented by comparison between a target value and the current RTC value. The RTC value is exported to SPE where it is used to generate local alarms on top of the alarms supported in RTC itself.

For safety related purposes, the RTC also provides a reference counter, always enabled, that increments at each rising edge of the 32,768 Hz clock. This can be used as a safety reference to assess the health of other timing elements. This reference counter is never halted, especially it is not stopped during debug and has no configuration.

RTC is not configurable but is instanced inside the AO partition, the same as in previous devices. It also contains the AOWDT.

8.3.2.13 Always On WatchDog Timer

A watchdog timer (WDT) is provided in the Always On (AO) power domain, with the following rationale:

- Its operation is not dependent on OSC; it only requires the 32,768 Hz clock and the main TSC to be active.
- The timer can be used to detect and flag SC7 related issues, such as an inability to wake up.

Using a different clock source and voltage domain for this time can be used for safety diversity.

The AO WDT requirements are only slightly different from a standard WDT:

- Operates on the 32,768 Hz clock, so that it can operate continuously, including during power states where OSC is OFF when WDT operation is suspended.
- The actions associated with some expiration levels are redefined as wake events:
  - Expiration level 2 asserts a standard wake signal.
  - Expiration level 3 asserts a wake signal that also clears the PMC scratch register info indicating warm boot.
- The debug reset associated with expiration level 4, while connected, must not be used.

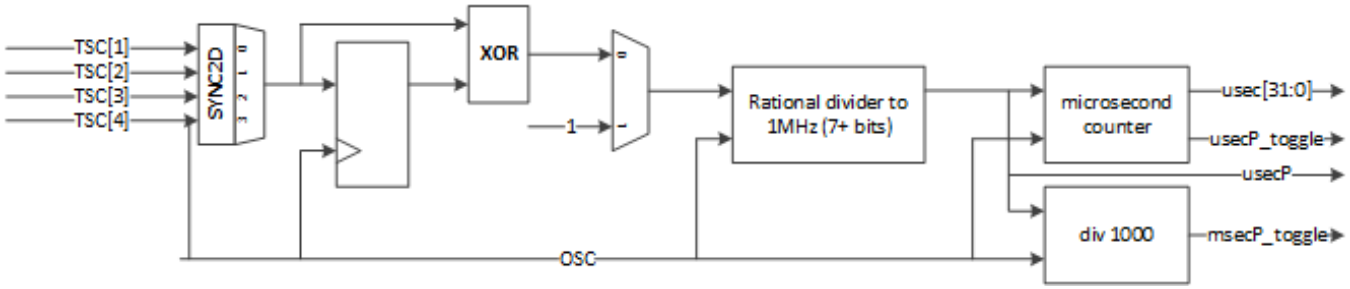
The AO WDT reuses the implementation of the TKE WDT, including the register interface; the only changes are in the inputs and outputs connected to it, especially the operating clock.

8.3.2.14 Microsecond Counter and Time Reference Pulses Generation

The USEC module broadcasts periodic pulses with 1 microsecond and 1 millisecond periods, allowing time to be measured in a platform-independent way. USEC operates on OSC to allow operation when no PLL is active. The time base can be OSC itself or TSC, and USEC also maintains a counter of elapsed microseconds.

While USEC operates in the AO partition, it relies on OSC for its operation and so may be suspended in some power states where RTC keeps counting, and so USEC and RTC are complementary, not redundant.

Figure 8.23 Reference Pulses Generation and Microsecond Counter



8.3.2.15 TKE Module

NV Timers (TMRs) and Watchdog Timers (WDTs) are instantiated as part of a TKE module. There are nT timers and nW watchdog timers per TKE. Each TKE also includes registers to make different measures of time available: OSC, microsecond counter, and TSC. This access is low latency to the local processor. Finally, each TKE can optionally contain an interrupt routing function, this is only present for the TKE instantiated at the top level to limit the number of top-level interrupts.

For debug purposes, the TKE operation can optionally be halted during debug. This is especially important to suspend the operation of enabled WDTs that could otherwise reset the system while debug is in progress.

8.3.2.15.1 NV Timer

Each NV timer is a general-purpose timer with the following characteristics:

- Downcounter of 29 bits generating an interrupt when reaching zero
- The decrement rate is on a configurable reference:
- At 1 MHz, as defined by the 1 MHz reference pulse
- At OSC rate
- On any edge of TSC[0], so normally at the recommended 31.25 MHz TSC update rate
- On any edge of TSC[12], so normally at 31.25 MHz / 4096, about 7.63 kHz
- Optionally operates in periodic mode, i.e., automatically reload the start value when reaching zero

The ability to select the timebase for each timer independently may be of assistance in functional safety applications.

8.3.2.15.2 Optional Interrupt Routing

The interrupt routing block takes the set of interrupts present in the TKE and aggregates them in a configurable fashion into nSI shared interrupts. This is only present for the top-level TKE and the shared interrupts are connected to the LIC in that case.

The number of (internal) interrupts inside a TKE is dependent on the number of NV timers and WDT:

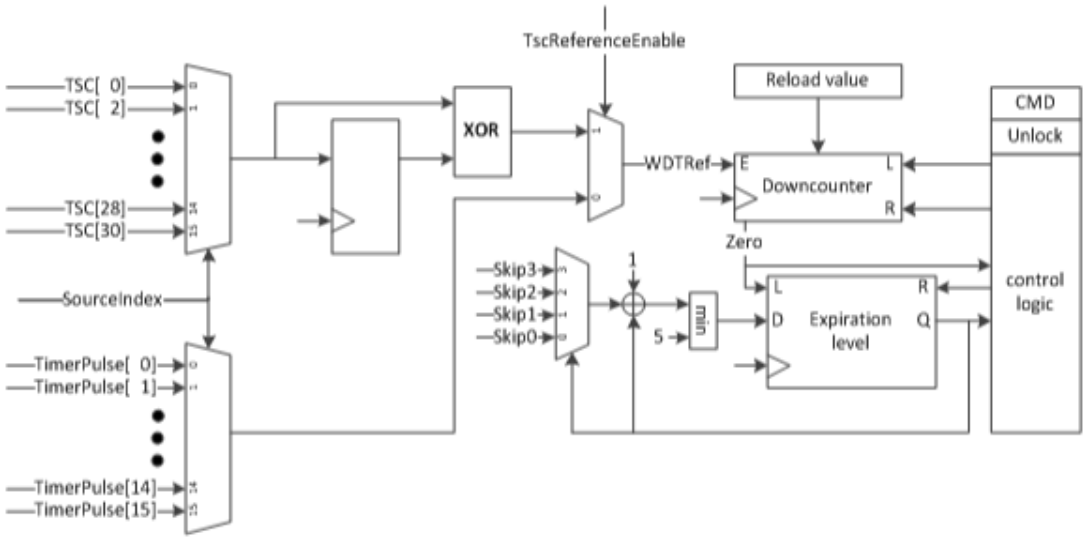
- Each NV timer has one associated interrupt, set when the downcounter reaches 0
- Each WDT has three associated interrupts, one for each of the three first expirations of the WDT

The interrupt logic uses nSI masks, each containing (nT + 3 * nW) bits. Each mask selects which local interrupt are included in the corresponding shared interrupt. The interrupt logic also exposes interrupt status registers to software.

8.3.2.15.3 Watchdog Timer

A general view of the WDT logic is shown below.

Figure 8.24 WDT Structure



The watchdog mechanism operates in the following way:

- Each processor complex is associated with one WDT. For the CCPLEX, all cores are in the same coherency domain and are considered as one SMP processor for software purposes; i.e., all cores inside CCPLEX are associated with the same WDT (WDT0 in the Top TKE).
- Select as reference time base either:
  - Any of the timer zero pulses
  - A transition on any bit in a defined subset of the bits of the local TSC counter

Having more than one reference source may be used for the safety requirements for diversity.

- Implements a periodic downcounter decrementing at the selected rate. The downcounter generates an expiration event when it reaches zero.
- At each expiration, the WDT increases the expiration level, saturating at level 5
- The increase may be larger than 1, the exact increase is configurable per expiration level
- The WDT logic can assert specific output signals based on the current expiration
  - For the first three expiration levels, the WDT can assert a corresponding interrupt, each with a specific use:

    - If the expiration level is 1 or more, a normal priority interrupt (IRQ) directly routed to the associated processor complex. This may be used to restart the timer inside an interrupt handler in non-safety critical applications.
    - If the expiration level is 2 or more, a high priority interrupt (FIQ) directly routed to the associated processor complex. In Android, the corresponding interrupt handler assumes the system is in a bad state and captures some state information for a potential post mortem analysis. The interrupt handler may not attempt to correct the problem.
    - If the expiration level is 3 or more, a normal interrupt (IRQ) routed to the LIC and from there routed to a set of processors, this is meant to trigger system wide actions, in particular recovery actions of the processor associated with the WDT. This is also known as a remote interrupt.
- If the expiration level transitions from a value strictly below 4 to a value above or equal to 4, the WDT can assert a system wide debug reset. When debug is allowed, this reset can be intercepted by debug logic, and specific actions taken to preserve information about the state of the system before reset assertion clears them, allowing for post-mortem debug analysis. In particular, DFD logic attempts to flush the processor caches and local memories to DRAM, save processor state, and put the DRAM in self-refresh before allowing the reset to take effect. WDT and CAR logic uses a four phase handshake to communicate the WDT debug reset request.
- If the expiration level is 5, the WDT can assert a system wide reset, equivalent to a Power On Reset, this reset can optionally include a PMIC power cycling depending on the configuration in PMC. This is the last safety net for recovery purposes, attempting to bring the system to a working state without any specific attempt at preserving information.
- Each WDT signals an error to the Hardware Safety Manager (HSM) when its expiration level is greater or equal than a programmable threshold. Orin supports a force error command for testing; software must not issue any restart command while error forcing takes place.

The watchdog timer supports a configurable restart mechanism. These features are disabled by default for backwards compatibility.

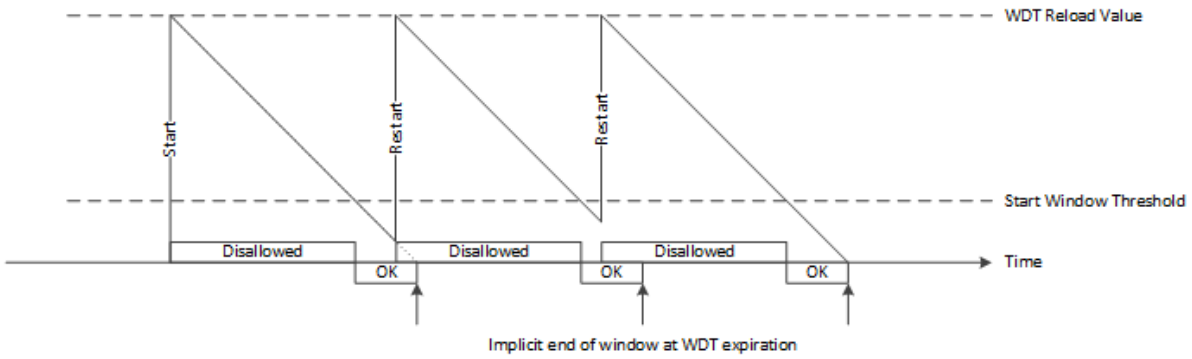
- The counter can only be optionally restarted in defined windows of time instead of at any time
    - Two window mechanisms with different granularity are supported, with coarse and fine granularity.
    - The window mechanism is a safety requirement.
- The counter can only be optionally restarted or disabled using a Challenge/Response pair (also known as Question/Answer, QA). In previous chips, only disabling required an unlock using a simpler fixed pattern unlock. The challenge/response mechanism is a safety requirement to avoid unexpected deactivation of a WDT.

The windowing operation works like this:

- Optionally enabled, normal operation is windowless and the WDT can be restarted at any time.
- When enabled, a restart of the timer is disabled:

- For a programmable set of expiration levels. This is achieved by disabling writes to the COMMAND register based on a bitmap of disallowed expiration levels.
- When outside an allowed window as measured by the combination of expiration level and downcounter, as shown in the figure below.

Figure 8.25 Fine-Grained WDT Windowing Mechanism



The Challenge/Response works like this:

- Optionally enabled, normal operation does not use Challenge/Response, unlocking is done by writing using a known fixed pattern to the unlock register address.
- When enabled, a restart is achieved only after unlocking via a Challenge/Response.
- The Challenge is that a Linear Feedback Shift Register (LFSR) value, present in the Unlock register and reading the Unlock register, is mandatory as the implementation is allowed to use the read to alter internal state.
- The Response is the next value of the LFSR that needs to be written in the Unlock register.
- The LFSR is (re)initialized with the value written to the Unlock register.
- Note that this is equivalent to advancing the LFSR if the unlock was successful.
- 0 is an allowed value that can be used for testing, the successor of 0 is 0.
- The LFSR uses the classic CRC-32 polynomial operating in shift right (reversed) direction; that is, the next value is calculated as LFSR = (LFSR >> 1) ^ ((LFSR & 0x1) ? 0xEDB88320 : 0).
- When locked, the Command register cannot be written.

An incorrect restart attempt can optionally result in an immediate jump in the expiration level. This is normally used to immediately trigger a reset condition when the WDT is used as a safety-relevant WDT.

For safety, the WDT enable is controlled by two bits, normal and sticky, and the platform policy can decide which to use, i.e., if a WDT may or not be disabled once enabled. The use of the sticky enable, making it impossible to disable the WDT until the next reset, is a safety requirement.

There is a watchdog timer in the Always On partition, that AO WDT is only reset by an external hardware Power On Reset, i.e., it is never disabled by a reset triggered by an SoC WD timer. The AO

WDT has the same structure as the SoC WDTs but with different wiring of some inputs and outputs.

WDTs can optionally be halted during debug; WDT operation may also be suspended when one or more of the TKE clocks is suspended. This can happen for a standard WDT during some low-power states where OSC is stopped; for example, AO WDT is specifically present to continue working during these power states.

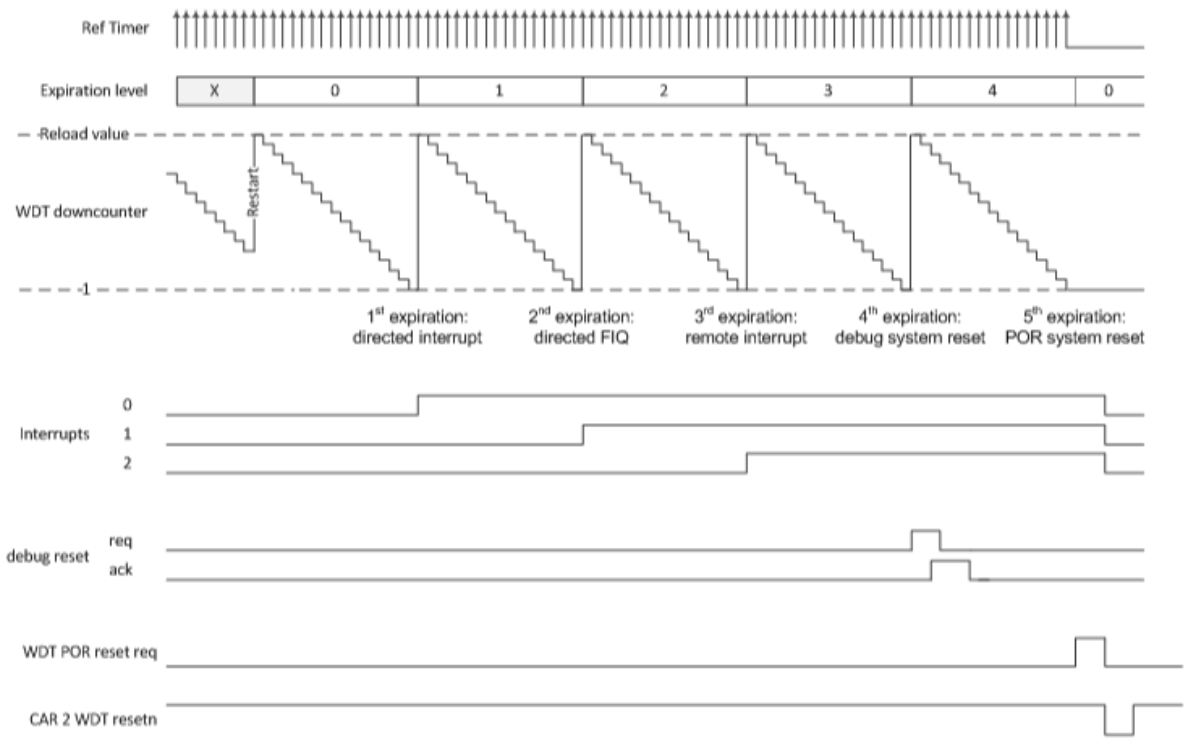
The SoC supports a modified mechanism to stop all WDTs, similar to the ability to stop all timers. The request comes on a wire broadcast to all TKEs (and AOWDTs). The request is further qualified by the local debug enable signal. Each WDT adds a configuration bit to allow/disallow stopping the WDT, allowed after reset but can be switched to disallow for safety use cases. Each WDT also includes a status bit indicating if it is currently frozen.

The local debug enable signal is connected to the invasive debug enable for the processor cluster associated with the TKE:

- CCPLEX DBGEN signal for top TKE: global_dbgen
- Cortex-R5 DBGEN for a Cortex-R5 cluster: {bpmp, spe, sce, rce, dce, pva{0}}_dbgen
- A9 DBGEN for APE cluster: ape_dbgen
- DBGEN connected to SPE for AOWDT: spe_dbgen

Note that this mechanism makes the use of the configuration bit disabling freeze optional in safety applications. It is recommended to leave that bit at 0b.

Figure 8.26 Example Timeline for Watchdog Operation



The WDT timers are reset on L2 reset except the AOWDT that is reset on L1 only.

The reset requests are individually exposed as ports, even when multiple WDT share the same TKE as the reset source must be captured for use by software.

8.3.2.16 Generic Timers

Arm Generic Timer (GT) architecture defines timers to be used by Arm CPUs. This architecture uses the system wide timing reference called Timestamp System Counter (TSC) that increases at a constant rate. This allows the Generic Timers inside a CPU cluster to operate correctly without reconfiguration when the CPU frequency changes.

The Arm Generic Timer includes the following features:

- Physical counter that contains the count value of the system-counter.
- Virtual counter that indicates virtual time. The virtual counter contains the value of the physical counter minus a 64-bit virtual offset.
- Multiple timers per CPU (the exact set depends on the core configuration):
  - EL1 physical timer
  - Non-secure EL2 physical timer
  - EL3 physical timer

- EL1 virtual timer
- Non-secure EL2 virtual timer
- Secure EL2 virtual timer
- Secure EL2 physical timer

The Arm architectural specification includes a mandatory (CNTCTLBase) registers frame that provides configuration information related to Arm generic timers implemented in MMIO fashion; The SoC does not currently have these registers (all Arm generic timers present use system registers).

Full details about the operation of the Generic Timers were introduced in different Arm documents and are now part of the *Arm Architecture Reference Manual*.

8.3.2.17 Cortex-A9 Timers

The Cortex-A9 complex inside the APE contains an instance of a Cortex-A9 timer block, including the following:

- Global timer, a 64-bit incrementing counter
- Per CPU core private timer, a 32-bit decrementing counter with optional prescaler
- Per CPU core watchdog timer. It has the same structure as a private timer augmented with watchdog capabilities and can be used as a second private timer. The Cortex-A9 WDT reset request signal is not connected.

All these timers operate on the Cortex-A9 clock, making them difficult to use for real-time purposes when the Cortex-A9 clock is subject to DVFS. The APE cluster includes a TKE (not affected by DVFS) and APE software usually do not use at all the Cortex-A9 timers.

8.3.2.18 Functional Safety Island Timers

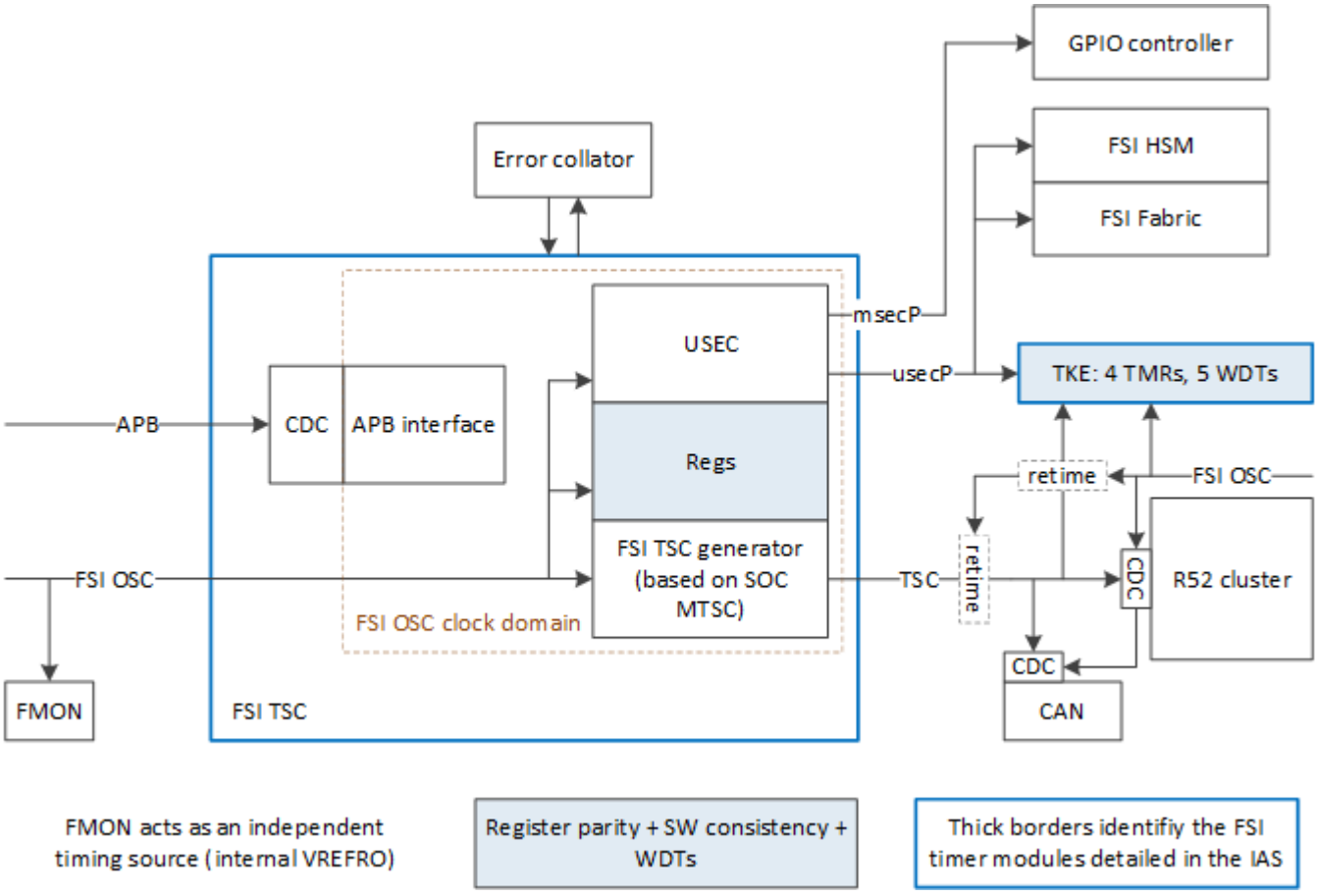
The Functional Safety Island (FSI) takes over most of the functions of the external safety MCU.

The FSI requires two timer modules:

- A global system counter, matching the Arm generic timer architecture; this is done reusing some of the TSC building blocks.
- A TKE that provides timers for the Cortex-R5 (similar to Cortex-R5 clusters on the SoC side) or Cortex-R52 (generic Arm timers part of the Cortex-R52 cluster itself). In addition, at least one WDT per CPU core (five WDTs: four Cortex-R52 and one Cortex-R5), as each core could run as an independent software entity.

To limit complexity, all timer elements run on the FSI local crystal clock, this simplifies the CAR logic and remove the need for TSC terminal nodes, as explained later on. The overall timer infrastructure logic inside FSI is illustrated below.

Figure 8.27 FSI Timer Infrastructure



The FSI module is relatively small, so TSC distribution does not need to cover large distances and the TSC value is only distributed to very few endpoints, so TSC is distributed in parallel binary format, with retime stages if needed.

The FSI operates at ASIL-D, this is achieved by using software consistency to check the different timer operations, register parity for all configuration registers and hardware watchdog timers.

8.3.2.18.1 FSI-TSC

The FSI-TSC implements a simplified TSC that performs the following functions:

- Supports the Arm architecture registers, except for TrustZone specific requirements

  Note: The Arm architecture requirements define some registers as TrustZone accessible only; this is not possible for R class Arm CPU cores that do not support TrustZone.

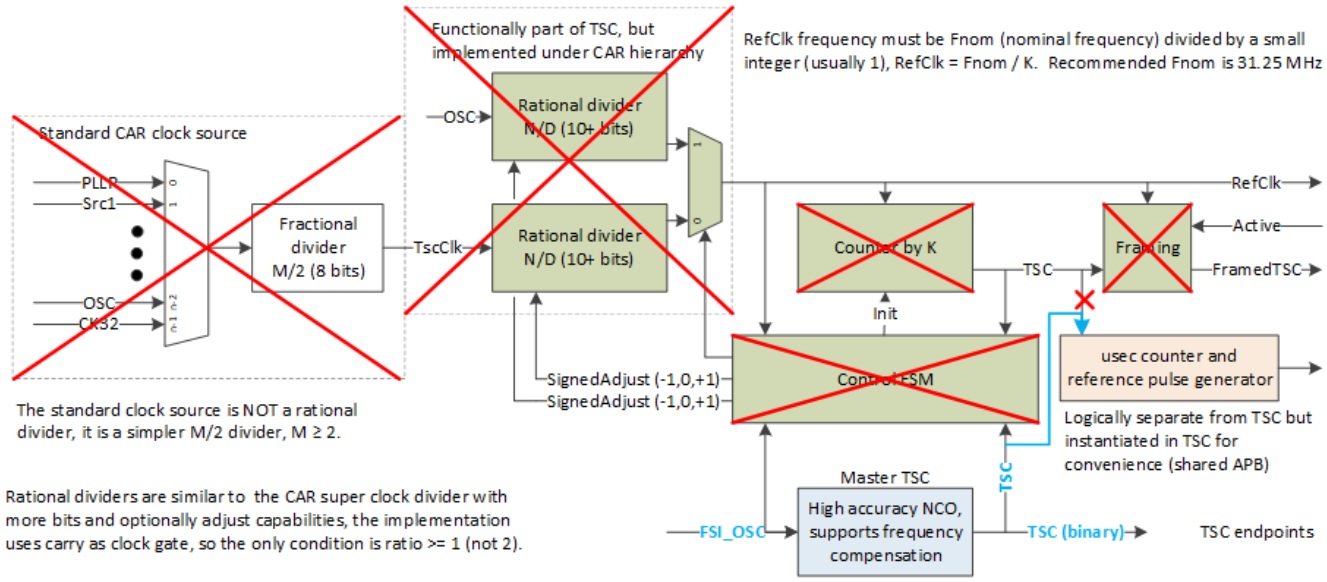
- Maintains a 56 bit counter, in steps of 32 ns. This is nominally the same rate as the SoC side TSC.

- Distribute the 56 bit counter in parallel binary format to the rest of the FSI logic, especially the local TKE, the CAN modules and the Cortex-R52 cores
- Also implements the USEC related counter and signals and sends the usec reference pulse to the local TKE and the FSI fabric, and the msec reference pulse to the local GPIO controller.

The FSI-TSC logic operates directly on the FSI XTAL clock (also known as OSC) working with 40 MHz, there is no 31.25 MHz clock and so no CAR level divider to generate such a clock.

The following figure shows the block diagram, showing how the SoC side TSC is modified to get the FSI-TSC.

Figure 8.28 FSI-TSC Block Diagram (Showing Similarities with SoC Side TSC)



8.3.2.18.2 Registers and APB Interface

FSI-TSC has the same register windows as the SoC counterpart:

- The Arm defined windows, arfsi_tsc_sysctr{0,1,2} are strictly identical to the equivalent SoC side windows, except that there is no TrustZone support.
- The NVIDIA implementation window, arfsi_tsc is much simpler than the SoC side and detailed below.
- The USEC address window, arfsi_usec_cntr is identical to the equivalent SoC side window, except reset values.

Note that to fit in the FSI Low Latency Peripheral address range, the different TSC windows are mapped on consecutive 4 KiB windows, not 64 KiB as on the SoC side. Offsets inside each window are unchanged. This is possible as support for page based address virtualization is not needed in

the FSI partition. Also, the SCR registers are not present, protection is done via one BLF per window.

The FSI-TSC implementation window contains the same registers as defined for the Main TSC, but with different default values, as summarized below. Values highlighted correspond to differences in reset values from the SoC side.

Register: FTSCACR, RW, equivalent to MTSACR, TSC Adjust Control Register			
Field name	**Bits**	**Reset**	**Description**
ABS	7:4	0	Adjust Bit Select, an edge on MTSC[9+ABS] triggers one adjust operation
DIR	1	1	Direction of the adjustment, 0 indicates A0 = -1, 1 indicates A0 = +1
EN	0	0	Enables frequency error correction

Register: FTSCADR, RW, equivalent to MTSCADR, TSC Adjust Divider Register			
Field name	**Bits**	**Reset**	**Description**
D1	27:16	0	Numerator of the adjust NCO
N1	11:0	0	Denominator of the adjust NCO

Register: FTSCANNR, RW, equivalent to MTSCANNR, TSC Adjustable NCO Numerator Register, different reset values			
Field name	**Bits**	**Reset**	**Description**
M0	27:16	25	Fractional part of the numerator of the adjustable NCO
R0	11:0	0	Integer part of the numerator of the adjustable NCO

Register: FTSCANDR, RW, equivalent to MTSCANDR, TSC Adjustable NCO Denominator Register, different reset value			
Field name	**Bits**	**Reset**	**Description**
D0	11:0	32	Denominator of the adjustable NCO

Register: FTSC MTSCCNTCV{0,1}, RO, equivalent to TSC_MTSCCNTCV{0,1}, TSC Counter Count Value Register			
Field name	**Bits**	**Reset**	**Description**
CV	31:0 {0}	0	Numerator of the adjust NCO

The FSI USEC window is also identical to the SoC counterpart except for some reset values.

Register: USECCVR, RO, Microsecond Counter Value Register			
Field name	**Bits**	**Reset**	**Description**
CNT	31:0	0	Number of usec pulses since last system reset

Register: USECCCR, Microsecond Counter Configuration Register, different reset values			
Field name	**Bits**	**Reset**	**Description**
Dividend	15:8	0	Numerator is Dividend + 1, aka N
Divisor	7:0	39	Denominator is Divisor + 1, default value corresponds to 40 MHz OSC

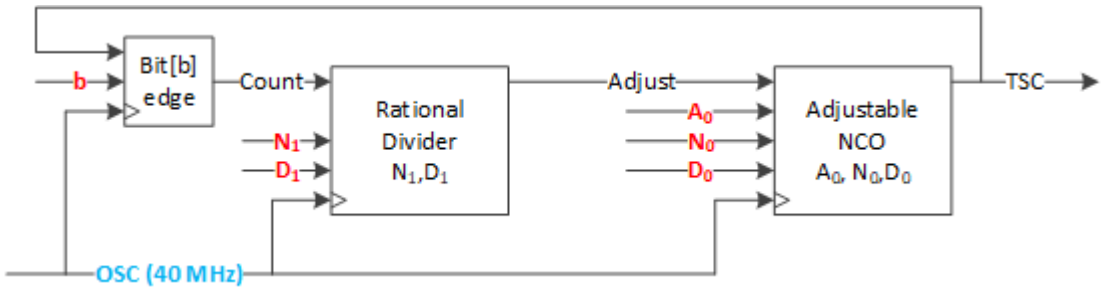
Register: USECCRSR, Microsecond Counter Reference Selection Register			
Field name	**Bits**	**Reset**	**Description**
MBS	5:4	0	TSC Bit Select, TSC[1+MBS] edge used as reference when not free running.
FR	0	1	Free Running, controls the microsecond counter TSC lock behavior When 0b, the microsecond locks the usec reference pulse to a selected bit of TSC[4:1] using the adjust signal, this is the recommended mode When 1b, usec counter counts on each edge of OSC in free running fashion, legacy mode

Register: USECCFR, Microsecond Counter Freeze Register			
Field name	**Bits**	**Reset**	**Description**
HDBG	0	0	Halt-on-debug. Controls whether a Halt-on-debug signal halts the microsecond and associated divider logic 0: Microsecond counter ignores Halt-on-debug. 1: Asserted Halt-on-debug signal halts the ration divider, automatically stopping the microsecond counter and the timing reference signals.

8.3.2.18.3 System Counter

The system counter reuses the Main TSC logic from the SoC side but running at 40 MHz. The diagram below is identical to the Main TSC Generation figure. The only change is a different target operating frequency.

Figure 8.29 FSI-TSC Generation



8.3.2.18.4 TSC Distribution

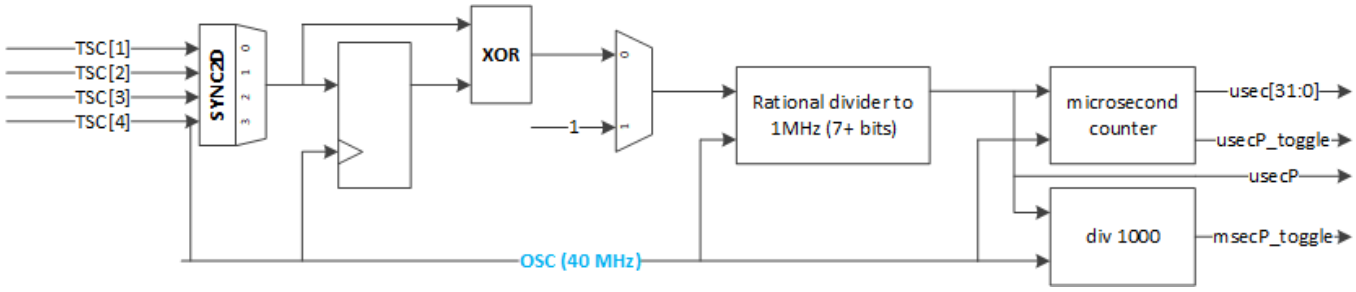
The TSC value generated by the system counter is broadcast in parallel binary format to all TSC endpoints: Cortex-R52 cluster, FSI-TKE, and CAN. Retime stages are allowed and inserted as required by physical design rules.

If the endpoint operates on a clock different from FSI OSC, the TSC value is passed through a Clock Domain Crossing element.

8.3.2.18.5 USEC Functions

The USEC functions are identical to the SoC counterpart, except again for the operating frequency, as shown in the diagram below.

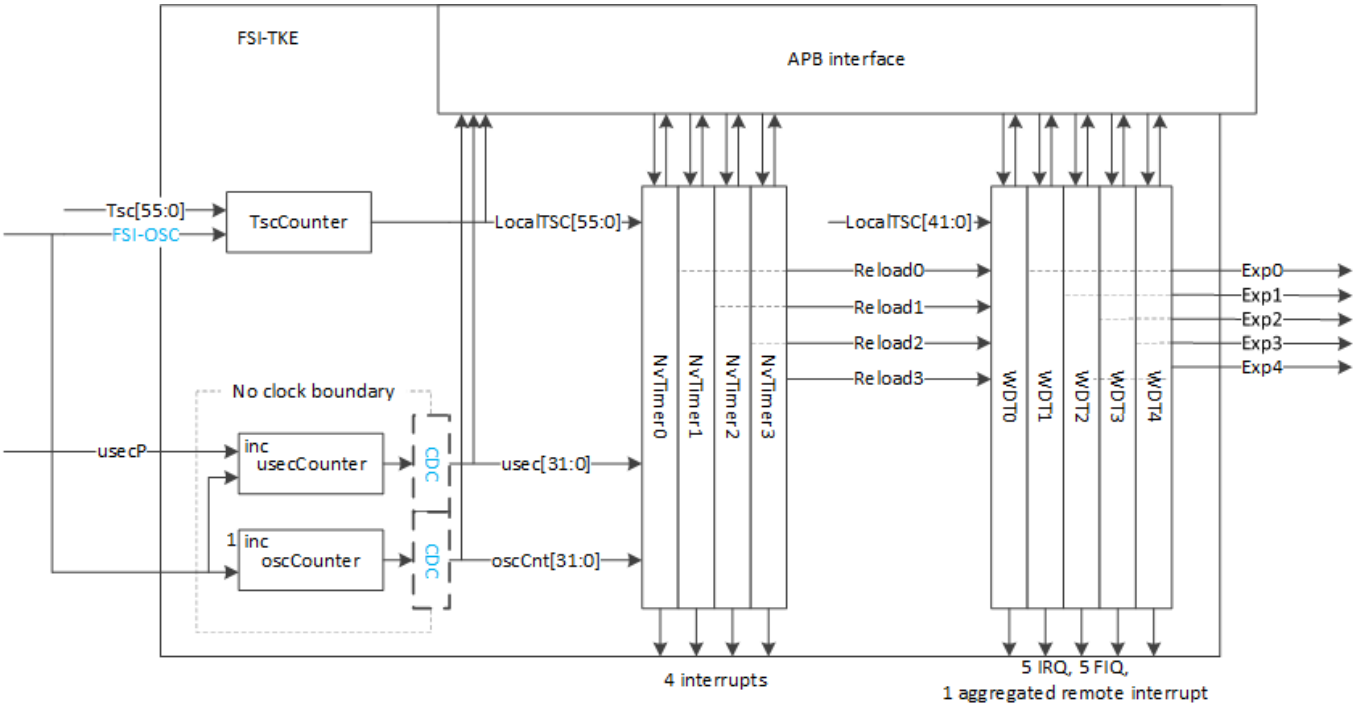
Figure 8.30 FSI Reference Pulses Generation and Microsecond Counter



8.3.2.18.6 FSI-TKE

The FSI-TKE is a standard instance of TKE, with four NV timers and five Watchdog Timers, but with the refclk input connected to the FSI crystal clock. This difference is transparent to hardware, in particular, the TSC input remains synchronous with the operating clock. The TKE logic includes CDC for the other two counters, these become unnecessary but are kept to maximize reuse. The figure below is derived from the TimeKeeping Element IP Module figure with the optional interrupt routing removed (not present in FSI-TKE), the correct number of timers and WDTs shown and the now unnecessary CDC highlighted.

Figure 8.31 FSI Reference Pulses Generation and Microsecond Counter (without Interrupt Routing)



As this is a standard instance, the registers are mapped in 64 KiB pages and include the legacy SCR, even if SCR are useless inside FSI. The different 64 KiB pages also get each a dedicated BLF to implement correct access rights.

The WDT error signals are directly connected to the FSI HSM.

The WDT debug reset requests originating in the FSI TKE are routed to PMC, i.e., outside FSI.

The different interrupts are connected in the following fashion; this also implies the allocation of timers and WDTs across the two processor clusters present in the FSI:

- Timers 0 and 1 interrupts to the GIC of the Cortex-R52 processor cluster
- Timers 2 and 3 interrupts to the AVIC of the CHSM cluster
- WDTs {0-3} IRQ and FIQ forwarded to the GIC of the Cortex-R52 processor cluster
- WDT 4 IRQ and FIQ forwarded to the AVIC of the CHSM cluster
- Aggregated remote interrupt of all WDTs, forwarded to be the GIC of the Cortex-R52 processor cluster and the AVIC of the CHSM cluster. The remote interrupt is also forwarded to the DFD Cross Trigger Interface (CTI) for debug.

Note that the FIQ are sent as active low signals for legacy reasons, they can be inverted to active high signal on their way to the different interrupt controllers.

8.3.2.19 TSC Update Frequency

The recommended frequency is 31.25 MHz, or equivalently, a 32 ns period.

8.3.2.19.1 TSC Update Frequency Starting from PLLP

The cardinality of the operands, Numerator (N) and Denominator (D), for the secondary TSC rational divider in path 0 is dependent on the division performed in the CAR divider. The ratio is the product of the CAR divider ratio times the rational divider ratio. At least a factor of two is mandatory, whereas the divide-by-2 case is preferred but the by-4 case is also allowed.

Table 8.20 Configuration to Generate 31.25 MHz

CAR Divider	CAR Frequency (MHz)	CAR Frequency/31.25 MHz		CAR Period, also Induced Jitter (ns)
		Ratio	# of Bits (N,D)	
2	204	$\frac{125}{816} = \frac{5^3}{2^4 \times 3 \times 17}$	(7,10)	4.902
4	102	$\frac{125}{408} = \frac{5^3}{2^3 \times 3 \times 17}$	(7,9)	9.804

The (N,D) of the table above are held in

- CLK_RST_CONTROLLER_TSC_HS_SUPER_CLK_DIVIDER_0.SUPER_TSC_HS_DIV_DIVIDEND (N)
- CLK_RST_CONTROLLER_TSC_HS_SUPER_CLK_DIVIDER_0.SUPER_TSC_HS_DIV_DIVISOR (D)

See Clock Controller and Reset (CAR) chapter for details.

8.3.2.19.2 TSC Update Frequency Starting from OSC

The cardinality of the operands, Numerator (N) and Denominator (D), for the secondary TSC rational divider in path 1 is constrained by the required support for a set of possible OSC frequencies. The table below shows the input frequency OSC_DIV (OSC frequency, 38.4 MHz, divided by 1) and the corresponding Rational Divider Parameters for the Update Frequency 31.25 MHz (F).

Table 8.21 TSC Rational Divider Parameters

OSC_Div (MHz)	K	F/K	(F/K) / OSC_Div	
			Ratio	# of Bits (N,D)
38.4 (38.4 / 1)	1	31.25	$\frac{5^4}{2^8 \times 3} = \frac{625}{768}$	(10,10)

The (N,D) of the table above are held in

- CLK_RST_CONTROLLER_TSC_OSC_SUPER_CLK_DIVIDER_0.SUPER_TSC_OSC_DIV_DIVIDEND (N)
- CLK_RST_CONTROLLER_TSC_OSC_SUPER_CLK_DIVIDER_0.SUPER_TSC_OSC_DIV_DIVISOR (D)

See Clock Controller and Reset (CAR) chapter for details.

8.3.2.20 Safety Mechanism

The following blocks support register parity and send the parity error signals to an error collator:

- RTC
- TSC
- USEC

They are all inside the Always-On cluster and share the Always-On cluster error collator with similar modules. The Always-On cluster error collator forwards an aggregated error signal to the Hardware Safety Manager (HSM).

All hardware watchdog timers also generate an error signal when their expiration level is above a programmed threshold. This error signal is directly connected to the HSM without going through an error collator.

8.3.3 Programming Guidelines

The following subsection provides guidance on the exact usage of the registers, especially when the sequence of accesses is important.

8.3.3.1 Use of Top TKE for CPU Core Wake

The Arm generic timers inside CCPLEX cannot always be used as wake timers. Specifically, the Arm generic timers stop operating when the associated cluster is power gated. So during the time a CPU cluster is power gated, associated NV timers in Top TKE become the CPU core wake timers.

The implementation is described below:

- There is a static one-to-one association between a CPU core, a NV timer in Top TKE and an interrupt, the mapping is dependent on the mode of operation of the cluster: split or lock-step. The exact mapping is defined by software, the following is a suggestion and not normative:
  - In split mode, core i in cluster j is associated with timer i + N * j, with N the number of cores per cluster and the same numbered shared interrupt.
  - In lockstep mode, core i in cluster j is associated with timer i + N * j, with N the number of core pairs per cluster and the same numbered shared interrupt. Each primary core has an associated redundant that is not associated to any timer (and not truly numbered or visible architecturally).
- Before going to sleep, software running on that core programs the earliest expiration of any Arm generic timer currently active to the associated NV timer. The NV timer is not enabled yet at that time as the Arm generic timers are still active, so only the expiration time is programmed. Software also stores the state of all Arm generic timers to memory.

  Note: Because Arm generic timers use a system register interface, only software running on a given core has access to the corresponding Arm generic timer, this is the underlying problem that requires to use the NV timer as a wake timer per CPU core instead of one NV timer per CPU cluster.
- Before power gating a CPU cluster, all NV timers associated with that CPU cluster are enabled. The interrupts may also be enabled at this time or may be always enabled
  - Software must insure that races are not possible, i.e., that the CPU cluster will enter power gating before any Arm generic timer reaches expiration.
  - The NV timer interrupts are connected as SPI
- When an NV timer used as wake timer expires, the corresponding interrupt wakes the CPU cluster (if needed), then wakes the corresponding core. Software running on the CPU core

- Identifies that this a wake coming from the NV timer
- In the improbable case that the wake time is less than the earliest expiration of an Arm timer, restart the NV timer and go back to sleep.
- Restores the state of the Arm generic timer, by construction this includes one Arm generic timer with the same expiration as the NV timer, and so that timer fires immediately. That interrupt will be treated as usual

When the NV timers in Top TKE are used as CPU core wake timers, they are preferably configured in the following way:

- Use TSC as the reference
- Use Timer Absolute Target Register (TMRATR) programmed to match the corresponding TSC value in the Arm generic timer. This requires some care as TMRATR is only 29 bits, and comparisons are done using finite horizon arithmetic.
  - If the expiration is less than 2^{28} TSC units in the future, about 8.5 seconds, program the 29 LSB of the expiration in TMRATR and program TSC itself as the reference in Timer Clock Source Selection Register (TMRCSSR).
  - If the expiration is more than 2^{28} but less than 2^{40} TSC units in the future, about 9.7 hours, change the reference in TMRCSSR to be TSC shifted by 12, shift the expiration by 12 to the right, and program the 29 LSB of the shifted value into TMRATR.
  - If the expiration is more than 2^{40} TSC units in the future, saturate. When the interrupt triggers, restart the NV timer using the same procedure (possibly saturating again).

8.3.3.2 Context Save and Restore Across Power State Transitions

Power state transitions require the affected modules to be restarted after the power state transition, with SC7 entry/exit the most important power state transition. There are a few canonical ways to handle this:

- Do nothing, the reset value of the register is correct.
- Registers are restored to a statically defined value. The register does not need to be fully static, but the initial value after the power state transition is not dependent on the previous history.
- Registers are restored based on their (dynamic) state prior to the power state transition. This requires to save a relevant context, normally in external memory, and later to use the saved context to restore registers. Note that preserving the context can be done either:
  - During SC7 entry
  - When the register is modified, this may be required for registers that are not accessible to the SC7 exit process, generally because the SC7 entry process does not have sufficient access rights.

This section provides guidance for the registers, in particular detailing all special cases where the software sequence may be more complex than writing back the stored value, including:

- Security aspects linked to the new security model.
- Registers that are have special access rules like indirection or lockable.
- Registers that require some adaptation between save and restore.

The SC7 restore proceeds in two steps to reduce the latency associated with power state transitions:

- A small set of registers, including the SCR are restored first by a secure software process. This set should be small for performance reason. Note that in many cases, the SCR have static values, so they do not require to be saved.
- Most of the registers protected by the SCR, especially most of the TrustZone secure registers are ideally restored later, while some unrelated non-secure software processes may already be active before full restoration takes place.

This specific way of handling SC7 requires checking if the default value of registers presents a security exposure. This section classifies the different registers in three categories:

- R0: Registers that must be restored in the first step to avoid a security exposure.
- R1: Registers that may be restored in the first step for a fully clean solution. These registers must be so that information leakage can be fully averted if context restoration proceeds in a specific sequence. However not following the ideal sequence only allows non-secure software to observe TrustZone events that otherwise would be hidden, but without compromising security itself.
- R2: Restored registers with no specific security requirements.
- NR: Registers that do not need to be restored

All TSC registers are in the A0 partition and are NR. Note however that a portion of the TSC logic depends OSC or PLLP being active, so there are specific requirements to handle SC7, but the sequence is not expected to impact the registers. There are two special cases, as explained below.

STSCCR is special, to avoid the main and secondary TSC to drift apart, an INIT must be performed at each system reset. This can be done by reading STSCCR then writing it back with the INIT bit set.

Note: Secondary and main can drift apart when they are not locked together and secondary TSC uses PLLP outside of SC7. To avoid a dependency on the platform, INIT must be performed at each system reset, i.e., when the distribution leaves are reset.

Software is allowed to explicitly suspend distribution during low power states, in which case distribution may need to be restarted during exit of the low power state, making TSCDCR special. This is not mandatory; the hardware distribution is designed to allow the distribution logic in TSC to remain continuously active even if the distribution network on the SoC side is in reset.

The tables below list the registers and the associated guidance and are extensions of the corresponding summary tables.

Table 8.22 Guidance for SC7 Handling of Arm TSC Control Registers

Register Name	Category	Remark
CNTCR	NR	In Always On partition, not affected by SC7.
CNTSR		
CNTCV0		
CNTCV1		
CNTFID0		
CNTFID1		
CounterID4		
CounterID{5:7}		
CounterID{0:3}		
CounterID{8:11}		

Table 8.23 Guidance for SC7 Handling of Arm TSC Status Registers

Register Name	Category	Remark
CNTCV0	NR	In Always On partition, not affected by SC7.
CNTCV1		
CounterID4		
CounterID{5:7}		
CounterID{0:3}		
CounterID{8:11}		

Table 8.24 Guidance for SC7 Handling of TSC Implementation Registers

Register Name	Category	Remark
MTSCACR	NR	In Always On partition, not affected by SC7. Note however that a portion of the TSC logic depends on OSC or PLLP being active, so there are specific requirements to handle SC7, but the sequence depicted there does not impact the registers.
MTSCADR		
MTSCANNR		
MTSCANDR		
MTSCCNTCV0		
MTSCCNTCV1		
STSCCR		

Register Name	Category	Remark
STSCRSR		
STSCIR{0,1}		
STSCSR		
TSCDCR		
TSCSCR		

Table 8.25 Guidance for SC7 Handling of TSC CAR Registers

Register Name	Category	Remark
CSTSCR0	R0	These are marked R0 because it is assumed they provide the reference for (secure) time, and so should probably restored quickly. There are no associated security requirements. The values restored are static for a given platform, so not clear if this requires a saving step.
TSCDIVR0	R0	
TSCDIVR1	R0	

Table 8.26 Guidance for SC7 Handling of Common GTE Registers, when Outside of AO Partition

Register Name	Category	Remark
TECTRL{t}	R2	Generic TimeStamp Engine Control
TETSCH{t}	NR	24 MSBs of the TSC associated with a timestamped event
TETSCL{t}	NR	32 LSBs of the TSC associated with a timestamped event
TECCV{t}	NR	Current captured value of bitmap for timestamped event
TEPVV{t}	NR	Previous captured value of bitmap for timestamped event
TESRC{t}	NR	Identifies the source of the timestamped event
TEENCV{t}	NR	Encoded version of TECV{t} and TEPV{t}, with explicit valid
TECMD{t}	NR	Command register, especially to pop the FIFO
TESTATUS{t}	NR	Status register
TESCR{t}	R0	Access rights control register for timestamp engine {t}.
TEDSCR{t}	R0	Specific security domain for disable bits for timestamp engine {t}.

Table 8.27 Guidance for SC7 Handling of per Slice GTE Registers, when Outside of AO Partition

Register Name	Category	Remark
TETEN{t}{s}	R2	

Register Name	Category	Remark
TETDN{t}{s}	R1	If not restored in step 1, non-secure software can observe the timing and occurrence of secure events. This can be avoided by restoring this register before enabling the event source. As only observation is possible if the sequence above is not followed, security is not directly compromised, allowing R1.

Table 8.28 Guidance for SC7 Handling of RTC Registers

Register Name	Category	Remark
RTCCR	NR	In Always On partition, not affected by SC7.
RTCBR		
RTCSCR		
RTCSSCR		
RTCMCR		
RTCSAR{0,1}		
RTCMAR		
RTCSCAR		
RTCMCAR		
RTCIER		
RTCISR		
RTCIVR		
RTCFIR		
RTCRSR		
RTCDR		
RTCSCR		
AOWDTCR		
AOWDTSR		
AOWDTCMDR		
AOWDTUR		
AOWDTSCOR		
AOWDTSCR		

Table 8.29 Guidance for SC7 Handling of USEC Registers

Register Name	Category	Remark
USECCVR	NR	In Always On partition, not affected by SC7.
USECCCR		
USECCRSR		
USECCFR		
USECSCR		

Table 8.30 Guidance for SC7 Handling of TKE Registers, when outside of AO partition

Register Name	Category	Remark
TKETSC{0,1}	NR	
TKEUSEC	NR	
TKEOSC	NR	
TKECR	R2	
TKEIE{i}	R2	
TKEIV	NR	
TKEIR	NR	
TKESCR	R0	
TMRCR{t}	R2	Restoring the state of a timer may require compensating for the amount of time the power state consumed.
TMRSR{t}	NR (but saved)	
TMRCSSR{t}	R2	
TMRSCR{t}	R0	
WDTCR{w}	R2	WDT cannot compensate for the amount of time the power of state consumed, they can only be restarted from scratch.
WDTSR{w}	NR	
WDTCMDR{w}	NR	
WDTUR{w}	NR	
WDTSCOR{w}	R2	
WDTSCR{w}	R0	

Restoring a TKE timer needs to consider a few special cases when trying to compensate for the amount of time consumed by the power state:

- For a one off timer
  - The timer may have expired during the power state, the restore code can perform either of the following:
    - Insert an (fake) event into the handler.
    - Program the timer to its shortest possible duration, resulting in the raising of the corresponding interrupt as usual.
  - If the timer has not expired, the compensation can be done by performing any of the following:
    - Restore without compensation, the timer is linked to the amount of time elapsed in the active state, not real time.
    - Restore in absolute fashion, if the timer targets a known real time deadline, the timer delay can be computed by target time minus current time. This has the advantage that errors do not accumulate if the timer is suspended multiple times. This requires to save the calculated expiration time, not the TMRSR{T} itself.
    - Restore in relative fashion, calculate the timer delay as the timer value at SC7 entry minus the amount of time consumed during SC7. This requires saving TMRSR{t}.PCV and to know the time spent in SC7. Note that the TMRSR{t}.PCV value is restored on a calculated value not the saved value itself.
- For a periodic timer:
  - The timer may have expired a number of times during the power state. Then, it may need to insert corresponding (fake) events into the handler.
  - A timer cannot be restarted at a value different than its period. Then the restore may need to proceed in two phases if the phase of the timer is important.
    - Restart the timer as a one off for the remainder of the period, using the same time of calculation for one-off timer.
    - At the first expiration after the power state transition, restarts the timer in periodic mode.
  - If the phase of the periodic timer is not important, but only its period, then restart it immediately.

A final requirement is to assert/deassert the reset to terminal nodes located in a power-gated partition if the reset is software controlled.

8.3.4 Timer Registers

Refer to "Reading Register Tables" in the Introduction chapter for the register table protocol as well as recommendations for accessing registers.

The register descriptions below give the offset of each register within the Timers' address range. The Base Addresses of these registers are specified in the Address Map section of the Orin TRM.

8.3.4.1 Time Keeping Element Shared Registers

TKE_AON_SHARED_TKETSC0_0

Value of Local TSC counter, synchronized across SOC

Offset: 0x0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TKE_AON_SCR_TKESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CV**: Current value of TSC, TSC[31:0]

TKE_AON_SHARED_TKETSC1_0

Value of Master TSC counter, synchronized across SOC

Offset: 0x4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TKE_AON_SCR_TKESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CV**: 8h0 followed by Master TSC[55:32]

TKE_AON_SHARED_TKEUSEC_0

Value of Local USEC counter, not synchronized across SOC

Offset: 0x8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TKE_AON_SCR_TKESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CV**: Current value of microsecond counter

TKE_AON_SHARED_TKEOSC_0

Value of Local OSC counter, not synchronized across SOC

Offset: 0xc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TKE_AON_SCR_TKESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CV**: Current value of oscillator counter

TKE_AON_SHARED_TKECR_0

TKE Control register

Offset: 0x10
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_AON_SCR_TKESCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	0x0	**HDBG**: Halt-on-debug. Controls whether a Halt-on-debug signal halts TKE logic. 0: TKE ignores Halt-on-debug. 1: If TKE reference is local USEC/OSC counter, asserting this field disables the update of the reference counters and reference timing signals for all timers and all WDTs
2	0x0	**OSC_DIS**: Disable the local OSC counter - Can be used to decrease the power consumption if OSC is not required either by hardware as a reference clock or by software

Bit	Reset	Description
1	0x0	**USEC_DIS**: Disable the local USEC counter - Can be used to decrease the power consumption if USEC is not required either by hardware as a reference clock or by software
0	0x0	**TSC_DIS**: If TKE reference used is TSC, setting this disables the TSC capture locally. This disables the update of the reference counters and reference timing signals for all timers and all WDTs.

TKE_AON_SHARED_CLK_OVR_ON_0

Offset: 0x14
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_AON_SCR_TKESCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**OSC**: SLCG override bit **0** = DISABLE **1** = ENABLE

TKE_AON_TIMER_TMR0_TMRCR_0

```
Register name       Offset        Description
TMRCR{t}        BT + P*{t} + 0    Timer Configuration Register
TMRSR{t}        BT + P*{t} + 4    Timer Status Register
TMRCSSR{t}      BT + P*{t} + 8    Timer Clock Source Selection Register
TMRATR{t}       BT + P*{t} + 12   Timer Absolute Target Register
```

Timer Configuration Register

Offset: 0x10000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_AON_SCR_TMRSCR0_0
Reset: 0x40000000 (0b01x0,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: Enable, when set, enables the countdown process. **0** = DISABLE **1** = ENABLE
30	0x1	**PER**: Periodic, when set, the internal counter is automatically reloaded after reaching 0 **0** = DISABLE **1** = ENABLE
28:0	0x0	**PTV**: Timer Present Trigger Value. When TMRCR is written and EN is 1b, the internal counter is loaded with PTV+1 and starts counting down at each timing reference pulse. An interrupt is set when the internal counter reaches 0. Note that writing TMRCR with EN=1 while EN was already 1 restarts the counter. Once the timer is started, TARGET can be used to either - observe the value of the reference counter when the next interrupt will be generated (read) - select the exact reference counter value at which the next interrupt will be generated (write) **536870911** = MAX

TKE_AON_TIMER_TMR0_TMRSR_0

Timer Status Register

Offset: 0x10004
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: TKE_AON_SCR_TMRSCR0_0
Reset: 0x00000000 (0bx0x0,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Parity Protection	Reset	Description
30	RW	Y	0x0	**INTR_CLR**: Clears the interrupt when written as 1b, read always returns 0 **0** = DISABLE **1** = ENABLE
28:0	RO	N	0x0	**PCV**: Current counter value, writing to this field has no effect

TKE_AON_TIMER_TMR0_TMRCSSR_0

Timer Clock Source Selection Register

Offset: 0x10008
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_AON_SCR_TMRSCR0_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**SRC_ID**: Select Count reference 2'b11 - TSC [41:12] 2'b10 - TSC [29:0] 2'b01 - oscCnt [29:0] 2'b00 - usecCnt [29:0] The implementation of each timer uses comparison against counters incrementing at the selected rate. **0** = SRC_USECCNT **1** = SRC_OSCCNT **2** = SRC_TSCCNT_29_0 **3** = SRC_TSCCNT_41_12

TKE_AON_TIMER_TMR0_TMRATR_0

Timer Absolute Target Register

Offset: 0x1000c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_AON_SCR_TMRSCR0_0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:0	0x0	**TARGET**: Target value of the selected reference counter, i.e., when value of the selected reference equals target value, the interrupt is generated. The difference between TARGET and the current value must be positive when interpreting TARGET and the 29 LSB of the selected reference counter as signed integers, otherwise the interrupt triggers immediately.

TKE_AON_TIMER_TMR1_TMRCR_0

```
Register name       Offset        Description
TMRCR{t}        BT + P*{t} + 0    Timer Configuration Register
TMRSR{t}        BT + P*{t} + 4    Timer Status Register
TMRCSSR{t}      BT + P*{t} + 8    Timer Clock Source Selection Register
TMRATR{t}       BT + P*{t} + 12   Timer Absolute Target Register
```

Timer Configuration Register

Offset: 0x20000

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_AON_SCR_TMRSCR1_0
Reset: 0x40000000 (0b01x0,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: Enable, when set, enables the countdown process. **0** = DISABLE **1** = ENABLE
30	0x1	**PER**: Periodic, when set, the internal counter is automatically reloaded after reaching 0 **0** = DISABLE **1** = ENABLE
28:0	0x0	**PTV**: Timer Present Trigger Value. When TMRCR is written and EN is 1b, the internal counter is loaded with PTV+1 and starts counting down at each timing reference pulse. An interrupt is set when the internal counter reaches 0. Note that writing TMRCR with EN=1 while EN was already 1 restarts the counter. Once the timer is started, TARGET can be used to either - observe the value of the reference counter when the next interrupt will be generated (read) - select the exact reference counter value at which the next interrupt will be generated (write) **536870911** = MAX

TKE_AON_TIMER_TMR1_TMRSR_0

Timer Status Register

Offset: 0x20004
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: TKE_AON_SCR_TMRSCR1_0
Reset: 0x00000000 (0bx0x0,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Parity Protection	Reset	Description
30	RW	Y	0x0	**INTR_CLR**: Clears the interrupt when written as 1b, read always returns 0 **0** = DISABLE **1** = ENABLE
28:0	RO	N	0x0	**PCV**: Current counter value, writing to this field has no effect

TKE_AON_TIMER_TMR1_TMRCSSR_0

Timer Clock Source Selection Register

Offset: 0x20008
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_AON_SCR_TMRSCR1_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**SRC_ID**: Select Count reference 2'b11 - TSC [41:12] 2'b10 - TSC [29:0] 2'b01 - oscCnt [29:0] 2'b00 - usecCnt [29:0] The implementation of each timer uses comparison against counters incrementing at the selected rate. **0** = SRC_USECCNT **1** = SRC_OSCCNT **2** = SRC_TSCCNT_29_0 **3** = SRC_TSCCNT_41_12

TKE_AON_TIMER_TMR1_TMRATR_0

Timer Absolute Target Register

Offset: 0x2000c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_AON_SCR_TMRSCR1_0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:0	0x0	**TARGET**: Target value of the selected reference counter, i.e., when value of the selected reference equals target value, the interrupt is generated. The difference between TARGET and the current value must be positive when interpreting TARGET and the 29 LSB of the selected reference counter as signed integers, otherwise the interrupt triggers immediately.

TKE_AON_TIMER_TMR2_TMRCR_0

```
Register name       Offset        Description
TMRCR{t}        BT + P*{t} + 0    Timer Configuration Register
TMRSR{t}        BT + P*{t} + 4    Timer Status Register
TMRCSSR{t}      BT + P*{t} + 8    Timer Clock Source Selection Register
TMRATR{t}       BT + P*{t} + 12   Timer Absolute Target Register
```

Timer Configuration Register

Offset: 0x30000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_AON_SCR_TMRSCR2_0
Reset: 0x40000000 (0b01x0,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: Enable, when set, enables the countdown process. **0** = DISABLE **1** = ENABLE
30	0x1	**PER**: Periodic, when set, the internal counter is automatically reloaded after reaching 0 **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
28:0	0x0	**PTV**: Timer Present Trigger Value. When TMRCR is written and EN is 1b, the internal counter is loaded with PTV+1 and starts counting down at each timing reference pulse. An interrupt is set when the internal counter reaches 0. Note that writing TMRCR with EN=1 while EN was already 1 restarts the counter. Once the timer is started, TARGET can be used to either - observe the value of the reference counter when the next interrupt will be generated (read) - select the exact reference counter value at which the next interrupt will be generated (write) **536870911** = MAX

TKE_AON_TIMER_TMR2_TMRSR_0

Timer Status Register

Offset: 0x30004
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: TKE_AON_SCR_TMRSCR2_0
Reset: 0x00000000 (0bx0x0,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Parity Protection	Reset	Description
30	RW	Y	0x0	**INTR_CLR**: Clears the interrupt when written as 1b, read always returns 0 **0** = DISABLE **1** = ENABLE
28:0	RO	N	0x0	**PCV**: Current counter value, writing to this field has no effect

TKE_AON_TIMER_TMR2_TMRCSSR_0

Timer Clock Source Selection Register

Offset: 0x30008
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: TKE_AON_SCR_TMRSCR2_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**SRC_ID**: Select Count reference 2'b11 - TSC [41:12] 2'b10 - TSC [29:0] 2'b01 - oscCnt [29:0] 2'b00 - usecCnt [29:0] The implementation of each timer uses comparison against counters incrementing at the selected rate. **0** = SRC_USECCNT **1** = SRC_OSCCNT **2** = SRC_TSCCNT_29_0 **3** = SRC_TSCCNT_41_12

TKE_AON_TIMER_TMR2_TMRATR_0

Timer Absolute Target Register

Offset: 0x3000c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_AON_SCR_TMRSCR2_0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:0	0x0	**TARGET**: Target value of the selected reference counter, i.e., when value of the selected reference equals target value, the interrupt is generated. The difference between TARGET and the current value must be positive when interpreting TARGET and the 29 LSB of the selected reference counter as signed integers, otherwise the interrupt triggers immediately.

TKE_AON_TIMER_TMR3_TMRCR_0

```
Register name       Offset        Description
TMRCR{t}        BT + P*{t} + 0    Timer Configuration Register
TMRSR{t}        BT + P*{t} + 4    Timer Status Register
TMRCSSR{t}      BT + P*{t} + 8    Timer Clock Source Selection Register
TMRATR{t}       BT + P*{t} + 12   Timer Absolute Target Register
```

Timer Configuration Register

Offset: 0x40000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_AON_SCR_TMRSCR3_0
Reset: 0x40000000 (0b01x0,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: Enable, when set, enables the countdown process. **0** = DISABLE **1** = ENABLE
30	0x1	**PER**: Periodic, when set, the internal counter is automatically reloaded after reaching 0 **0** = DISABLE **1** = ENABLE
28:0	0x0	**PTV**: Timer Present Trigger Value. When TMRCR is written and EN is 1b, the internal counter is loaded with PTV+1 and starts counting down at each timing reference pulse. An interrupt is set when the internal counter reaches 0. Note that writing TMRCR with EN=1 while EN was already 1 restarts the counter. Once the timer is started, TARGET can be used to either - observe the value of the reference counter when the next interrupt will be generated (read) - select the exact reference counter value at which the next interrupt will be generated (write) **536870911** = MAX

TKE_AON_TIMER_TMR3_TMRSR_0

Timer Status Register

Offset: 0x40004
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: TKE_AON_SCR_TMRSCR3_0
Reset: 0x00000000 (0bx0x0,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Parity Protection	Reset	Description
30	RW	Y	0x0	**INTR_CLR**: Clears the interrupt when written as 1b, read always returns 0 **0** = DISABLE **1** = ENABLE
28:0	RO	N	0x0	**PCV**: Current counter value, writing to this field has no effect

TKE_AON_TIMER_TMR3_TMRCSSR_0

Timer Clock Source Selection Register

Offset: 0x40008
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_AON_SCR_TMRSCR3_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**SRC_ID**: Select Count reference 2'b11 - TSC [41:12] 2'b10 - TSC [29:0] 2'b01 - oscCnt [29:0] 2'b00 - usecCnt [29:0] The implementation of each timer uses comparison against counters incrementing at the selected rate. **0** = SRC_USECCNT **1** = SRC_OSCCNT **2** = SRC_TSCCNT_29_0 **3** = SRC_TSCCNT_41_12

TKE_AON_TIMER_TMR3_TMRATR_0

Timer Absolute Target Register

Offset: 0x4000c
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: TKE_AON_SCR_TMRSCR3_0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:0	0x0	**TARGET**: Target value of the selected reference counter, i.e., when value of the selected reference equals target value, the interrupt is generated. The difference between TARGET and the current value must be positive when interpreting TARGET and the 29 LSB of the selected reference counter as signed integers, otherwise the interrupt triggers immediately.

TKE_AON_WDT0_WDTCR_0

```
WDTCR{w}   BW + P*{w} + 0  RW    Watchdog Timer Configuration Register
WDTSR{w}   BW + P*{w} + 4  RO    Watchdog Timer Status Register
WDTCMDR{w} BW + P*{w} + 8  WO    Watchdog Timer Command Register
WDTUR{w}   BW + P*{w} + 12 RW    Watchdog Timer Unlock Register
```

Watchdog Timer Configuration Register

Offset: 0x50000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_AON_SCR_WDTSCR0_0
Reset: 0x00700000 (0b0000,xx00,0111,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**WindowedRestartDisableMap**: When WindowedOperationEnable is 1b, restart commands attempted when expiration level has value b are rejected if bit b equals 1b in this field
25	0x0	**DisallowWDTFreeze**: When 1b, the signal watchdog_freeze_request has no effect When 0b, when the signal watchdog_freeze_request is asserted, the WDT stops decrementing CurrentCount if the WDT is enabled

Bit	Reset	Description
24:23	0x0	**RestartErrorAction**: If a restart action is attempted but fails, indicate which of four actions to take 0x0: do nothing, silently ignore the restart 0x1: immediately jump to expiration level 4 and reload the downcounter 0x2: immediately jump to expiration level 5 and reload the downcounter 0x3: immediately request a system power on reset (POR) When jumping levels, the normal actions configured per level must take place **0** = NOP **1** = JUMP_TO_4 **2** = JUMP_TO_5 **3** = POR_RESET
22:20	0x7	**ErrorThreshold**: The WDT logic asserts an error signal to HSM when ExpirationLevel >= ErrorThreshold.
19	0x0	**TscReferenceEnable**: Select as timing reference transitions on a configured TSC bit.
18	0x0	**ChallengeResponseEnable**: Enable the Challenge Response mode of operation. When 1b, the Unlock register operates as a Challenge Response register.
17	0x0	**WindowedOperationEnable**: Enable the windowed mode of operation. When 1b, a restart command is rejected when the current expiration level is identified as disallowed in the WindowedRestartDisableMap or the current time is below the window threshold. The rejection is either silent or performs a reset based on RestartErrorAction
16	0x0	**SystemPOResetEnable**: Enable internal full system reset at fifth expiration of the counter. This reset cannot be intercepted and acts as a Power On Reset **0** = DISABLE **1** = ENABLE
15	0x0	**SystemDebugResetEnable**: Enable systemwide reset assertion at fourth expiration of the counter. This reset can be intercepted by debug logic, e.g., to preserve information
14	0x0	**RemoteInterruptEnable**: Enable normal priority interrupt assertion at third expiration of the counter, connected to LIC for systemwide visibility
13	0x0	**LocalFIQEnable**: Enable high priority interrupt (FIQ) assertion at second expiration of the counter, connected to local interrupt controller
12	0x0	**LocalInterruptEnable**: Enable normal priority interrupt assertion at first expiration of the counter, connected to local interrupt controller

Bit	Reset	Description
11:4	0x0	**Period**: Measured in periods of the timer selected as source, this is the reload value, so 0 is treated as maximum period
3:0	0x0	**TimerSource**: The timer source interpretation is dependent on the value of TscReferenceEnable: TscReferenceEnable = 0b: timer<TimerSource> is the timing reference TscReferenceEnable = 1b: a transition of TSC[2*TimerSource] is the timing reference **0** = TMR0

TKE_AON_WDT0_WDTSR_0

Watchdog Timer Status Register

Offset: 0x50004
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TKE_AON_SCR_WDTSCR0_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,x000,0000,0000,0000)

Bit	Reset	Description
19	0x0	**Frozen**: 1b indicates that the WDT is frozen because of the assertion of the global WDT freeze request (when allowed) or because of the assertion of the global timer freeze request
18	0x0	**ErrorForced**: 1b indicates that the error reported to HSM is forced active for testing purposes
17	0x0	**StickyEnabled**: 1b when the counter is enabled and cannot be disabled, set by StickyStart command, cleared by reset only.
16	0x0	**CurrentError**: Current error reported to HSM, this is the output after combining the functional and forced error paths
14:12	0x0	**CurrentExpirationCount**: Current count of expiration since last start operation, cleared by Start command, saturating at 5. **0** = NO_EXPIRY **1** = LOCAL_IRQ **2** = LOCAL_FIQ **3** = REMOTE_IRQ **4** = DEBUG_RESET **5** = SYSTEM_RESET

Bit	Reset	Description
11:4	0x0	**CurrentCount**: Current value of the downcounter, reloaded with Period by Start command.
3	0x0	**RemoteInterruptStatus**: Current status of remote interrupt, cleared by Start command.
2	0x0	**LocalFIQStatus**: Current status of FIQ, cleared by Start command.
1	0x0	**LocalInterruptStatus**: Current status of local interrupt, cleared by Start command.
0	0x0	**Enabled**: 1b when the counter is active. When Enabled is 1b, the configuration of the WDT is frozen by silently ignoring write attempts to the configuration registers: WDTCR, WDTSCR and WDTFWCR. Set by Start or StickyStart commands, cleared by Disable when StickyEnabled is 0b. **0** = DISABLE **1** = ENABLE

TKE_AON_WDT0_WDTCMDR_0

Watchdog Timer Command Register

Offset: 0x50008
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_AON_SCR_WDTSCR0_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	0x0	**ForceError**: Only working if the Command register is unlocked. 1b sets an internal flop that is ORed with the functional error signal (based on threshold), 0b clears the bit. The state of the flop is shown in the status register.
2	0x0	**StickyStart**: Qualifies the StartCounter bit, if both bits are 1b, the counter is enabled in permanent fashion, i.e., it remains enabled until the next reset, and a disable counter operation has no effect. Ignored if StartCounter is 0b

Bit	Reset	Description
1	0x0	**DisableCounter**: Only working if the unlock register (WDTUR{w}) has been programmed before with the correct pattern. Writing to the command register always clears the unlock register. If written to 1b, while StartCounter is 0b and the unlock register contains the unlock pattern, the watchdog transitions back to disabled. **0** = DISABLE **1** = ENABLE
0	0x0	**StartCounter**: When written to 1b, enables the counter operation, loads the counter with Period and starts downcounting, resets the expiration count to 0 and clears all status flags. Enabled is always 1b after this. This is also used as Restart. Writing to StartCounter requires unlocking when ChallengeResponseEnable is 1b, and writing to StartCounter is also restricted to windows of time given by WindowedRestartDisableMap when WindowedOperationEnable is 1b. **0** = DISABLE **1** = ENABLE

TKE_AON_WDT0_WDTUR_0

Watchdog Timer Unlock Register

Offset: 0x5000c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_AON_SCR_WDTSCR0_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UnlockPattern**: The behavior of the Unlock pattern is dependent on the ChallengeResponseEnable configuration bit. When ChallengeResponseEnable is 0b, UnlockPattern must be written with value UNLOCK to allow a write to DisableCounter to take effect. UnlockPattern is reset at each write to the Command register. When ChallengeResponseEnable is 1b, a write to the Command register is only allowed after reading UnlockPattern to get the current value of an LFSR, followed by writing the next value of the LFSR to the UnlockPattern. If the two values match, the unlock is successful. In all cases the LFSR gets (re)initialized with the written value. **50266** = UNLOCK

TKE_AON_WDT0_WDTSCR_0

Watchdog Timer Skip Configuration Register

Offset: 0x50010
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_AON_SCR_WDTSCR0_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,x000,x000,x000)

Bit	Reset	Description
14:12	0x0	**Skip3**: Skip Value at Expiration count 3 At a Downcounter expiration, the expiration level changes as ExpC <= ExpC + 1 + Skip(ExpC), saturating at 5
10:8	0x0	**Skip2**: Skip Value at Expiration count 2
6:4	0x0	**Skip1**: Skip Value at Expiration count 1
2:0	0x0	**Skip0**: Skip Value at Expiration count 0

TKE_AON_WDT0_WDTFWCR_0

Watchdog Timer Fine Window Configuration Register

Offset: 0x50014
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_AON_SCR_WDTSCR0_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,x000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
18:16	0x0	**StartLevel**: If restart (i.e., start while WDT is already enabled) is attempted and windowing operation is enabled, restart is rejected if current expiration < StartLevel

Bit	Reset	Description
7:0	0x0	**StartCount**: If restart (i.e., start while WDT is already enabled) is attempted and windowing operation is enabled and current expiration level == StartLevel, restart is rejected if current downcounter > (Period - StartCount). StartCount is the number of cycles where rejection takes place at the start of that level.

TKE_FSI_SHARED_TKETSC0_0

Value of Local TSC counter, synchronized across SOC

Offset: 0x0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TKE_FSI_SCR_TKESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CV**: Current value of TSC, TSC[31:0]

TKE_FSI_SHARED_TKETSC1_0

Value of Master TSC counter, synchronized across SOC

Offset: 0x4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TKE_FSI_SCR_TKESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CV**: 8h0 followed by Master TSC[55:32]

TKE_FSI_SHARED_TKEUSEC_0

Value of Local USEC counter, not synchronized across SOC

Offset: 0x8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TKE_FSI_SCR_TKESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CV**: Current value of microsecond counter

TKE_FSI_SHARED_TKEOSC_0

Value of Local OSC counter, not synchronized across SOC

Offset: 0xc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TKE_FSI_SCR_TKESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CV**: Current value of oscillator counter

TKE_FSI_SHARED_TKECR_0

TKE Control register

Offset: 0x10
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_TKESCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	0x0	**HDBG**: Halt-on-debug. Controls whether a Halt-on-debug signal halts TKE logic. 0: TKE ignores Halt-on-debug. 1: If TKE reference is local USEC/OSC counter, asserting this field disables the update of the reference counters and reference timing signals for all timers and all WDTs

Bit	Reset	Description
2	0x0	**OSC_DIS**: Disable the local OSC counter - Can be used to decrease the power consumption if OSC is not required either by hardware as a reference clock or by software
1	0x0	**USEC_DIS**: Disable the local USEC counter - Can be used to decrease the power consumption if USEC is not required either by hardware as a reference clock or by software
0	0x0	**TSC_DIS**: If TKE reference used is TSC, setting this disables the TSC capture locally. This disables the update of the reference counters and reference timing signals for all timers and all WDTs.

TKE_FSI_SHARED_CLK_OVR_ON_0

Offset: 0x14
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_TKESCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**OSC**: SLCG override bit **0** = DISABLE **1** = ENABLE

TKE_FSI_TIMER_TMR0_TMRCR_0

```
Register name       Offset          Description
TMRCR{t}        BT + P*{t} + 0   Timer Configuration Register
TMRSR{t}        BT + P*{t} + 4   Timer Status Register
TMRCSSR{t}      BT + P*{t} + 8   Timer Clock Source Selection Register
TMRATR{t}       BT + P*{t} + 12  Timer Absolute Target Register
```

Timer Configuration Register

Offset: 0x10000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_TMRSCR0_0
Reset: 0x40000000 (0b01x0,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: Enable, when set, enables the countdown process. **0** = DISABLE **1** = ENABLE
30	0x1	**PER**: Periodic, when set, the internal counter is automatically reloaded after reaching 0 **0** = DISABLE **1** = ENABLE
28:0	0x0	**PTV**: Timer Present Trigger Value. When TMRCR is written and EN is 1b, the internal counter is loaded with PTV+1 and starts counting down at each timing reference pulse. An interrupt is set when the internal counter reaches 0. Note that writing TMRCR with EN=1 while EN was already 1 restarts the counter. Once the timer is started, TARGET can be used to either - observe the value of the reference counter when the next interrupt will be generated (read) - select the exact reference counter value at which the next interrupt will be generated (write) **536870911** = MAX

TKE_FSI_TIMER_TMR0_TMRSR_0

Timer Status Register

Offset: 0x10004
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: TKE_FSI_SCR_TMRSCR0_0
Reset: 0x00000000 (0bx0x0,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Parity Protection	Reset	Description
30	RW	Y	0x0	**INTR_CLR**: Clears the interrupt when written as 1b, read always returns 0 **0** = DISABLE **1** = ENABLE
28:0	RO	N	0x0	**PCV**: Current counter value, writing to this field has no effect

TKE_FSI_TIMER_TMR0_TMRCSSR_0

Timer Clock Source Selection Register

Offset: 0x10008
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_TMRSCR0_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**SRC_ID**: Select Count reference 2'b11 - TSC [41:12] 2'b10 - TSC [29:0] 2'b01 - oscCnt [29:0] 2'b00 - usecCnt [29:0] The implementation of each timer uses comparison against counters incrementing at the selected rate. **0** = SRC_USECCNT **1** = SRC_OSCCNT **2** = SRC_TSCCNT_29_0 **3** = SRC_TSCCNT_41_12

TKE_FSI_TIMER_TMR0_TMRATR_0

Timer Absolute Target Register

Offset: 0x1000c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_TMRSCR0_0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:0	0x0	**TARGET**: Target value of the selected reference counter, i.e., when value of the selected reference equals target value, the interrupt is generated. The difference between TARGET and the current value must be positive when interpreting TARGET and the 29 LSB of the selected reference counter as signed integers, otherwise the interrupt triggers immediately.

TKE_FSI_TIMER_TMR1_TMRCR_0

```
Register name       Offset          Description
TMRCR{t}        BT + P*{t} + 0      Timer Configuration Register
TMRSR{t}        BT + P*{t} + 4      Timer Status Register
TMRCSSR{t}      BT + P*{t} + 8      Timer Clock Source Selection Register
TMRATR{t}       BT + P*{t} + 12     Timer Absolute Target Register
```

Timer Configuration Register

Offset: 0x20000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_TMRSCR1_0
Reset: 0x40000000 (0b01x0,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: Enable, when set, enables the countdown process. **0** = DISABLE **1** = ENABLE
30	0x1	**PER**: Periodic, when set, the internal counter is automatically reloaded after reaching 0 **0** = DISABLE **1** = ENABLE
28:0	0x0	**PTV**: Timer Present Trigger Value. When TMRCR is written and EN is 1b, the internal counter is loaded with PTV+1 and starts counting down at each timing reference pulse. An interrupt is set when the internal counter reaches 0. Note that writing TMRCR with EN=1 while EN was already 1 restarts the counter. Once the timer is started, TARGET can be used to either - observe the value of the reference counter when the next interrupt will be generated (read) - select the exact reference counter value at which the next interrupt will be generated (write) **536870911** = MAX

TKE_FSI_TIMER_TMR1_TMRSR_0

Timer Status Register

Offset: 0x20004
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: TKE_FSI_SCR_TMRSCR1_0
Reset: 0x00000000 (0bx0x0,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Parity Protection	Reset	Description
30	RW	Y	0x0	**INTR_CLR**: Clears the interrupt when written as 1b, read always returns 0 **0** = DISABLE **1** = ENABLE
28:0	RO	N	0x0	**PCV**: Current counter value, writing to this field has no effect

TKE_FSI_TIMER_TMR1_TMRCSSR_0

Timer Clock Source Selection Register

Offset: 0x20008
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_TMRSCR1_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**SRC_ID**: Select Count reference 2'b11 - TSC [41:12] 2'b10 - TSC [29:0] 2'b01 - oscCnt [29:0] 2'b00 - usecCnt [29:0] The implementation of each timer uses comparison against counters incrementing at the selected rate. **0** = SRC_USECCNT **1** = SRC_OSCCNT **2** = SRC_TSCCNT_29_0 **3** = SRC_TSCCNT_41_12

TKE_FSI_TIMER_TMR1_TMRATR_0

Timer Absolute Target Register

Offset: 0x2000c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_TMRSCR1_0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:0	0x0	**TARGET**: Target value of the selected reference counter, i.e., when value of the selected reference equals target value, the interrupt is generated. The difference between TARGET and the current value must be positive when interpreting TARGET and the 29 LSB of the selected reference counter as signed integers, otherwise the interrupt triggers immediately.

TKE_FSI_TIMER_TMR2_TMRCR_0

```
Register name       Offset          Description
TMRCR{t}        BT + P*{t} + 0      Timer Configuration Register
TMRSR{t}        BT + P*{t} + 4      Timer Status Register
TMRCSSR{t}      BT + P*{t} + 8      Timer Clock Source Selection Register
TMRATR{t}       BT + P*{t} + 12     Timer Absolute Target Register
```

Timer Configuration Register

Offset: 0x30000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_TMRSCR2_0
Reset: 0x40000000 (0b01x0,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: Enable, when set, enables the countdown process. **0** = DISABLE **1** = ENABLE
30	0x1	**PER**: Periodic, when set, the internal counter is automatically reloaded after reaching 0 **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
28:0	0x0	**PTV**: Timer Present Trigger Value. When TMRCR is written and EN is 1b, the internal counter is loaded with PTV+1 and starts counting down at each timing reference pulse. An interrupt is set when the internal counter reaches 0. Note that writing TMRCR with EN=1 while EN was already 1 restarts the counter. Once the timer is started, TARGET can be used to either - observe the value of the reference counter when the next interrupt will be generated (read) - select the exact reference counter value at which the next interrupt will be generated (write) **536870911** = MAX

TKE_FSI_TIMER_TMR2_TMRSR_0

Timer Status Register

Offset: 0x30004
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: TKE_FSI_SCR_TMRSCR2_0
Reset: 0x00000000 (0bx0x0,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Parity Protection	Reset	Description
30	RW	Y	0x0	**INTR_CLR**: Clears the interrupt when written as 1b, read always returns 0 **0** = DISABLE **1** = ENABLE
28:0	RO	N	0x0	**PCV**: Current counter value, writing to this field has no effect

TKE_FSI_TIMER_TMR2_TMRCSSR_0

Timer Clock Source Selection Register

Offset: 0x30008
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: TKE_FSI_SCR_TMRSCR2_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**SRC_ID**: Select Count reference 2'b11 - TSC [41:12] 2'b10 - TSC [29:0] 2'b01 - oscCnt [29:0] 2'b00 - usecCnt [29:0] The implementation of each timer uses comparison against counters incrementing at the selected rate. **0** = SRC_USECCNT **1** = SRC_OSCCNT **2** = SRC_TSCCNT_29_0 **3** = SRC_TSCCNT_41_12

TKE_FSI_TIMER_TMR2_TMRATR_0

Timer Absolute Target Register

Offset: 0x3000c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_TMRSCR2_0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:0	0x0	**TARGET**: Target value of the selected reference counter, i.e., when value of the selected reference equals target value, the interrupt is generated. The difference between TARGET and the current value must be positive when interpreting TARGET and the 29 LSB of the selected reference counter as signed integers, otherwise the interrupt triggers immediately.

TKE_FSI_TIMER_TMR3_TMRCR_0

```
Register name       Offset          Description
TMRCR{t}        BT + P*{t} + 0      Timer Configuration Register
TMRSR{t}        BT + P*{t} + 4      Timer Status Register
TMRCSSR{t}      BT + P*{t} + 8      Timer Clock Source Selection Register
TMRATR{t}       BT + P*{t} + 12     Timer Absolute Target Register
```

Timer Configuration Register

Offset: 0x40000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_TMRSCR3_0
Reset: 0x40000000 (0b01x0,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: Enable, when set, enables the countdown process. **0** = DISABLE **1** = ENABLE
30	0x1	**PER**: Periodic, when set, the internal counter is automatically reloaded after reaching 0 **0** = DISABLE **1** = ENABLE
28:0	0x0	**PTV**: Timer Present Trigger Value. When TMRCR is written and EN is 1b, the internal counter is loaded with PTV+1 and starts counting down at each timing reference pulse. An interrupt is set when the internal counter reaches 0. Note that writing TMRCR with EN=1 while EN was already 1 restarts the counter. Once the timer is started, TARGET can be used to either - observe the value of the reference counter when the next interrupt will be generated (read) - select the exact reference counter value at which the next interrupt will be generated (write) **536870911** = MAX

TKE_FSI_TIMER_TMR3_TMRSR_0

Timer Status Register

Offset: 0x40004
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: TKE_FSI_SCR_TMRSCR3_0
Reset: 0x00000000 (0bx0x0,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Parity Protection	Reset	Description
30	RW	Y	0x0	**INTR_CLR**: Clears the interrupt when written as 1b, read always returns 0 **0** = DISABLE **1** = ENABLE
28:0	RO	N	0x0	**PCV**: Current counter value, writing to this field has no effect

TKE_FSI_TIMER_TMR3_TMRCSSR_0

Timer Clock Source Selection Register

Offset: 0x40008
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_TMRSCR3_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**SRC_ID**: Select Count reference 2'b11 - TSC [41:12] 2'b10 - TSC [29:0] 2'b01 - oscCnt [29:0] 2'b00 - usecCnt [29:0] The implementation of each timer uses comparison against counters incrementing at the selected rate. **0** = SRC_USECCNT **1** = SRC_OSCCNT **2** = SRC_TSCCNT_29_0 **3** = SRC_TSCCNT_41_12

TKE_FSI_TIMER_TMR3_TMRATR_0

Timer Absolute Target Register

Offset: 0x4000c
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: TKE_FSI_SCR_TMRSCR3_0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:0	0x0	**TARGET**: Target value of the selected reference counter, i.e., when value of the selected reference equals target value, the interrupt is generated. The difference between TARGET and the current value must be positive when interpreting TARGET and the 29 LSB of the selected reference counter as signed integers, otherwise the interrupt triggers immediately.

TKE_FSI_WDT0_WDTCR_0

```
WDTCR{w}    BW + P*{w} + 0  RW    Watchdog Timer Configuration Register
WDTSR{w}    BW + P*{w} + 4  RO    Watchdog Timer Status Register
WDTCMDR{w}  BW + P*{w} + 8  WO    Watchdog Timer Command Register
WDTUR{w}    BW + P*{w} + 12 RW    Watchdog Timer Unlock Register
```

Watchdog Timer Configuration Register

Offset: 0x50000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_WDTSCR0_0
Reset: 0x00700000 (0b0000,xx00,0111,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**WindowedRestartDisableMap**: When WindowedOperationEnable is 1b, restart commands attempted when expiration level has value b are rejected if bit b equals 1b in this field
25	0x0	**DisallowWDTFreeze**: When 1b, the signal watchdog_freeze_request has no effect When 0b, when the signal watchdog_freeze_request is asserted, the WDT stops decrementing CurrentCount if the WDT is enabled

Bit	Reset	Description
24:23	0x0	**RestartErrorAction**: If a restart action is attempted but fails, indicate which of four actions to take 0x0: do nothing, silently ignore the restart 0x1: immediately jump to expiration level 4 and reload the downcounter 0x2: immediately jump to expiration level 5 and reload the downcounter 0x3: immediately request a system power on reset (POR) When jumping levels, the normal actions configured per level must take place **0** = NOP **1** = JUMP_TO_4 **2** = JUMP_TO_5 **3** = POR_RESET
22:20	0x7	**ErrorThreshold**: The WDT logic asserts an error signal to HSM when ExpirationLevel >= ErrorThreshold.
19	0x0	**TscReferenceEnable**: Select as timing reference transitions on a configured TSC bit.
18	0x0	**ChallengeResponseEnable**: Enable the Challenge Response mode of operation. When 1b, the Unlock register operates as a Challenge Response register.
17	0x0	**WindowedOperationEnable**: Enable the windowed mode of operation. When 1b, a restart command is rejected when the current expiration level is identified as disallowed in the WindowedRestartDisableMap or the current time is below the window threshold. The rejection is either silent or performs a reset based on RestartErrorAction
16	0x0	**SystemPOResetEnable**: Enable internal full system reset at fifth expiration of the counter. This reset cannot be intercepted and acts as a Power On Reset **0** = DISABLE **1** = ENABLE
15	0x0	**SystemDebugResetEnable**: Enable systemwide reset assertion at fourth expiration of the counter. This reset can be intercepted by debug logic, e.g., to preserve information
14	0x0	**RemoteInterruptEnable**: Enable normal priority interrupt assertion at third expiration of the counter, connected to LIC for systemwide visibility
13	0x0	**LocalFIQEnable**: Enable high priority interrupt (FIQ) assertion at second expiration of the counter, connected to local interrupt controller
12	0x0	**LocalInterruptEnable**: Enable normal priority interrupt assertion at first expiration of the counter, connected to local interrupt controller

Bit	Reset	Description
11:4	0x0	**Period**: Measured in periods of the timer selected as source, this is the reload value, so 0 is treated as maximum period
3:0	0x0	**TimerSource**: The timer source interpretation is dependent on the value of TscReferenceEnable: TscReferenceEnable = 0b: timer<TimerSource> is the timing reference TscReferenceEnable = 1b: a transition of TSC[2*TimerSource] is the timing reference **0** = TMR0

TKE_FSI_WDT0_WDTSR_0

Watchdog Timer Status Register

Offset: 0x50004
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TKE_FSI_SCR_WDTSCR0_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,x000,0000,0000,0000)

Bit	Reset	Description
19	0x0	**Frozen**: 1b indicates that the WDT is frozen because of the assertion of the global WDT freeze request (when allowed) or because of the assertion of the global timer freeze request
18	0x0	**ErrorForced**: 1b indicates that the error reported to HSM is forced active for testing purposes
17	0x0	**StickyEnabled**: 1b when the counter is enabled and cannot be disabled, set by StickyStart command, cleared by reset only.
16	0x0	**CurrentError**: Current error reported to HSM, this is the output after combining the functional and forced error paths
14:12	0x0	**CurrentExpirationCount**: Current count of expiration since last start operation, cleared by Start command, saturating at 5. **0** = NO_EXPIRY **1** = LOCAL_IRQ **2** = LOCAL_FIQ **3** = REMOTE_IRQ **4** = DEBUG_RESET **5** = SYSTEM_RESET

Bit	Reset	Description
11:4	0x0	**CurrentCount**: Current value of the downcounter, reloaded with Period by Start command.
3	0x0	**RemoteInterruptStatus**: Current status of remote interrupt, cleared by Start command.
2	0x0	**LocalFIQStatus**: Current status of FIQ, cleared by Start command.
1	0x0	**LocalInterruptStatus**: Current status of local interrupt, cleared by Start command.
0	0x0	**Enabled**: 1b when the counter is active. When Enabled is 1b, the configuration of the WDT is frozen by silently ignoring write attempts to the configuration registers: WDTCR, WDTSCR and WDTFWCR. Set by Start or StickyStart commands, cleared by Disable when StickyEnabled is 0b. **0** = DISABLE **1** = ENABLE

TKE_FSI_WDT0_WDTCMDR_0

Watchdog Timer Command Register

Offset: 0x50008
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_WDTSCR0_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	0x0	**ForceError**: Only working if the Command register is unlocked. 1b sets an internal flop that is ORed with the functional error signal (based on threshold), 0b clears the bit. The state of the flop is shown in the status register.
2	0x0	**StickyStart**: Qualifies the StartCounter bit, if both bits are 1b, the counter is enabled in permanent fashion, i.e., it remains enabled until the next reset, and a disable counter operation has no effect. Ignored if StartCounter is 0b

Bit	Reset	Description
1	0x0	**DisableCounter**: Only working if the unlock register (WDTUR{w}) has been programmed before with the correct pattern. Writing to the command register always clears the unlock register. If written to 1b, while StartCounter is 0b and the unlock register contains the unlock pattern, the watchdog transitions back to disabled. **0** = DISABLE **1** = ENABLE
0	0x0	**StartCounter**: When written to 1b, enables the counter operation, loads the counter with Period and starts downcounting, resets the expiration count to 0 and clears all status flags. Enabled is always 1b after this. This is also used as Restart. Writing to StartCounter requires unlocking when ChallengeResponseEnable is 1b, and writing to StartCounter is also restricted to windows of time given by WindowedRestartDisableMap when WindowedOperationEnable is 1b. **0** = DISABLE **1** = ENABLE

TKE_FSI_WDT0_WDTUR_0

Watchdog Timer Unlock Register

Offset: 0x5000c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_WDTSCR0_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UnlockPattern**: The behavior of the Unlock pattern is dependent on the ChallengeResponseEnable configuration bit. When ChallengeResponseEnable is 0b, UnlockPattern must be written with value UNLOCK to allow a write to DisableCounter to take effect. UnlockPattern is reset at each write to the Command register. When ChallengeResponseEnable is 1b, a write to the Command register is only allowed after reading UnlockPattern to get the current value of an LFSR, followed by writing the next value of the LFSR to the UnlockPattern. If the two values match, the unlock is successful. In all cases the LFSR gets (re)initialized with the written value. **50266** = UNLOCK

TKE_FSI_WDT0_WDTSCR_0

Watchdog Timer Skip Configuration Register

Offset: 0x50010
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_WDTSCR0_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,x000,x000,x000)

Bit	Reset	Description
14:12	0x0	**Skip3**: Skip Value at Expiration count 3 At a Downcounter expiration, the expiration level changes as ExpC <= ExpC + 1 + Skip(ExpC), saturating at 5
10:8	0x0	**Skip2**: Skip Value at Expiration count 2
6:4	0x0	**Skip1**: Skip Value at Expiration count 1
2:0	0x0	**Skip0**: Skip Value at Expiration count 0

TKE_FSI_WDT0_WDTFWCR_0

Watchdog Timer Fine Window Configuration Register

Offset: 0x50014
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_WDTSCR0_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,x000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
18:16	0x0	**StartLevel**: If restart (i.e., start while WDT is already enabled) is attempted and windowing operation is enabled, restart is rejected if current expiration < StartLevel

Bit	Reset	Description
7:0	0x0	**StartCount**: If restart (i.e., start while WDT is already enabled) is attempted and windowing operation is enabled and current expiration level == StartLevel, restart is rejected if current downcounter > (Period - StartCount). StartCount is the number of cycles where rejection takes place at the start of that level.

TKE_FSI_WDT1_WDTCR_0

```
WDTCR{w}    BW + P*{w} + 0  RW    Watchdog Timer Configuration Register
WDTSR{w}    BW + P*{w} + 4  RO    Watchdog Timer Status Register
WDTCMDR{w}  BW + P*{w} + 8  WO    Watchdog Timer Command Register
WDTUR{w}    BW + P*{w} + 12 RW    Watchdog Timer Unlock Register
```

Watchdog Timer Configuration Register

Offset: 0x60000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_WDTSCR1_0
Reset: 0x00700000 (0b0000,xx00,0111,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**WindowedRestartDisableMap**: When WindowedOperationEnable is 1b, restart commands attempted when expiration level has value b are rejected if bit b equals 1b in this field
25	0x0	**DisallowWDTFreeze**: When 1b, the signal watchdog_freeze_request has no effect When 0b, when the signal watchdog_freeze_request is asserted, the WDT stops decrementing CurrentCount if the WDT is enabled
24:23	0x0	**RestartErrorAction**: If a restart action is attempted but fails, indicate which of four actions to take 0x0: do nothing, silently ignore the restart 0x1: immediately jump to expiration level 4 and reload the downcounter 0x2: immediately jump to expiration level 5 and reload the downcounter 0x3: immediately request a system POR When jumping levels, the normal actions configured per level must take place **0** = NOP **1** = JUMP_TO_4 **2** = JUMP_TO_5 **3** = POR_RESET

Bit	Reset	Description
22:20	0x7	**ErrorThreshold**: The WDT logic asserts an error signal to HSM when ExpirationLevel >= ErrorThreshold.
19	0x0	**TscReferenceEnable**: Select as timing reference transitions on a configured TSC bit.
18	0x0	**ChallengeResponseEnable**: Enable the Challenge Response mode of operation. When 1b, the Unlock register operates as a Challenge Response register.
17	0x0	**WindowedOperationEnable**: Enable the windowed mode of operation. When 1b, a restart command is rejected when the current expiration level is identified as disallowed in the WindowedRestartDisableMap or the current time is below the window threshold. The rejection is either silent or performs a reset based on RestartErrorAction
16	0x0	**SystemPOResetEnable**: Enable internal full system reset at fifth expiration of the counter. This reset cannot be intercepted and acts as a Power On Reset **0** = DISABLE **1** = ENABLE
15	0x0	**SystemDebugResetEnable**: Enable systemwide reset assertion at fourth expiration of the counter. This reset can be intercepted by debug logic, e.g., to preserve information
14	0x0	**RemoteInterruptEnable**: Enable normal priority interrupt assertion at third expiration of the counter, connected to LIC for systemwide visibility
13	0x0	**LocalFIQEnable**: Enable high priority interrupt (FIQ) assertion at second expiration of the counter, connected to local interrupt controller
12	0x0	**LocalInterruptEnable**: Enable normal priority interrupt assertion at first expiration of the counter, connected to local interrupt controller
11:4	0x0	**Period**: Measured in periods of the timer selected as source, this is the reload value, so 0 is treated as maximum period
3:0	0x0	**TimerSource**: The timer source interpretation is dependent on the value of TscReferenceEnable: TscReferenceEnable = 0b: timer<TimerSource> is the timing reference TscReferenceEnable = 1b: a transition of TSC[2*TimerSource] is the timing reference **0** = TMR0

TKE_FSI_WDT1_WDTSR_0

Watchdog Timer Status Register

Offset: 0x60004
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TKE_FSI_SCR_WDTSCR1_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,x000,0000,0000,0000)

Bit	Reset	Description
19	0x0	**Frozen**: 1b indicates that the WDT is frozen because of the assertion of the global WDT freeze request (when allowed) or because of the assertion of the global timer freeze request
18	0x0	**ErrorForced**: 1b indicates that the error reported to HSM is forced active for testing purposes
17	0x0	**StickyEnabled**: 1b when the counter is enabled and cannot be disabled, set by StickyStart command, cleared by reset only.
16	0x0	**CurrentError**: Current error reported to HSM, this is the output after combining the functional and forced error paths
14:12	0x0	**CurrentExpirationCount**: Current count of expiration since last start operation, cleared by Start command, saturating at 5. **0** = NO_EXPIRY **1** = LOCAL_IRQ **2** = LOCAL_FIQ **3** = REMOTE_IRQ **4** = DEBUG_RESET **5** = SYSTEM_RESET
11:4	0x0	**CurrentCount**: Current value of the downcounter, reloaded with Period by Start command.
3	0x0	**RemoteInterruptStatus**: Current status of remote interrupt, cleared by Start command.
2	0x0	**LocalFIQStatus**: Current status of FIQ, cleared by Start command.
1	0x0	**LocalInterruptStatus**: Current status of local interrupt, cleared by Start command.

Bit	Reset	Description
0	0x0	**Enabled**: 1b when the counter is active. When Enabled is 1b, the configuration of the WDT is frozen by silently ignoring write attempts to the configuration registers: WDTCR, WDTSCR and WDTFWCR. Set by Start or StickyStart commands, cleared by Disable when StickyEnabled is 0b. **0** = DISABLE **1** = ENABLE

TKE_FSI_WDT1_WDTCMDR_0

Watchdog Timer Command Register

Offset: 0x60008
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_WDTSCR1_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	0x0	**ForceError**: Only working if the Command register is unlocked. 1b sets an internal flop that is ORed with the functional error signal (based on threshold), 0b clears the bit. The state of the flop is shown in the status register.
2	0x0	**StickyStart**: Qualifies the StartCounter bit, if both bits are 1b, the counter is enabled in permanent fashion, i.e., it remains enabled until the next reset, and a disable counter operation has no effect. Ignored if StartCounter is 0b
1	0x0	**DisableCounter**: Only working if the unlock register (WDTUR{w}) has been programmed before with the correct pattern. Writing to the command register always clears the unlock register. If written to 1b, while StartCounter is 0b and the unlock register contains the unlock pattern, the watchdog transitions back to disabled. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
0	0x0	**StartCounter**: When written to 1b, enables the counter operation, loads the counter with Period and starts downcounting, resets the expiration count to 0 and clears all status flags. Enabled is always 1b after this. This is also used as Restart. Writing to StartCounter requires unlocking when ChallengeResponseEnable is 1b, and writing to StartCounter is also restricted to windows of time given by WindowedRestartDisableMap when WindowedOperationEnable is 1b. **0** = DISABLE **1** = ENABLE

TKE_FSI_WDT1_WDTUR_0

Watchdog Timer Unlock Register

Offset: 0x6000c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_WDTSCR1_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UnlockPattern**: The behavior of the Unlock pattern is dependent on the ChallengeResponseEnable configuration bit. When ChallengeResponseEnable is 0b, UnlockPattern must be written with value UNLOCK to allow a write to DisableCounter to take effect. UnlockPattern is reset at each write to the Command register. When ChallengeResponseEnable is 1b, a write to the Command register is only allowed after reading UnlockPattern to get the current value of an LFSR, followed by writing the next value of the LFSR to the UnlockPattern. If the two values match, the unlock is successful. In all cases the LFSR gets (re)initialized with the written value. **50266** = UNLOCK

TKE_FSI_WDT1_WDTSCR_0

Watchdog Timer Skip Configuration Register

Offset: 0x60010
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: TKE_FSI_SCR_WDTSCR1_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,x000,x000,x000)

Bit	Reset	Description
14:12	0x0	**Skip3**: Skip Value at Expiration count 3 At a Downcounter expiration, the expiration level changes as ExpC <= ExpC + 1 + Skip(ExpC), saturating at 5
10:8	0x0	**Skip2**: Skip Value at Expiration count 2
6:4	0x0	**Skip1**: Skip Value at Expiration count 1
2:0	0x0	**Skip0**: Skip Value at Expiration count 0

TKE_FSI_WDT1_WDTFWCR_0

Watchdog Timer Fine Window Configuration Register

Offset: 0x60014
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_WDTSCR1_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,x000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
18:16	0x0	**StartLevel**: If restart (i.e., start while WDT is already enabled) is attempted and windowing operation is enabled, restart is rejected if current expiration < StartLevel
7:0	0x0	**StartCount**: If restart (i.e., start while WDT is already enabled) is attempted and windowing operation is enabled and current expiration level == StartLevel, restart is rejected if current downcounter > (Period - StartCount). StartCount is the number of cycles where rejection takes place at the start of that level.

TKE_FSI_WDT2_WDTCR_0

```
WDTCR{w}    BW + P*{w} + 0  RW    Watchdog Timer Configuration Register
WDTSR{w}    BW + P*{w} + 4  RO    Watchdog Timer Status Register
```

```
WDTCMDR{w}  BW + P*{w} + 8  WO    Watchdog Timer Command Register
WDTUR{w}    BW + P*{w} + 12 RW    Watchdog Timer Unlock Register
```

Watchdog Timer Configuration Register

Offset: 0x70000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_WDTSCR2_0
Reset: 0x00700000 (0b0000,xx00,0111,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**WindowedRestartDisableMap**: When WindowedOperationEnable is 1b, restart commands attempted when expiration level has value b are rejected if bit b equals 1b in this field
25	0x0	**DisallowWDTFreeze**: When 1b, the signal watchdog_freeze_request has no effect When 0b, when the signal watchdog_freeze_request is asserted, the WDT stops decrementing CurrentCount if the WDT is enabled
24:23	0x0	**RestartErrorAction**: If a restart action is attempted but fails, indicate which of four actions to take 0x0: do nothing, silently ignore the restart 0x1: immediately jump to expiration level 4 and reload the downcounter 0x2: immediately jump to expiration level 5 and reload the downcounter 0x3: immediately request a system POR When jumping levels, the normal actions configured per level must take place **0** = NOP **1** = JUMP_TO_4 **2** = JUMP_TO_5 **3** = POR_RESET
22:20	0x7	**ErrorThreshold**: The WDT logic asserts an error signal to HSM when ExpirationLevel >= ErrorThreshold.
19	0x0	**TscReferenceEnable**: Select as timing reference transitions on a configured TSC bit.
18	0x0	**ChallengeResponseEnable**: Enable the Challenge Response mode of operation. When 1b, the Unlock register operates as a Challenge Response register.

Bit	Reset	Description
17	0x0	**WindowedOperationEnable**: Enable the windowed mode of operation. When 1b, a restart command is rejected when the current expiration level is identified as disallowed in the WindowedRestartDisableMap or the current time is below the window threshold. The rejection is either silent or performs a reset based on RestartErrorAction
16	0x0	**SystemPOResetEnable**: Enable internal full system reset at fifth expiration of the counter. This reset cannot be intercepted and acts as a Power On Reset **0** = DISABLE **1** = ENABLE
15	0x0	**SystemDebugResetEnable**: Enable systemwide reset assertion at fourth expiration of the counter. This reset can be intercepted by debug logic, e.g., to preserve information
14	0x0	**RemoteInterruptEnable**: Enable normal priority interrupt assertion at third expiration of the counter, connected to LIC for systemwide visibility
13	0x0	**LocalFIQEnable**: Enable high priority interrupt (FIQ) assertion at second expiration of the counter, connected to local interrupt controller
12	0x0	**LocalInterruptEnable**: Enable normal priority interrupt assertion at first expiration of the counter, connected to local interrupt controller
11:4	0x0	**Period**: Measured in periods of the timer selected as source, this is the reload value, so 0 is treated as maximum period
3:0	0x0	**TimerSource**: The timer source interpretation is dependent on the value of TscReferenceEnable: TscReferenceEnable = 0b: timer<TimerSource> is the timing reference TscReferenceEnable = 1b: a transition of TSC[2*TimerSource] is the timing reference **0** = TMR0

TKE_FSI_WDT2_WDTSR_0

Watchdog Timer Status Register

Offset: 0x70004
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: TKE_FSI_SCR_WDTSCR2_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,x000,0000,0000,0000)

Bit	Reset	Description
19	0x0	**Frozen**: 1b indicates that the WDT is frozen because of the assertion of the global WDT freeze request (when allowed) or because of the assertion of the global timer freeze request
18	0x0	**ErrorForced**: 1b indicates that the error reported to HSM is forced active for testing purposes
17	0x0	**StickyEnabled**: 1b when the counter is enabled and cannot be disabled, set by StickyStart command, cleared by reset only.
16	0x0	**CurrentError**: Current error reported to HSM, this is the output after combining the functional and forced error paths
14:12	0x0	**CurrentExpirationCount**: Current count of expiration since last start operation, cleared by Start command, saturating at 5. **0** = NO_EXPIRY **1** = LOCAL_IRQ **2** = LOCAL_FIQ **3** = REMOTE_IRQ **4** = DEBUG_RESET **5** = SYSTEM_RESET
11:4	0x0	**CurrentCount**: Current value of the downcounter, reloaded with Period by Start command.
3	0x0	**RemoteInterruptStatus**: Current status of remote interrupt, cleared by Start command.
2	0x0	**LocalFIQStatus**: Current status of FIQ, cleared by Start command.
1	0x0	**LocalInterruptStatus**: Current status of local interrupt, cleared by Start command.
0	0x0	**Enabled**: 1b when the counter is active. When Enabled is 1b, the configuration of the WDT is frozen by silently ignoring write attempts to the configuration registers: WDTCR, WDTSCR and WDTFWCR. Set by Start or StickyStart commands, cleared by Disable when StickyEnabled is 0b. **0** = DISABLE **1** = ENABLE

TKE_FSI_WDT2_WDTCMDR_0

Watchdog Timer Command Register

Offset: 0x70008
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_WDTSCR2_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	0x0	**ForceError**: Only working if the Command register is unlocked. 1b sets an internal flop that is ORed with the functional error signal (based on threshold), 0b clears the bit. The state of the flop is shown in the status register.
2	0x0	**StickyStart**: Qualifies the StartCounter bit, if both bits are 1b, the counter is enabled in permanent fashion, i.e., it remains enabled until the next reset, and a disable counter operation has no effect. Ignored if StartCounter is 0b
1	0x0	**DisableCounter**: Only working if the unlock register (WDTUR{w}) has been programmed before with the correct pattern. Writing to the command register always clears the unlock register. If written to 1b, while StartCounter is 0b and the unlock register contains the unlock pattern, the watchdog transitions back to disabled. **0** = DISABLE **1** = ENABLE
0	0x0	**StartCounter**: When written to 1b, enables the counter operation, loads the counter with Period and starts downcounting, resets the expiration count to 0 and clears all status flags. Enabled is always 1b after this. This is also used as Restart. Writing to StartCounter requires unlocking when ChallengeResponseEnable is 1b, and writing to StartCounter is also restricted to windows of time given by WindowedRestartDisableMap when WindowedOperationEnable is 1b. **0** = DISABLE **1** = ENABLE

TKE_FSI_WDT2_WDTUR_0

Watchdog Timer Unlock Register

Offset: 0x7000c
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: TKE_FSI_SCR_WDTSCR2_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UnlockPattern**: The behavior of the Unlock pattern is dependent on the ChallengeResponseEnable configuration bit. When ChallengeResponseEnable is 0b, UnlockPattern must be written with value UNLOCK to allow a write to DisableCounter to take effect. UnlockPattern is reset at each write to the Command register. When ChallengeResponseEnable is 1b, a write to the Command register is only allowed after reading UnlockPattern to get the current value of an LFSR, followed by writing the next value of the LFSR to the UnlockPattern. If the two values match, the unlock is successful. In all cases the LFSR gets (re)initialized with the written value. **50266** = UNLOCK

TKE_FSI_WDT2_WDTSCR_0

Watchdog Timer Skip Configuration Register

Offset: 0x70010
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_WDTSCR2_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,x000,x000,x000)

Bit	Reset	Description
14:12	0x0	**Skip3**: Skip Value at Expiration count 3 At a Downcounter expiration, the expiration level changes as ExpC <= ExpC + 1 + Skip(ExpC), saturating at 5
10:8	0x0	**Skip2**: Skip Value at Expiration count 2
6:4	0x0	**Skip1**: Skip Value at Expiration count 1
2:0	0x0	**Skip0**: Skip Value at Expiration count 0

TKE_FSI_WDT2_WDTFWCR_0

Watchdog Timer Fine Window Configuration Register

Offset: 0x70014
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_WDTSCR2_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,x000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
18:16	0x0	**StartLevel**: If restart (i.e., start while WDT is already enabled) is attempted and windowing operation is enabled, restart is rejected if current expiration < StartLevel
7:0	0x0	**StartCount**: If restart (i.e., start while WDT is already enabled) is attempted and windowing operation is enabled and current expiration level == StartLevel, restart is rejected if current downcounter > (Period - StartCount). StartCount is the number of cycles where rejection takes place at the start of that level.

TKE_FSI_WDT3_WDTCR_0

```
WDTCR{w}    BW + P*{w} + 0  RW    Watchdog Timer Configuration Register
WDTSR{w}    BW + P*{w} + 4  RO    Watchdog Timer Status Register
WDTCMDR{w}  BW + P*{w} + 8  WO    Watchdog Timer Command Register
WDTUR{w}    BW + P*{w} + 12 RW    Watchdog Timer Unlock Register
```

Watchdog Timer Configuration Register

Offset: 0x80000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_WDTSCR3_0
Reset: 0x00700000 (0b0000,xx00,0111,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**WindowedRestartDisableMap**: When WindowedOperationEnable is 1b, restart commands attempted when expiration level has value b are rejected if bit b equals 1b in this field
25	0x0	**DisallowWDTFreeze**: When 1b, the signal watchdog_freeze_request has no effect When 0b, when the signal watchdog_freeze_request is asserted, the WDT stops decrementing CurrentCount if the WDT is enabled

Bit	Reset	Description
24:23	0x0	**RestartErrorAction**: If a restart action is attempted but fails, indicate which of four actions to take 0x0: do nothing, silently ignore the restart 0x1: immediately jump to expiration level 4 and reload the downcounter 0x2: immediately jump to expiration level 5 and reload the downcounter 0x3: immediately request a system POR When jumping levels, the normal actions configured per level must take place **0** = NOP **1** = JUMP_TO_4 **2** = JUMP_TO_5 **3** = POR_RESET
22:20	0x7	**ErrorThreshold**: The WDT logic asserts an error signal to HSM when ExpirationLevel >= ErrorThreshold.
19	0x0	**TscReferenceEnable**: Select as timing reference transitions on a configured TSC bit.
18	0x0	**ChallengeResponseEnable**: Enable the Challenge Response mode of operation. When 1b, the Unlock register operates as a Challenge Response register.
17	0x0	**WindowedOperationEnable**: Enable the windowed mode of operation. When 1b, a restart command is rejected when the current expiration level is identified as disallowed in the WindowedRestartDisableMap or the current time is below the window threshold. The rejection is either silent or performs a reset based on RestartErrorAction
16	0x0	**SystemPOResetEnable**: Enable internal full system reset at fifth expiration of the counter. This reset cannot be intercepted and acts as a Power On Reset **0** = DISABLE **1** = ENABLE
15	0x0	**SystemDebugResetEnable**: Enable systemwide reset assertion at fourth expiration of the counter. This reset can be intercepted by debug logic, e.g., to preserve information
14	0x0	**RemoteInterruptEnable**: Enable normal priority interrupt assertion at third expiration of the counter, connected to LIC for systemwide visibility
13	0x0	**LocalFIQEnable**: Enable high priority interrupt (FIQ) assertion at second expiration of the counter, connected to local interrupt controller
12	0x0	**LocalInterruptEnable**: Enable normal priority interrupt assertion at first expiration of the counter, connected to local interrupt controller

Bit	Reset	Description
11:4	0x0	**Period**: Measured in periods of the timer selected as source, this is the reload value, so 0 is treated as maximum period
3:0	0x0	**TimerSource**: The timer source interpretation is dependent on the value of TscReferenceEnable: TscReferenceEnable = 0b: timer<TimerSource> is the timing reference TscReferenceEnable = 1b: a transition of TSC[2*TimerSource] is the timing reference **0** = TMR0

TKE_FSI_WDT3_WDTSR_0

Watchdog Timer Status Register

Offset: 0x80004
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TKE_FSI_SCR_WDTSCR3_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,x000,0000,0000,0000)

Bit	Reset	Description
19	0x0	**Frozen**: 1b indicates that the WDT is frozen because of the assertion of the global WDT freeze request (when allowed) or because of the assertion of the global timer freeze request
18	0x0	**ErrorForced**: 1b indicates that the error reported to HSM is forced active for testing purposes
17	0x0	**StickyEnabled**: 1b when the counter is enabled and cannot be disabled, set by StickyStart command, cleared by reset only.
16	0x0	**CurrentError**: Current error reported to HSM, this is the output after combining the functional and forced error paths
14:12	0x0	**CurrentExpirationCount**: Current count of expiration since last start operation, cleared by Start command, saturating at 5. **0** = NO_EXPIRY **1** = LOCAL_IRQ **2** = LOCAL_FIQ **3** = REMOTE_IRQ **4** = DEBUG_RESET **5** = SYSTEM_RESET

Bit	Reset	Description
11:4	0x0	**CurrentCount**: Current value of the downcounter, reloaded with Period by Start command.
3	0x0	**RemoteInterruptStatus**: Current status of remote interrupt, cleared by Start command.
2	0x0	**LocalFIQStatus**: Current status of FIQ, cleared by Start command.
1	0x0	**LocalInterruptStatus**: Current status of local interrupt, cleared by Start command.
0	0x0	**Enabled**: 1b when the counter is active. When Enabled is 1b, the configuration of the WDT is frozen by silently ignoring write attempts to the configuration registers: WDTCR, WDTSCR and WDTFWCR. Set by Start or StickyStart commands, cleared by Disable when StickyEnabled is 0b. **0** = DISABLE **1** = ENABLE

TKE_FSI_WDT3_WDTCMDR_0

Watchdog Timer Command Register

Offset: 0x80008
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_WDTSCR3_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	0x0	**ForceError**: Only working if the Command register is unlocked. 1b sets an internal flop that is ORed with the functional error signal (based on threshold), 0b clears the bit. The state of the flop is shown in the status register.
2	0x0	**StickyStart**: Qualifies the StartCounter bit, if both bits are 1b, the counter is enabled in permanent fashion, i.e., it remains enabled until the next reset, and a disable counter operation has no effect. Ignored if StartCounter is 0b

Bit	Reset	Description
1	0x0	**DisableCounter**: Only working if the unlock register (WDTUR{w}) has been programmed before with the correct pattern. Writing to the command register always clears the unlock register. If written to 1b, while StartCounter is 0b and the unlock register contains the unlock pattern, the watchdog transitions back to disabled. **0** = DISABLE **1** = ENABLE
0	0x0	**StartCounter**: When written to 1b, enables the counter operation, loads the counter with Period and starts downcounting, resets the expiration count to 0 and clears all status flags. Enabled is always 1b after this. This is also used as Restart. Writing to StartCounter requires unlocking when ChallengeResponseEnable is 1b, and writing to StartCounter is also restricted to windows of time given by WindowedRestartDisableMap when WindowedOperationEnable is 1b. **0** = DISABLE **1** = ENABLE

TKE_FSI_WDT3_WDTUR_0

Watchdog Timer Unlock Register

Offset: 0x8000c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_WDTSCR3_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UnlockPattern**: The behavior of the Unlock pattern is dependent on the ChallengeResponseEnable configuration bit. When ChallengeResponseEnable is 0b, UnlockPattern must be written with value UNLOCK to allow a write to DisableCounter to take effect. UnlockPattern is reset at each write to the Command register. When ChallengeResponseEnable is 1b, a write to the Command register is only allowed after reading UnlockPattern to get the current value of an LFSR, followed by writing the next value of the LFSR to the UnlockPattern. If the two values match, the unlock is successful. In all cases the LFSR gets (re)initialized with the written value. **50266** = UNLOCK

TKE_FSI_WDT3_WDTSCR_0

Watchdog Timer Skip Configuration Register

Offset: 0x80010
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_WDTSCR3_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,x000,x000,x000)

Bit	Reset	Description
14:12	0x0	**Skip3**: Skip Value at Expiration count 3 At a Downcounter expiration, the expiration level changes as ExpC <= ExpC + 1 + Skip(ExpC), saturating at 5
10:8	0x0	**Skip2**: Skip Value at Expiration count 2
6:4	0x0	**Skip1**: Skip Value at Expiration count 1
2:0	0x0	**Skip0**: Skip Value at Expiration count 0

TKE_FSI_WDT3_WDTFWCR_0

Watchdog Timer Fine Window Configuration Register

Offset: 0x80014
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_WDTSCR3_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,x000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
18:16	0x0	**StartLevel**: If restart (i.e., start while WDT is already enabled) is attempted and windowing operation is enabled, restart is rejected if current expiration < StartLevel

Bit	Reset	Description
7:0	0x0	**StartCount**: If restart (i.e, start while WDT is already enabled) is attempted and windowing operation is enabled and current expiration level == StartLevel, restart is rejected if current downcounter > (Period - StartCount). StartCount is the number of cycles where rejection takes place at the start of that level.

TKE_FSI_WDT4_WDTCR_0

```
WDTCR{w}   BW + P*{w} + 0  RW    Watchdog Timer Configuration Register
WDTSR{w}   BW + P*{w} + 4  RO    Watchdog Timer Status Register
WDTCMDR{w} BW + P*{w} + 8  WO    Watchdog Timer Command Register
WDTUR{w}   BW + P*{w} + 12 RW    Watchdog Timer Unlock Register
```

Watchdog Timer Configuration Register

Offset: 0x90000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_WDTSCR4_0
Reset: 0x00700000 (0b0000,xx00,0111,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**WindowedRestartDisableMap**: When WindowedOperationEnable is 1b, restart commands attempted when expiration level has value b are rejected if bit b equals 1b in this field
25	0x0	**DisallowWDTFreeze**: When 1b, the signal watchdog_freeze_request has no effect When 0b, when the signal watchdog_freeze_request is asserted, the WDT stops decrementing CurrentCount if the WDT is enabled
24:23	0x0	**RestartErrorAction**: If a restart action is attempted but fails, indicate which of four actions to take 0x0: do nothing, silently ignore the restart 0x1: immediately jump to expiration level 4 and reload the downcounter 0x2: immediately jump to expiration level 5 and reload the downcounter 0x3: immediately request a system POR When jumping levels, the normal actions configured per level must take place **0** = NOP **1** = JUMP_TO_4 **2** = JUMP_TO_5 **3** = POR_RESET

Bit	Reset	Description
22:20	0x7	**ErrorThreshold**: The WDT logic asserts an error signal to HSM when ExpirationLevel >= ErrorThreshold.
19	0x0	**TscReferenceEnable**: Select as timing reference transitions on a configured TSC bit.
18	0x0	**ChallengeResponseEnable**: Enable the Challenge Response mode of operation. When 1b, the Unlock register operates as a Challenge Response register.
17	0x0	**WindowedOperationEnable**: Enable the windowed mode of operation. When 1b, a restart command is rejected when the current expiration level is identified as disallowed in the WindowedRestartDisableMap or the current time is below the window threshold. The rejection is either silent or performs a reset based on RestartErrorAction
16	0x0	**SystemPOResetEnable**: Enable internal full system reset at fifth expiration of the counter. This reset cannot be intercepted and acts as a Power On Reset **0** = DISABLE **1** = ENABLE
15	0x0	**SystemDebugResetEnable**: Enable systemwide reset assertion at fourth expiration of the counter. This reset can be intercepted by debug logic, e.g., to preserve information
14	0x0	**RemoteInterruptEnable**: Enable normal priority interrupt assertion at third expiration of the counter, connected to LIC for systemwide visibility
13	0x0	**LocalFIQEnable**: Enable high priority interrupt (FIQ) assertion at second expiration of the counter, connected to local interrupt controller
12	0x0	**LocalInterruptEnable**: Enable normal priority interrupt assertion at first expiration of the counter, connected to local interrupt controller
11:4	0x0	**Period**: Measured in periods of the timer selected as source, this is the reload value, so 0 is treated as maximum period
3:0	0x0	**TimerSource**: The timer source interpretation is dependent on the value of TscReferenceEnable: TscReferenceEnable = 0b: timer<TimerSource> is the timing reference TscReferenceEnable = 1b: a transition of TSC[2*TimerSource] is the timing reference **0** = TMR0

TKE_FSI_WDT4_WDTSR_0

Watchdog Timer Status Register

Offset: 0x90004
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TKE_FSI_SCR_WDTSCR4_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,x000,0000,0000,0000)

Bit	Reset	Description
19	0x0	**Frozen**: 1b indicates that the WDT is frozen because of the assertion of the global WDT freeze request (when allowed) or because of the assertion of the global timer freeze request
18	0x0	**ErrorForced**: 1b indicates that the error reported to HSM is forced active for testing purposes
17	0x0	**StickyEnabled**: 1b when the counter is enabled and cannot be disabled, set by StickyStart command, cleared by reset only.
16	0x0	**CurrentError**: Current error reported to HSM, this is the output after combining the functional and forced error paths
14:12	0x0	**CurrentExpirationCount**: Current count of expiration since last start operation, cleared by Start command, saturating at 5. **0** = NO_EXPIRY **1** = LOCAL_IRQ **2** = LOCAL_FIQ **3** = REMOTE_IRQ **4** = DEBUG_RESET **5** = SYSTEM_RESET
11:4	0x0	**CurrentCount**: Current value of the downcounter, reloaded with Period by Start command.
3	0x0	**RemoteInterruptStatus**: Current status of remote interrupt, cleared by Start command.
2	0x0	**LocalFIQStatus**: Current status of FIQ, cleared by Start command.
1	0x0	**LocalInterruptStatus**: Current status of local interrupt, cleared by Start command.

Bit	Reset	Description
0	0x0	**Enabled**: 1b when the counter is active. When Enabled is 1b, the configuration of the WDT is frozen by silently ignoring write attempts to the configuration registers: WDTCR, WDTSCR and WDTFWCR. Set by Start or StickyStart commands, cleared by Disable when StickyEnabled is 0b. **0** = DISABLE **1** = ENABLE

TKE_FSI_WDT4_WDTCMDR_0

Watchdog Timer Command Register

Offset: 0x90008
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_WDTSCR4_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	0x0	**ForceError**: Only working if the Command register is unlocked. 1b sets an internal flop that is ORed with the functional error signal (based on threshold), 0b clears the bit. The state of the flop is shown in the status register.
2	0x0	**StickyStart**: Qualifies the StartCounter bit, if both bits are 1b, the counter is enabled in permanent fashion, i.e., it remains enabled until the next reset, and a disable counter operation has no effect. Ignored if StartCounter is 0b
1	0x0	**DisableCounter**: Only working if the unlock register (WDTUR{w}) has been programmed before with the correct pattern. Writing to the command register always clears the unlock register. If written to 1b, while StartCounter is 0b and the unlock register contains the unlock pattern, the watchdog transitions back to disabled. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
0	0x0	**StartCounter**: When written to 1b, enables the counter operation, loads the counter with Period and starts downcounting, resets the expiration count to 0 and clears all status flags. Enabled is always 1b after this. This is also used as Restart. Writing to StartCounter requires unlocking when ChallengeResponseEnable is 1b, and writing to StartCounter is also restricted to windows of time given by WindowedRestartDisableMap when WindowedOperationEnable is 1b. **0** = DISABLE **1** = ENABLE

TKE_FSI_WDT4_WDTUR_0

Watchdog Timer Unlock Register

Offset: 0x9000c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_WDTSCR4_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UnlockPattern**: The behavior of the Unlock pattern is dependent on the ChallengeResponseEnable configuration bit. When ChallengeResponseEnable is 0b, UnlockPattern must be written with value UNLOCK to allow a write to DisableCounter to take effect. UnlockPattern is reset at each write to the Command register. When ChallengeResponseEnable is 1b, a write to the Command register is only allowed after reading UnlockPattern to get the current value of an LFSR, followed by writing the next value of the LFSR to the UnlockPattern. If the two values match, the unlock is successful. In all cases the LFSR gets (re)initialized with the written value. **50266** = UNLOCK

TKE_FSI_WDT4_WDTSCR_0

Watchdog Timer Skip Configuration Register

Offset: 0x90010
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: TKE_FSI_SCR_WDTSCR4_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,x000,x000,x000)

Bit	Reset	Description
14:12	0x0	**Skip3**: Skip Value at Expiration count 3 At a Downcounter expiration, the expiration level changes as ExpC <= ExpC + 1 + Skip(ExpC), saturating at 5
10:8	0x0	**Skip2**: Skip Value at Expiration count 2
6:4	0x0	**Skip1**: Skip Value at Expiration count 1
2:0	0x0	**Skip0**: Skip Value at Expiration count 0

TKE_FSI_WDT4_WDTFWCR_0

Watchdog Timer Fine Window Configuration Register

Offset: 0x90014
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_FSI_SCR_WDTSCR4_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,x000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
18:16	0x0	**StartLevel**: If restart (i.e., start while WDT is already enabled) is attempted and windowing operation is enabled, restart is rejected if current expiration < StartLevel
7:0	0x0	**StartCount**: If restart (i.e., start while WDT is already enabled) is attempted and windowing operation is enabled and current expiration level == StartLevel, restart is rejected if current downcounter > (Period - StartCount). StartCount is the number of cycles where rejection takes place at the start of that level.

8.3.4.2 Time Keeping Element: TOP Registers

TKE_TOP_SHARED_TKETSC0_0

Value of Local TSC counter, synchronized across SOC

Offset: 0x0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TKE_TOP_SCR_TKESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CV**: Current value of TSC, TSC[31:0]

TKE_TOP_SHARED_TKETSC1_0

Value of Master TSC counter, synchronized across SOC

Offset: 0x4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TKE_TOP_SCR_TKESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CV**: 8h0 followed by Master TSC[55:32]

TKE_TOP_SHARED_TKEUSEC_0

Value of Local USEC counter, not synchronized across SOC

Offset: 0x8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TKE_TOP_SCR_TKESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CV**: Current value of microsecond counter

TKE_TOP_SHARED_TKEOSC_0

Value of Local OSC counter, not synchronized across SOC

Offset: 0xc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TKE_TOP_SCR_TKESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CV**: Current value of oscillator counter

TKE_TOP_SHARED_TKECR_0

TKE Control register

Offset: 0x10
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TKESCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	0x0	**HDBG**: Halt-on-debug. Controls whether a Halt-on-debug signal halts TKE logic. 0: TKE ignores Halt-on-debug. 1: If TKE reference is local USEC/OSC counter, asserting this field disables the update of the reference counters and reference timing signals for all timers and all WDTs
2	0x0	**OSC_DIS**: Disable the local OSC counter - Can be used to decrease the power consumption if OSC is not required either by hardware as a reference clock or by software
1	0x0	**USEC_DIS**: Disable the local USEC counter - Can be used to decrease the power consumption if USEC is not required either by hardware as a reference clock or by software
0	0x0	**TSC_DIS**: If TKE reference used is TSC, setting this disables the TSC capture locally. This disables the update of the reference counters and reference timing signals for all timers and all WDTs.

TKE_TOP_SHARED_CLK_OVR_ON_0

Offset: 0x14
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TKESCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**OSC**: SLCG override bit **0** = DISABLE **1** = ENABLE

TKE_TOP_SHARED_TKEIE0_0

```
================================================================================================
==
Optional, controls the routing of shared interrupt {i}: 0 to 9
  BITMAPPING:
   15:0 of register TKEIE{i} = Timer Single Interrupt, for Timers 15 - 0.
  18:16 of register TKEIE{i} = Watchdog Timer 0 internal interrupt status
  19:19 of register TKEIE{i} = Reserved
  22:20 of register TKEIE{i} = Watchdog Timer 1 internal interrupt status
  23:23 of register TKEIE{i} = Reserved
  26:24 of register TKEIE{i} = Watchdog Timer 2 internal interrupt status
  27:27 of register TKEIE{i} = Reserved
  30:28 of register TKEIE{i} = Watchdog Timer 3 internal interrupt status
  31:31 of register TKEIE{i} = Reserved
================================================================================================
==
```

Offset: 0x100
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TKESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EnableMap**: Register for Shared interrupt 0, routing control

TKE_TOP_SHARED_TKEIE1_0

Offset: 0x104
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TKESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EnableMap**: Register for Shared interrupt 1, routing control

TKE_TOP_SHARED_TKEIE2_0

Offset: 0x108
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TKESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EnableMap**: Register for Shared interrupt 2, routing control

TKE_TOP_SHARED_TKEIE3_0

Offset: 0x10c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TKESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EnableMap**: Register for Shared interrupt 3, routing control

TKE_TOP_SHARED_TKEIE4_0

Offset: 0x110
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: TKE_TOP_SCR_TKESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EnableMap**: Register for Shared interrupt 4, routing control

TKE_TOP_SHARED_TKEIE5_0
Offset: 0x114
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TKESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EnableMap**: Register for Shared interrupt 5, routing control

TKE_TOP_SHARED_TKEIE6_0
Offset: 0x118
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TKESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EnableMap**: Register for Shared interrupt 6, routing control

TKE_TOP_SHARED_TKEIE7_0
Offset: 0x11c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TKESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EnableMap**: Register for Shared interrupt 7, routing control

TKE_TOP_SHARED_TKEIE8_0

Offset: 0x120
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TKESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EnableMap**: Register for Shared interrupt 8, routing control

TKE_TOP_SHARED_TKEIE9_0

Offset: 0x124
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TKESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EnableMap**: Register for Shared interrupt 9, routing control

TKE_TOP_SHARED_TKEIE10_0

Offset: 0x128
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TKESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EnableMap**: Register for Shared interrupt 10, routing control

TKE_TOP_SHARED_TKEIE11_0
Offset: 0x12c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TKESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EnableMap**: Register for Shared interrupt 11, routing control

TKE_TOP_SHARED_TKEIE12_0
Offset: 0x130
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TKESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EnableMap**: Register for Shared interrupt 12, routing control

TKE_TOP_SHARED_TKEIE13_0
Offset: 0x134
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TKESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EnableMap**: Register for Shared interrupt 13, routing control

TKE_TOP_SHARED_TKEIE14_0
Offset: 0x138

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TKESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EnableMap**: Register for Shared interrupt 14, routing control

TKE_TOP_SHARED_TKEIE15_0

Offset: 0x13c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TKESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EnableMap**: Register for Shared interrupt 15, routing control

TKE_TOP_SHARED_TKEIV_0

```
================================================================
Indicates which shared interrupts are currently asserted
BITMAPPING:
      0:0   -   Current Status of external interrupt 0
      1:1   -   Current Status of external interrupt 1
      2:2   -   Current Status of external interrupt 2
      3:3   -   Current Status of external interrupt 3
      4:4   -   Current Status of external interrupt 4
      5:5   -   Current Status of external interrupt 5
      6:6   -   Current Status of external interrupt 6
      7:7   -   Current Status of external interrupt 7
      8:8   -   Current Status of external interrupt 8
      9:9   -   Current Status of external interrupt 9
================================================================
```

Offset: 0x200
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TKE_TOP_SCR_TKESCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**ValidMap**: Currently enabled interrupts

TKE_TOP_SHARED_TKEIR_0

Which internal interrupts are currently asserted, before applying the TKEIE masks

Offset: 0x204
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TKE_TOP_SCR_TKESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**RawMap**: Currently asserted internal interrupts The mapping of the Internal Interrupts is defined as follows: Bits[00 + [nT-1:0]] are the interrupts of the nT instantiated timers Bits[16 + {w} * 4 + [2:0]] are the sets of three interrupts of the nW instantiated watchdog timers

TKE_TOP_TIMER_TMR0_TMRCR_0

```
Register name       Offset          Description
TMRCR{t}        BT + P*{t} + 0      Timer Configuration Register
TMRSR{t}        BT + P*{t} + 4      Timer Status Register
TMRCSSR{t}      BT + P*{t} + 8      Timer Clock Source Selection Register
TMRATR{t}       BT + P*{t} + 12     Timer Absolute Target Register
```

Timer Configuration Register

Offset: 0x10000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR0_0
Reset: 0x40000000 (0b01x0,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: Enable, when set, enables the countdown process. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
30	0x1	**PER**: Periodic, when set, the internal counter is automatically reloaded after reaching 0 **0** = DISABLE **1** = ENABLE
28:0	0x0	**PTV**: Timer Present Trigger Value. When TMRCR is written and EN is 1b, the internal counter is loaded with PTV+1 and starts counting down at each timing reference pulse. An interrupt is set when the internal counter reaches 0. Note that writing TMRCR with EN=1 while EN was already 1 restarts the counter. Once the timer is started, TARGET can be used to either - observe the value of the reference counter when the next interrupt will be generated (read) - select the exact reference counter value at which the next interrupt will be generated (write) **536870911** = MAX

TKE_TOP_TIMER_TMR0_TMRSR_0

Timer Status Register

Offset: 0x10004
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR0_0
Reset: 0x00000000 (0bx0x0,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Parity Protection	Reset	Description
30	RW	Y	0x0	**INTR_CLR**: Clears the interrupt when written as 1b, read always returns 0 **0** = DISABLE **1** = ENABLE
28:0	RO	N	0x0	**PCV**: Current counter value, writing to this field has no effect

TKE_TOP_TIMER_TMR0_TMRCSSR_0

Timer Clock Source Selection Register

Offset: 0x10008
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR0_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**SRC_ID**: Select Count reference 2'b11 - TSC [41:12] 2'b10 - TSC [29:0] 2'b01 - oscCnt [29:0] 2'b00 - usecCnt [29:0] The implementation of each timer uses comparison against counters incrementing at the selected rate. **0** = SRC_USECCNT **1** = SRC_OSCCNT **2** = SRC_TSCCNT_29_0 **3** = SRC_TSCCNT_41_12

TKE_TOP_TIMER_TMR0_TMRATR_0

Timer Absolute Target Register

Offset: 0x1000c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR0_0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:0	0x0	**TARGET**: Target value of the selected reference counter, i.e., when value of the selected reference equals target value, the interrupt is generated. The difference between TARGET and the current value must be positive when interpreting TARGET and the 29 LSB of the selected reference counter as signed integers, otherwise the interrupt triggers immediately.

TKE_TOP_TIMER_TMR1_TMRCR_0

```
Register name       Offset          Description
TMRCR{t}        BT + P*{t} + 0      Timer Configuration Register
TMRSR{t}        BT + P*{t} + 4      Timer Status Register
```

```
TMRCSSR{t}      BT + P*{t} + 8      Timer Clock Source Selection Register
TMRATR{t}       BT + P*{t} + 12     Timer Absolute Target Register
```

Timer Configuration Register

Offset: 0x20000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR1_0
Reset: 0x40000000 (0b01x0,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: Enable, when set, enables the countdown process. **0** = DISABLE **1** = ENABLE
30	0x1	**PER**: Periodic, when set, the internal counter is automatically reloaded after reaching 0 **0** = DISABLE **1** = ENABLE
28:0	0x0	**PTV**: Timer Present Trigger Value. When TMRCR is written and EN is 1b, the internal counter is loaded with PTV+1 and starts counting down at each timing reference pulse. An interrupt is set when the internal counter reaches 0. Note that writing TMRCR with EN=1 while EN was already 1 restarts the counter. Once the timer is started, TARGET can be used to either - observe the value of the reference counter when the next interrupt will be generated (read) - select the exact reference counter value at which the next interrupt will be generated (write) **536870911** = MAX

TKE_TOP_TIMER_TMR1_TMRSR_0

Timer Status Register

Offset: 0x20004
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR1_0
Reset: 0x00000000 (0bx0x0,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Parity Protection	Reset	Description
30	RW	Y	0x0	**INTR_CLR**: Clears the interrupt when written as 1b, read always returns 0 **0** = DISABLE **1** = ENABLE
28:0	RO	N	0x0	**PCV**: Current counter value, writing to this field has no effect

TKE_TOP_TIMER_TMR1_TMRCSSR_0

Timer Clock Source Selection Register

Offset: 0x20008
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR1_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**SRC_ID**: Select Count reference 2'b11 - TSC [41:12] 2'b10 - TSC [29:0] 2'b01 - oscCnt [29:0] 2'b00 - usecCnt [29:0] The implementation of each timer uses comparison against counters incrementing at the selected rate. **0** = SRC_USECCNT **1** = SRC_OSCCNT **2** = SRC_TSCCNT_29_0 **3** = SRC_TSCCNT_41_12

TKE_TOP_TIMER_TMR1_TMRATR_0

Timer Absolute Target Register

Offset: 0x2000c
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: TKE_TOP_SCR_TMRSCR1_0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:0	0x0	**TARGET**: Target value of the selected reference counter, i.e., when value of the selected reference equals target value, the interrupt is generated. The difference between TARGET and the current value must be positive when interpreting TARGET and the 29 LSB of the selected reference counter as signed integers, otherwise the interrupt triggers immediately.

TKE_TOP_TIMER_TMR2_TMRCR_0

```
Register name       Offset          Description
TMRCR{t}        BT + P*{t} + 0      Timer Configuration Register
TMRSR{t}        BT + P*{t} + 4      Timer Status Register
TMRCSSR{t}      BT + P*{t} + 8      Timer Clock Source Selection Register
TMRATR{t}       BT + P*{t} + 12     Timer Absolute Target Register
```

Timer Configuration Register

Offset: 0x30000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR2_0
Reset: 0x40000000 (0b01x0,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: Enable, when set, enables the countdown process. **0** = DISABLE **1** = ENABLE
30	0x1	**PER**: Periodic, when set, the internal counter is automatically reloaded after reaching 0 **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
28:0	0x0	**PTV**: Timer Present Trigger Value. When TMRCR is written and EN is 1b, the internal counter is loaded with PTV+1 and starts counting down at each timing reference pulse. An interrupt is set when the internal counter reaches 0. Note that writing TMRCR with EN=1 while EN was already 1 restarts the counter. Once the timer is started, TARGET can be used to either - observe the value of the reference counter when the next interrupt will be generated (read) - select the exact reference counter value at which the next interrupt will be generated (write) **536870911** = MAX

TKE_TOP_TIMER_TMR2_TMRSR_0

Timer Status Register

Offset: 0x30004
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR2_0
Reset: 0x00000000 (0bx0x0,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Parity Protection	Reset	Description
30	RW	Y	0x0	**INTR_CLR**: Clears the interrupt when written as 1b, read always returns 0 **0** = DISABLE **1** = ENABLE
28:0	RO	N	0x0	**PCV**: Current counter value, writing to this field has no effect

TKE_TOP_TIMER_TMR2_TMRCSSR_0

Timer Clock Source Selection Register

Offset: 0x30008
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: TKE_TOP_SCR_TMRSCR2_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**SRC_ID**: Select Count reference 2'b11 - TSC [41:12] 2'b10 - TSC [29:0] 2'b01 - oscCnt [29:0] 2'b00 - usecCnt [29:0] The implementation of each timer uses comparison against counters incrementing at the selected rate. **0** = SRC_USECCNT **1** = SRC_OSCCNT **2** = SRC_TSCCNT_29_0 **3** = SRC_TSCCNT_41_12

TKE_TOP_TIMER_TMR2_TMRATR_0

Timer Absolute Target Register

Offset: 0x3000c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR2_0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:0	0x0	**TARGET**: Target value of the selected reference counter, i.e., when value of the selected reference equals target value, the interrupt is generated. The difference between TARGET and the current value must be positive when interpreting TARGET and the 29 LSB of the selected reference counter as signed integers, otherwise the interrupt triggers immediately.

TKE_TOP_TIMER_TMR3_TMRCR_0

```
Register name       Offset          Description
TMRCR{t}        BT + P*{t} + 0      Timer Configuration Register
TMRSR{t}        BT + P*{t} + 4      Timer Status Register
TMRCSSR{t}      BT + P*{t} + 8      Timer Clock Source Selection Register
TMRATR{t}       BT + P*{t} + 12     Timer Absolute Target Register
```

Timer Configuration Register

Offset: 0x40000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR3_0
Reset: 0x40000000 (0b01x0,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: Enable, when set, enables the countdown process. **0** = DISABLE **1** = ENABLE
30	0x1	**PER**: Periodic, when set, the internal counter is automatically reloaded after reaching 0 **0** = DISABLE **1** = ENABLE
28:0	0x0	**PTV**: Timer Present Trigger Value. When TMRCR is written and EN is 1b, the internal counter is loaded with PTV+1 and starts counting down at each timing reference pulse. An interrupt is set when the internal counter reaches 0. Note that writing TMRCR with EN=1 while EN was already 1 restarts the counter. Once the timer is started, TARGET can be used to either - observe the value of the reference counter when the next interrupt will be generated (read) - select the exact reference counter value at which the next interrupt will be generated (write) **536870911** = MAX

TKE_TOP_TIMER_TMR3_TMRSR_0

Timer Status Register

Offset: 0x40004
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR3_0
Reset: 0x00000000 (0bx0x0,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Parity Protection	Reset	Description
30	RW	Y	0x0	**INTR_CLR**: Clears the interrupt when written as 1b, read always returns 0 **0** = DISABLE **1** = ENABLE
28:0	RO	N	0x0	**PCV**: Current counter value, writing to this field has no effect

TKE_TOP_TIMER_TMR3_TMRCSSR_0

Timer Clock Source Selection Register

Offset: 0x40008
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR3_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**SRC_ID**: Select Count reference 2'b11 - TSC [41:12] 2'b10 - TSC [29:0] 2'b01 - oscCnt [29:0] 2'b00 - usecCnt [29:0] The implementation of each timer uses comparison against counters incrementing at the selected rate. **0** = SRC_USECCNT **1** = SRC_OSCCNT **2** = SRC_TSCCNT_29_0 **3** = SRC_TSCCNT_41_12

TKE_TOP_TIMER_TMR3_TMRATR_0

Timer Absolute Target Register

Offset: 0x4000c
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: TKE_TOP_SCR_TMRSCR3_0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:0	0x0	**TARGET**: Target value of the selected reference counter, i.e., when value of the selected reference equals target value, the interrupt is generated. The difference between TARGET and the current value must be positive when interpreting TARGET and the 29 LSB of the selected reference counter as signed integers, otherwise the interrupt triggers immediately.

TKE_TOP_TIMER_TMR4_TMRCR_0

```
Register name       Offset          Description
TMRCR{t}        BT + P*{t} + 0      Timer Configuration Register
TMRSR{t}        BT + P*{t} + 4      Timer Status Register
TMRCSSR{t}      BT + P*{t} + 8      Timer Clock Source Selection Register
TMRATR{t}       BT + P*{t} + 12     Timer Absolute Target Register
```

Timer Configuration Register

Offset: 0x50000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR4_0
Reset: 0x40000000 (0b01x0,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: Enable, when set, enables the countdown process. **0** = DISABLE **1** = ENABLE
30	0x1	**PER**: Periodic, when set, the internal counter is automatically reloaded after reaching 0 **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
28:0	0x0	**PTV**: Timer Present Trigger Value. When TMRCR is written and EN is 1b, the internal counter is loaded with PTV+1 and starts counting down at each timing reference pulse. An interrupt is set when the internal counter reaches 0. Note that writing TMRCR with EN=1 while EN was already 1 restarts the counter. Once the timer is started, TARGET can be used to either - observe the value of the reference counter when the next interrupt will be generated (read) - select the exact reference counter value at which the next interrupt will be generated (write) **536870911** = MAX

TKE_TOP_TIMER_TMR4_TMRSR_0

Timer Status Register

Offset: 0x50004
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR4_0
Reset: 0x00000000 (0bx0x0,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Parity Protection	Reset	Description
30	RW	Y	0x0	**INTR_CLR**: Clears the interrupt when written as 1b, read always returns 0 **0** = DISABLE **1** = ENABLE
28:0	RO	N	0x0	**PCV**: Current counter value, writing to this field has no effect

TKE_TOP_TIMER_TMR4_TMRCSSR_0

Timer Clock Source Selection Register

Offset: 0x50008
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: TKE_TOP_SCR_TMRSCR4_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**SRC_ID**: Select Count reference 2'b11 - TSC [41:12] 2'b10 - TSC [29:0] 2'b01 - oscCnt [29:0] 2'b00 - usecCnt [29:0] The implementation of each timer uses comparison against counters incrementing at the selected rate. **0** = SRC_USECCNT **1** = SRC_OSCCNT **2** = SRC_TSCCNT_29_0 **3** = SRC_TSCCNT_41_12

TKE_TOP_TIMER_TMR4_TMRATR_0

Timer Absolute Target Register

Offset: 0x5000c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR4_0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:0	0x0	**TARGET**: Target value of the selected reference counter, i.e., when value of the selected reference equals target value, the interrupt is generated. The difference between TARGET and the current value must be positive when interpreting TARGET and the 29 LSB of the selected reference counter as signed integers, otherwise the interrupt triggers immediately.

TKE_TOP_TIMER_TMR5_TMRCR_0

```
Register name       Offset          Description
TMRCR{t}        BT + P*{t} + 0      Timer Configuration Register
TMRSR{t}        BT + P*{t} + 4      Timer Status Register
TMRCSSR{t}      BT + P*{t} + 8      Timer Clock Source Selection Register
TMRATR{t}       BT + P*{t} + 12     Timer Absolute Target Register
```

Timer Configuration Register

Offset: 0x60000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR5_0
Reset: 0x40000000 (0b01x0,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: Enable, when set, enables the countdown process. **0** = DISABLE **1** = ENABLE
30	0x1	**PER**: Periodic, when set, the internal counter is automatically reloaded after reaching 0 **0** = DISABLE **1** = ENABLE
28:0	0x0	**PTV**: Timer Present Trigger Value. When TMRCR is written and EN is 1b, the internal counter is loaded with PTV+1 and starts counting down at each timing reference pulse. An interrupt is set when the internal counter reaches 0. Note that writing TMRCR with EN=1 while EN was already 1 restarts the counter. Once the timer is started, TARGET can be used to either - observe the value of the reference counter when the next interrupt will be generated (read) - select the exact reference counter value at which the next interrupt will be generated (write) **536870911** = MAX

TKE_TOP_TIMER_TMR5_TMRSR_0

Timer Status Register

Offset: 0x60004
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR5_0
Reset: 0x00000000 (0bx0x0,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Parity Protection	Reset	Description
30	RW	Y	0x0	**INTR_CLR**: Clears the interrupt when written as 1b, read always returns 0 **0** = DISABLE **1** = ENABLE
28:0	RO	N	0x0	**PCV**: Current counter value, writing to this field has no effect

TKE_TOP_TIMER_TMR5_TMRCSSR_0

Timer Clock Source Selection Register

Offset: 0x60008
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR5_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**SRC_ID**: Select Count reference 2'b11 - TSC [41:12] 2'b10 - TSC [29:0] 2'b01 - oscCnt [29:0] 2'b00 - usecCnt [29:0] The implementation of each timer uses comparison against counters incrementing at the selected rate. **0** = SRC_USECCNT **1** = SRC_OSCCNT **2** = SRC_TSCCNT_29_0 **3** = SRC_TSCCNT_41_12

TKE_TOP_TIMER_TMR5_TMRATR_0

Timer Absolute Target Register

Offset: 0x6000c
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: TKE_TOP_SCR_TMRSCR5_0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:0	0x0	**TARGET**: Target value of the selected reference counter, i.e., when value of the selected reference equals target value, the interrupt is generated. The difference between TARGET and the current value must be positive when interpreting TARGET and the 29 LSB of the selected reference counter as signed integers, otherwise the interrupt triggers immediately.

TKE_TOP_TIMER_TMR6_TMRCR_0

```
Register name       Offset          Description
TMRCR{t}        BT + P*{t} + 0      Timer Configuration Register
TMRSR{t}        BT + P*{t} + 4      Timer Status Register
TMRCSSR{t}      BT + P*{t} + 8      Timer Clock Source Selection Register
TMRATR{t}       BT + P*{t} + 12     Timer Absolute Target Register
```

Timer Configuration Register

Offset: 0x70000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR6_0
Reset: 0x40000000 (0b01x0,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: Enable, when set, enables the countdown process. **0** = DISABLE **1** = ENABLE
30	0x1	**PER**: Periodic, when set, the internal counter is automatically reloaded after reaching 0 **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
28:0	0x0	**PTV**: Timer Present Trigger Value. When TMRCR is written and EN is 1b, the internal counter is loaded with PTV+1 and starts counting down at each timing reference pulse. An interrupt is set when the internal counter reaches 0. Note that writing TMRCR with EN=1 while EN was already 1 restarts the counter. Once the timer is started, TARGET can be used to either - observe the value of the reference counter when the next interrupt will be generated (read) - select the exact reference counter value at which the next interrupt will be generated (write) **536870911** = MAX

TKE_TOP_TIMER_TMR6_TMRSR_0

Timer Status Register

Offset: 0x70004
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR6_0
Reset: 0x00000000 (0bx0x0,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Parity Protection	Reset	Description
30	RW	Y	0x0	**INTR_CLR**: Clears the interrupt when written as 1b, read always returns 0 **0** = DISABLE **1** = ENABLE
28:0	RO	N	0x0	**PCV**: Current counter value, writing to this field has no effect

TKE_TOP_TIMER_TMR6_TMRCSSR_0

Timer Clock Source Selection Register

Offset: 0x70008
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: TKE_TOP_SCR_TMRSCR6_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**SRC_ID**: Select Count reference 2'b11 - TSC [41:12] 2'b10 - TSC [29:0] 2'b01 - oscCnt [29:0] 2'b00 - usecCnt [29:0] The implementation of each timer uses comparison against counters incrementing at the selected rate. **0** = SRC_USECCNT **1** = SRC_OSCCNT **2** = SRC_TSCCNT_29_0 **3** = SRC_TSCCNT_41_12

TKE_TOP_TIMER_TMR6_TMRATR_0

Timer Absolute Target Register

Offset: 0x7000c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR6_0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:0	0x0	**TARGET**: Target value of the selected reference counter, i.e., when value of the selected reference equals target value, the interrupt is generated. The difference between TARGET and the current value must be positive when interpreting TARGET and the 29 LSB of the selected reference counter as signed integers, otherwise the interrupt triggers immediately.

TKE_TOP_TIMER_TMR7_TMRCR_0

```
Register name       Offset          Description
TMRCR{t}        BT + P*{t} + 0      Timer Configuration Register
TMRSR{t}        BT + P*{t} + 4      Timer Status Register
TMRCSSR{t}      BT + P*{t} + 8      Timer Clock Source Selection Register
TMRATR{t}       BT + P*{t} + 12     Timer Absolute Target Register
```

Timer Configuration Register

Offset: 0x80000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR7_0
Reset: 0x40000000 (0b01x0,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: Enable, when set, enables the countdown process. **0** = DISABLE **1** = ENABLE
30	0x1	**PER**: Periodic, when set, the internal counter is automatically reloaded after reaching 0 **0** = DISABLE **1** = ENABLE
28:0	0x0	**PTV**: Timer Present Trigger Value. When TMRCR is written and EN is 1b, the internal counter is loaded with PTV+1 and starts counting down at each timing reference pulse. An interrupt is set when the internal counter reaches 0. Note that writing TMRCR with EN=1 while EN was already 1 restarts the counter. Once the timer is started, TARGET can be used to either - observe the value of the reference counter when the next interrupt will be generated (read) - select the exact reference counter value at which the next interrupt will be generated (write) **536870911** = MAX

TKE_TOP_TIMER_TMR7_TMRSR_0

Timer Status Register

Offset: 0x80004
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR7_0
Reset: 0x00000000 (0bx0x0,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Parity Protection	Reset	Description
30	RW	Y	0x0	**INTR_CLR**: Clears the interrupt when written as 1b, read always returns 0 **0** = DISABLE **1** = ENABLE
28:0	RO	N	0x0	**PCV**: Current counter value, writing to this field has no effect

TKE_TOP_TIMER_TMR7_TMRCSSR_0

Timer Clock Source Selection Register

Offset: 0x80008
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR7_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**SRC_ID**: Select Count reference 2'b11 - TSC [41:12] 2'b10 - TSC [29:0] 2'b01 - oscCnt [29:0] 2'b00 - usecCnt [29:0] The implementation of each timer uses comparison against counters incrementing at the selected rate. **0** = SRC_USECCNT **1** = SRC_OSCCNT **2** = SRC_TSCCNT_29_0 **3** = SRC_TSCCNT_41_12

TKE_TOP_TIMER_TMR7_TMRATR_0

Timer Absolute Target Register

Offset: 0x8000c
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: TKE_TOP_SCR_TMRSCR7_0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:0	0x0	**TARGET**: Target value of the selected reference counter, i.e., when value of the selected reference equals target value, the interrupt is generated. The difference between TARGET and the current value must be positive when interpreting TARGET and the 29 LSB of the selected reference counter as signed integers, otherwise the interrupt triggers immediately.

TKE_TOP_TIMER_TMR8_TMRCR_0

```
Register name       Offset          Description
TMRCR{t}        BT + P*{t} + 0      Timer Configuration Register
TMRSR{t}        BT + P*{t} + 4      Timer Status Register
TMRCSSR{t}      BT + P*{t} + 8      Timer Clock Source Selection Register
TMRATR{t}       BT + P*{t} + 12     Timer Absolute Target Register
```

Timer Configuration Register

Offset: 0x90000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR8_0
Reset: 0x40000000 (0b01x0,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: Enable, when set, enables the countdown process. **0** = DISABLE **1** = ENABLE
30	0x1	**PER**: Periodic, when set, the internal counter is automatically reloaded after reaching 0 **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
28:0	0x0	**PTV**: Timer Present Trigger Value. When TMRCR is written and EN is 1b, the internal counter is loaded with PTV+1 and starts counting down at each timing reference pulse. An interrupt is set when the internal counter reaches 0. Note that writing TMRCR with EN=1 while EN was already 1 restarts the counter. Once the timer is started, TARGET can be used to either - observe the value of the reference counter when the next interrupt will be generated (read) - select the exact reference counter value at which the next interrupt will be generated (write) **536870911** = MAX

TKE_TOP_TIMER_TMR8_TMRSR_0

Timer Status Register

Offset: 0x90004
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR8_0
Reset: 0x00000000 (0bx0x0,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Parity Protection	Reset	Description
30	RW	Y	0x0	**INTR_CLR**: Clears the interrupt when written as 1b, read always returns 0 **0** = DISABLE **1** = ENABLE
28:0	RO	N	0x0	**PCV**: Current counter value, writing to this field has no effect

TKE_TOP_TIMER_TMR8_TMRCSSR_0

Timer Clock Source Selection Register

Offset: 0x90008
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: TKE_TOP_SCR_TMRSCR8_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**SRC_ID**: Select Count reference 2'b11 - TSC [41:12] 2'b10 - TSC [29:0] 2'b01 - oscCnt [29:0] 2'b00 - usecCnt [29:0] The implementation of each timer uses comparison against counters incrementing at the selected rate. **0** = SRC_USECCNT **1** = SRC_OSCCNT **2** = SRC_TSCCNT_29_0 **3** = SRC_TSCCNT_41_12

TKE_TOP_TIMER_TMR8_TMRATR_0

Timer Absolute Target Register

Offset: 0x9000c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR8_0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:0	0x0	**TARGET**: Target value of the selected reference counter, i.e., when value of the selected reference equals target value, the interrupt is generated. The difference between TARGET and the current value must be positive when interpreting TARGET and the 29 LSB of the selected reference counter as signed integers, otherwise the interrupt triggers immediately.

TKE_TOP_TIMER_TMR9_TMRCR_0

```
Register name       Offset          Description
TMRCR{t}        BT + P*{t} + 0      Timer Configuration Register
TMRSR{t}        BT + P*{t} + 4      Timer Status Register
TMRCSSR{t}      BT + P*{t} + 8      Timer Clock Source Selection Register
TMRATR{t}       BT + P*{t} + 12     Timer Absolute Target Register
```

Timer Configuration Register

Offset: 0xa0000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR9_0
Reset: 0x40000000 (0b01x0,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: Enable, when set, enables the countdown process. **0** = DISABLE **1** = ENABLE
30	0x1	**PER**: Periodic, when set, the internal counter is automatically reloaded after reaching 0 **0** = DISABLE **1** = ENABLE
28:0	0x0	**PTV**: Timer Present Trigger Value. When TMRCR is written and EN is 1b, the internal counter is loaded with PTV+1 and starts counting down at each timing reference pulse. An interrupt is set when the internal counter reaches 0. Note that writing TMRCR with EN=1 while EN was already 1 restarts the counter. Once the timer is started, TARGET can be used to either - observe the value of the reference counter when the next interrupt will be generated (read) - select the exact reference counter value at which the next interrupt will be generated (write) **536870911** = MAX

TKE_TOP_TIMER_TMR9_TMRSR_0

Timer Status Register

Offset: 0xa0004
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR9_0
Reset: 0x00000000 (0bx0x0,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Parity Protection	Reset	Description
30	RW	Y	0x0	**INTR_CLR**: Clears the interrupt when written as 1b, read always returns 0 **0** = DISABLE **1** = ENABLE
28:0	RO	N	0x0	**PCV**: Current counter value, writing to this field has no effect

TKE_TOP_TIMER_TMR9_TMRCSSR_0

Timer Clock Source Selection Register

Offset: 0xa0008
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR9_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**SRC_ID**: Select Count reference 2'b11 - TSC [41:12] 2'b10 - TSC [29:0] 2'b01 - oscCnt [29:0] 2'b00 - usecCnt [29:0] The implementation of each timer uses comparison against counters incrementing at the selected rate. **0** = SRC_USECCNT **1** = SRC_OSCCNT **2** = SRC_TSCCNT_29_0 **3** = SRC_TSCCNT_41_12

TKE_TOP_TIMER_TMR9_TMRATR_0

Timer Absolute Target Register

Offset: 0xa000c
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: TKE_TOP_SCR_TMRSCR9_0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:0	0x0	**TARGET**: Target value of the selected reference counter, i.e., when value of the selected reference equals target value, the interrupt is generated. The difference between TARGET and the current value must be positive when interpreting TARGET and the 29 LSB of the selected reference counter as signed integers, otherwise the interrupt triggers immediately.

TKE_TOP_TIMER_TMR10_TMRCR_0

```
Register name       Offset          Description
TMRCR{t}        BT + P*{t} + 0      Timer Configuration Register
TMRSR{t}        BT + P*{t} + 4      Timer Status Register
TMRCSSR{t}      BT + P*{t} + 8      Timer Clock Source Selection Register
TMRATR{t}       BT + P*{t} + 12     Timer Absolute Target Register
```

Timer Configuration Register

Offset: 0xb0000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR10_0
Reset: 0x40000000 (0b01x0,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: Enable, when set, enables the countdown process. **0** = DISABLE **1** = ENABLE
30	0x1	**PER**: Periodic, when set, the internal counter is automatically reloaded after reaching 0 **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
28:0	0x0	**PTV**: Timer Present Trigger Value. When TMRCR is written and EN is 1b, the internal counter is loaded with PTV+1 and starts counting down at each timing reference pulse. An interrupt is set when the internal counter reaches 0. Note that writing TMRCR with EN=1 while EN was already 1 restarts the counter. Once the timer is started, TARGET can be used to either - observe the value of the reference counter when the next interrupt will be generated (read) - select the exact reference counter value at which the next interrupt will be generated (write) **536870911** = MAX

TKE_TOP_TIMER_TMR10_TMRSR_0

Timer Status Register

Offset: 0xb0004
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR10_0
Reset: 0x00000000 (0bx0x0,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Parity Protection	Reset	Description
30	RW	Y	0x0	**INTR_CLR**: Clears the interrupt when written as 1b, read always returns 0 **0** = DISABLE **1** = ENABLE
28:0	RO	N	0x0	**PCV**: Current counter value, writing to this field has no effect

TKE_TOP_TIMER_TMR10_TMRCSSR_0

Timer Clock Source Selection Register

Offset: 0xb0008
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: TKE_TOP_SCR_TMRSCR10_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**SRC_ID**: Select Count reference 2'b11 - TSC [41:12] 2'b10 - TSC [29:0] 2'b01 - oscCnt [29:0] 2'b00 - usecCnt [29:0] The implementation of each timer uses comparison against counters incrementing at the selected rate. **0** = SRC_USECCNT **1** = SRC_OSCCNT **2** = SRC_TSCCNT_29_0 **3** = SRC_TSCCNT_41_12

TKE_TOP_TIMER_TMR10_TMRATR_0

Timer Absolute Target Register

Offset: 0xb000c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR10_0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:0	0x0	**TARGET**: Target value of the selected reference counter, i.e., when value of the selected reference equals target value, the interrupt is generated. The difference between TARGET and the current value must be positive when interpreting TARGET and the 29 LSB of the selected reference counter as signed integers, otherwise the interrupt triggers immediately.

TKE_TOP_TIMER_TMR11_TMRCR_0

```
Register name       Offset          Description
TMRCR{t}        BT + P*{t} + 0      Timer Configuration Register
TMRSR{t}        BT + P*{t} + 4      Timer Status Register
TMRCSSR{t}      BT + P*{t} + 8      Timer Clock Source Selection Register
TMRATR{t}       BT + P*{t} + 12     Timer Absolute Target Register
```

Timer Configuration Register

Offset: 0xc0000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR11_0
Reset: 0x40000000 (0b01x0,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: Enable, when set, enables the countdown process. **0** = DISABLE **1** = ENABLE
30	0x1	**PER**: Periodic, when set, the internal counter is automatically reloaded after reaching 0 **0** = DISABLE **1** = ENABLE
28:0	0x0	**PTV**: Timer Present Trigger Value. When TMRCR is written and EN is 1b, the internal counter is loaded with PTV+1 and starts counting down at each timing reference pulse. An interrupt is set when the internal counter reaches 0. Note that writing TMRCR with EN=1 while EN was already 1 restarts the counter. Once the timer is started, TARGET can be used to either - observe the value of the reference counter when the next interrupt will be generated (read) - select the exact reference counter value at which the next interrupt will be generated (write) **536870911** = MAX

TKE_TOP_TIMER_TMR11_TMRSR_0

Timer Status Register

Offset: 0xc0004
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR11_0
Reset: 0x00000000 (0bx0x0,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Parity Protection	Reset	Description
30	RW	Y	0x0	**INTR_CLR**: Clears the interrupt when written as 1b, read always returns 0 **0** = DISABLE **1** = ENABLE
28:0	RO	N	0x0	**PCV**: Current counter value, writing to this field has no effect

TKE_TOP_TIMER_TMR11_TMRCSSR_0

Timer Clock Source Selection Register

Offset: 0xc0008
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR11_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**SRC_ID**: Select Count reference 2'b11 - TSC [41:12] 2'b10 - TSC [29:0] 2'b01 - oscCnt [29:0] 2'b00 - usecCnt [29:0] The implementation of each timer uses comparison against counters incrementing at the selected rate. **0** = SRC_USECCNT **1** = SRC_OSCCNT **2** = SRC_TSCCNT_29_0 **3** = SRC_TSCCNT_41_12

TKE_TOP_TIMER_TMR11_TMRATR_0

Timer Absolute Target Register

Offset: 0xc000c
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: TKE_TOP_SCR_TMRSCR11_0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:0	0x0	**TARGET**: Target value of the selected reference counter, i.e., when value of the selected reference equals target value, the interrupt is generated. The difference between TARGET and the current value must be positive when interpreting TARGET and the 29 LSB of the selected reference counter as signed integers, otherwise the interrupt triggers immediately.

TKE_TOP_TIMER_TMR12_TMRCR_0

```
Register name       Offset          Description
TMRCR{t}        BT + P*{t} + 0      Timer Configuration Register
TMRSR{t}        BT + P*{t} + 4      Timer Status Register
TMRCSSR{t}      BT + P*{t} + 8      Timer Clock Source Selection Register
TMRATR{t}       BT + P*{t} + 12     Timer Absolute Target Register
```

Timer Configuration Register

Offset: 0xd0000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR12_0
Reset: 0x40000000 (0b01x0,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: Enable, when set, enables the countdown process. **0** = DISABLE **1** = ENABLE
30	0x1	**PER**: Periodic, when set, the internal counter is automatically reloaded after reaching 0 **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
28:0	0x0	**PTV**: Timer Present Trigger Value. When TMRCR is written and EN is 1b, the internal counter is loaded with PTV+1 and starts counting down at each timing reference pulse. An interrupt is set when the internal counter reaches 0. Note that writing TMRCR with EN=1 while EN was already 1 restarts the counter. Once the timer is started, TARGET can be used to either - observe the value of the reference counter when the next interrupt will be generated (read) - select the exact reference counter value at which the next interrupt will be generated (write) **536870911** = MAX

TKE_TOP_TIMER_TMR12_TMRSR_0

Timer Status Register

Offset: 0xd0004
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR12_0
Reset: 0x00000000 (0bx0x0,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Parity Protection	Reset	Description
30	RW	Y	0x0	**INTR_CLR**: Clears the interrupt when written as 1b, read always returns 0 **0** = DISABLE **1** = ENABLE
28:0	RO	N	0x0	**PCV**: Current counter value, writing to this field has no effect

TKE_TOP_TIMER_TMR12_TMRCSSR_0

Timer Clock Source Selection Register

Offset: 0xd0008
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: TKE_TOP_SCR_TMRSCR12_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**SRC_ID**: Select Count reference 2'b11 - TSC [41:12] 2'b10 - TSC [29:0] 2'b01 - oscCnt [29:0] 2'b00 - usecCnt [29:0] The implementation of each timer uses comparison against counters incrementing at the selected rate. **0** = SRC_USECCNT **1** = SRC_OSCCNT **2** = SRC_TSCCNT_29_0 **3** = SRC_TSCCNT_41_12

TKE_TOP_TIMER_TMR12_TMRATR_0

Timer Absolute Target Register

Offset: 0xd000c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR12_0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:0	0x0	**TARGET**: Target value of the selected reference counter, i.e., when value of the selected reference equals target value, the interrupt is generated. The difference between TARGET and the current value must be positive when interpreting TARGET and the 29 LSB of the selected reference counter as signed integers, otherwise the interrupt triggers immediately.

TKE_TOP_TIMER_TMR13_TMRCR_0

```
Register name       Offset          Description
TMRCR{t}        BT + P*{t} + 0      Timer Configuration Register
TMRSR{t}        BT + P*{t} + 4      Timer Status Register
TMRCSSR{t}      BT + P*{t} + 8      Timer Clock Source Selection Register
TMRATR{t}       BT + P*{t} + 12     Timer Absolute Target Register
```

Timer Configuration Register

Offset: 0xe0000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR13_0
Reset: 0x40000000 (0b01x0,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: Enable, when set, enables the countdown process. **0** = DISABLE **1** = ENABLE
30	0x1	**PER**: Periodic, when set, the internal counter is automatically reloaded after reaching 0 **0** = DISABLE **1** = ENABLE
28:0	0x0	**PTV**: Timer Present Trigger Value. When TMRCR is written and EN is 1b, the internal counter is loaded with PTV+1 and starts counting down at each timing reference pulse. An interrupt is set when the internal counter reaches 0. Note that writing TMRCR with EN=1 while EN was already 1 restarts the counter. Once the timer is started, TARGET can be used to either - observe the value of the reference counter when the next interrupt will be generated (read) - select the exact reference counter value at which the next interrupt will be generated (write) **536870911** = MAX

TKE_TOP_TIMER_TMR13_TMRSR_0

Timer Status Register

Offset: 0xe0004
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR13_0
Reset: 0x00000000 (0bx0x0,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Parity Protection	Reset	Description
30	RW	Y	0x0	**INTR_CLR**: Clears the interrupt when written as 1b, read always returns 0 **0** = DISABLE **1** = ENABLE
28:0	RO	N	0x0	**PCV**: Current counter value, writing to this field has no effect

TKE_TOP_TIMER_TMR13_TMRCSSR_0

Timer Clock Source Selection Register

Offset: 0xe0008
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR13_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**SRC_ID**: Select Count reference 2'b11 - TSC [41:12] 2'b10 - TSC [29:0] 2'b01 - oscCnt [29:0] 2'b00 - usecCnt [29:0] The implementation of each timer uses comparison against counters incrementing at the selected rate. **0** = SRC_USECCNT **1** = SRC_OSCCNT **2** = SRC_TSCCNT_29_0 **3** = SRC_TSCCNT_41_12

TKE_TOP_TIMER_TMR13_TMRATR_0

Timer Absolute Target Register

Offset: 0xe000c
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: TKE_TOP_SCR_TMRSCR13_0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:0	0x0	**TARGET**: Target value of the selected reference counter, i.e., when value of the selected reference equals target value, the interrupt is generated. The difference between TARGET and the current value must be positive when interpreting TARGET and the 29 LSB of the selected reference counter as signed integers, otherwise the interrupt triggers immediately.

TKE_TOP_TIMER_TMR14_TMRCR_0

```
Register name       Offset          Description
TMRCR{t}        BT + P*{t} + 0      Timer Configuration Register
TMRSR{t}        BT + P*{t} + 4      Timer Status Register
TMRCSSR{t}      BT + P*{t} + 8      Timer Clock Source Selection Register
TMRATR{t}       BT + P*{t} + 12     Timer Absolute Target Register
```

Timer Configuration Register

Offset: 0xf0000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR14_0
Reset: 0x40000000 (0b01x0,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: Enable, when set, enables the countdown process. **0** = DISABLE **1** = ENABLE
30	0x1	**PER**: Periodic, when set, the internal counter is automatically reloaded after reaching 0 **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
28:0	0x0	**PTV**: Timer Present Trigger Value. When TMRCR is written and EN is 1b, the internal counter is loaded with PTV+1 and starts counting down at each timing reference pulse. An interrupt is set when the internal counter reaches 0. Note that writing TMRCR with EN=1 while EN was already 1 restarts the counter. Once the timer is started, TARGET can be used to either - observe the value of the reference counter when the next interrupt will be generated (read) - select the exact reference counter value at which the next interrupt will be generated (write) **536870911** = MAX

TKE_TOP_TIMER_TMR14_TMRSR_0

Timer Status Register

Offset: 0xf0004
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR14_0
Reset: 0x00000000 (0bx0x0,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Parity Protection	Reset	Description
30	RW	Y	0x0	**INTR_CLR**: Clears the interrupt when written as 1b, read always returns 0 **0** = DISABLE **1** = ENABLE
28:0	RO	N	0x0	**PCV**: Current counter value, writing to this field has no effect

TKE_TOP_TIMER_TMR14_TMRCSSR_0

Timer Clock Source Selection Register

Offset: 0xf0008
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: TKE_TOP_SCR_TMRSCR14_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**SRC_ID**: Select Count reference 2'b11 - TSC [41:12] 2'b10 - TSC [29:0] 2'b01 - oscCnt [29:0] 2'b00 - usecCnt [29:0] The implementation of each timer uses comparison against counters incrementing at the selected rate. **0** = SRC_USECCNT **1** = SRC_OSCCNT **2** = SRC_TSCCNT_29_0 **3** = SRC_TSCCNT_41_12

TKE_TOP_TIMER_TMR14_TMRATR_0

Timer Absolute Target Register

Offset: 0xf000c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR14_0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:0	0x0	**TARGET**: Target value of the selected reference counter, i.e., when value of the selected reference equals target value, the interrupt is generated. The difference between TARGET and the current value must be positive when interpreting TARGET and the 29 LSB of the selected reference counter as signed integers, otherwise the interrupt triggers immediately.

TKE_TOP_TIMER_TMR15_TMRCR_0

```
Register name       Offset          Description
TMRCR{t}        BT + P*{t} + 0      Timer Configuration Register
TMRSR{t}        BT + P*{t} + 4      Timer Status Register
TMRCSSR{t}      BT + P*{t} + 8      Timer Clock Source Selection Register
TMRATR{t}       BT + P*{t} + 12     Timer Absolute Target Register
```

Timer Configuration Register

Offset: 0x100000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR15_0
Reset: 0x40000000 (0b01x0,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: Enable, when set, enables the countdown process. **0** = DISABLE **1** = ENABLE
30	0x1	**PER**: Periodic, when set, the internal counter is automatically reloaded after reaching 0 **0** = DISABLE **1** = ENABLE
28:0	0x0	**PTV**: Timer Present Trigger Value. When TMRCR is written and EN is 1b, the internal counter is loaded with PTV+1 and starts counting down at each timing reference pulse. An interrupt is set when the internal counter reaches 0. Note that writing TMRCR with EN=1 while EN was already 1 restarts the counter. Once the timer is started, TARGET can be used to either - observe the value of the reference counter when the next interrupt will be generated (read) - select the exact reference counter value at which the next interrupt will be generated (write) **536870911** = MAX

TKE_TOP_TIMER_TMR15_TMRSR_0

Timer Status Register

Offset: 0x100004
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR15_0
Reset: 0x00000000 (0bx0x0,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Parity Protection	Reset	Description
30	RW	Y	0x0	**INTR_CLR**: Clears the interrupt when written as 1b, read always returns 0 **0** = DISABLE **1** = ENABLE
28:0	RO	N	0x0	**PCV**: Current counter value, writing to this field has no effect

TKE_TOP_TIMER_TMR15_TMRCSSR_0

Timer Clock Source Selection Register

Offset: 0x100008
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_TMRSCR15_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**SRC_ID**: Select Count reference 2'b11 - TSC [41:12] 2'b10 - TSC [29:0] 2'b01 - oscCnt [29:0] 2'b00 - usecCnt [29:0] The implementation of each timer uses comparison against counters incrementing at the selected rate. **0** = SRC_USECCNT **1** = SRC_OSCCNT **2** = SRC_TSCCNT_29_0 **3** = SRC_TSCCNT_41_12

TKE_TOP_TIMER_TMR15_TMRATR_0

Timer Absolute Target Register

Offset: 0x10000c
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: TKE_TOP_SCR_TMRSCR15_0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:0	0x0	**TARGET**: Target value of the selected reference counter, i.e., when value of the selected reference equals target value, the interrupt is generated. The difference between TARGET and the current value must be positive when interpreting TARGET and the 29 LSB of the selected reference counter as signed integers, otherwise the interrupt triggers immediately.

TKE_TOP_WDT0_WDTCR_0

```
WDTCR{w} BW + P*{w} + 0    RW    Watchdog Timer Configuration Register
WDTSR{w} BW + P*{w} + 4    RO    Watchdog Timer Status Register
WDTCMDR{w} BW + P*{w} + 8  WO    Watchdog Timer Command Register
WDTUR{w} BW + P*{w} + 12   RW    Watchdog Timer Unlock Register
```

Watchdog Timer Configuration Register

Offset: 0x110000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_WDTSCR0_0
Reset: 0x00700000 (0b0000,xx00,0111,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**WindowedRestartDisableMap**: When WindowedOperationEnable is 1b, restart commands attempted when expiration level has value b are rejected if bit b equals 1b in this field
25	0x0	**DisallowWDTFreeze**: When 1b, the signal watchdog_freeze_request has no effect When 0b, when the signal watchdog_freeze_request is asserted, the WDT stops decrementing CurrentCount if the WDT is enabled

Bit	Reset	Description
24:23	0x0	**RestartErrorAction**: If a restart action is attempted but fails, indicate which of four actions to take 0x0: do nothing, silently ignore the restart 0x1: immediately jump to expiration level 4 and reload the downcounter 0x2: immediately jump to expiration level 5 and reload the downcounter 0x3: immediately request a system power on reset (POR) When jumping levels, the normal actions configured per level must take place **0** = NOP **1** = JUMP_TO_4 **2** = JUMP_TO_5 **3** = POR_RESET
22:20	0x7	**ErrorThreshold**: The WDT logic asserts an error signal to HSM when ExpirationLevel >= ErrorThreshold.
19	0x0	**TscReferenceEnable**: Select as timing reference transitions on a configured TSC bit.
18	0x0	**ChallengeResponseEnable**: Enable the Challenge Response mode of operation. When 1b, the Unlock register operates as a Challenge Response register.
17	0x0	**WindowedOperationEnable**: Enable the windowed mode of operation. When 1b, a restart command is rejected when the current expiration level is identified as disallowed in the WindowedRestartDisableMap or the current time is below the window threshold. The rejection is either silent or performs a reset based on RestartErrorAction
16	0x0	**SystemPOResetEnable**: Enable internal full system reset at fifth expiration of the counter. This reset cannot be intercepted and acts as a Power On Reset **0** = DISABLE **1** = ENABLE
15	0x0	**SystemDebugResetEnable**: Enable systemwide reset assertion at fourth expiration of the counter. This reset can be intercepted by debug logic, e.g., to preserve information
14	0x0	**RemoteInterruptEnable**: Enable normal priority interrupt assertion at third expiration of the counter, connected to LIC for systemwide visibility
13	0x0	**LocalFIQEnable**: Enable high priority interrupt (FIQ) assertion at second expiration of the counter, connected to local interrupt controller
12	0x0	**LocalInterruptEnable**: Enable normal priority interrupt assertion at first expiration of the counter, connected to local interrupt controller

Bit	Reset	Description
11:4	0x0	**Period**: Measured in periods of the timer selected as source, this is the reload value, so 0 is treated as maximum period
3:0	0x0	**TimerSource**: The timer source interpretation is dependent on the value of TscReferenceEnable: TscReferenceEnable = 0b: timer<TimerSource> is the timing reference TscReferenceEnable = 1b: a transition of TSC[2*TimerSource] is the timing reference **0** = TMR0

TKE_TOP_WDT0_WDTSR_0

Watchdog Timer Status Register

Offset: 0x110004
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TKE_TOP_SCR_WDTSCR0_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,x000,0000,0000,0000)

Bit	Reset	Description
19	0x0	**Frozen**: 1b indicates that the WDT is frozen because of the assertion of the global WDT freeze request (when allowed) or because of the assertion of the global timer freeze request
18	0x0	**ErrorForced**: 1b indicates that the error reported to HSM is forced active for testing purposes
17	0x0	**StickyEnabled**: 1b when the counter is enabled and cannot be disabled, set by StickyStart command, cleared by reset only.
16	0x0	**CurrentError**: Current error reported to HSM, this is the output after combining the functional and forced error paths
14:12	0x0	**CurrentExpirationCount**: Current count of expiration since last start operation, cleared by Start command, saturating at 5. **0** = NO_EXPIRY **1** = LOCAL_IRQ **2** = LOCAL_FIQ **3** = REMOTE_IRQ **4** = DEBUG_RESET **5** = SYSTEM_RESET

Bit	Reset	Description
11:4	0x0	**CurrentCount**: Current value of the downcounter, reloaded with Period by Start command.
3	0x0	**RemoteInterruptStatus**: Current status of remote interrupt, cleared by Start command.
2	0x0	**LocalFIQStatus**: Current status of FIQ, cleared by Start command.
1	0x0	**LocalInterruptStatus**: Current status of local interrupt, cleared by Start command.
0	0x0	**Enabled**: 1b when the counter is active. When Enabled is 1b, the configuration of the WDT is frozen by silently ignoring write attempts to the configuration registers: WDTCR, WDTSCR and WDTFWCR. Set by Start or StickyStart commands, cleared by Disable when StickyEnabled is 0b. **0** = DISABLE **1** = ENABLE

TKE_TOP_WDT0_WDTCMDR_0

Watchdog Timer Command Register

Offset: 0x110008
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_WDTSCR0_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	0x0	**ForceError**: Only working if the Command register is unlocked. 1b sets an internal flop that is ORed with the functional error signal (based on threshold), 0b clears the bit. The state of the flop is shown in the status register.
2	0x0	**StickyStart**: Qualifies the StartCounter bit, if both bits are 1b, the counter is enabled in permanent fashion, i.e., it remains enabled until the next reset, and a disable counter operation has no effect. Ignored if StartCounter is 0b

Bit	Reset	Description
1	0x0	**DisableCounter**: Only working if the unlock register (WDTUR{w}) has been programmed before with the correct pattern. Writing to the command register always clears the unlock register. If written to 1b, while StartCounter is 0b and the unlock register contains the unlock pattern, the watchdog transitions back to disabled. **0** = DISABLE **1** = ENABLE
0	0x0	**StartCounter**: When written to 1b, enables the counter operation, loads the counter with Period and starts downcounting, resets the expiration count to 0 and clears all status flags. Enabled is always 1b after this. This is also used as Restart. Writing to StartCounter requires unlocking when ChallengeResponseEnable is 1b, and writing to StartCounter is also restricted to windows of time given by WindowedRestartDisableMap when WindowedOperationEnable is 1b. **0** = DISABLE **1** = ENABLE

TKE_TOP_WDT0_WDTUR_0

Watchdog Timer Unlock Register

Offset: 0x11000c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_WDTSCR0_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UnlockPattern**: The behavior of the Unlock pattern is dependent on the ChallengeResponseEnable configuration bit. When ChallengeResponseEnable is 0b, UnlockPattern must be written with value UNLOCK to allow a write to DisableCounter to take effect. UnlockPattern is reset at each write to the Command register. When ChallengeResponseEnable is 1b, a write to the Command register is only allowed after reading UnlockPattern to get the current value of an LFSR, followed by writing the next value of the LFSR to the UnlockPattern. If the two values match, the unlock is successful. In all cases the LFSR gets (re)initialized with the written value. **50266** = UNLOCK

TKE_TOP_WDT0_WDTSCR_0

Watchdog Timer Skip Configuration Register

Offset: 0x110010
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_WDTSCR0_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,x000,x000,x000)

Bit	Reset	Description
14:12	0x0	**Skip3**: Skip Value at Expiration count 3 At a Downcounter expiration, the expiration level changes as ExpC <= ExpC + 1 + Skip(ExpC), saturating at 5
10:8	0x0	**Skip2**: Skip Value at Expiration count 2
6:4	0x0	**Skip1**: Skip Value at Expiration count 1
2:0	0x0	**Skip0**: Skip Value at Expiration count 0

TKE_TOP_WDT0_WDTFWCR_0

Watchdog Timer Fine Window Configuration Register

Offset: 0x110014
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_WDTSCR0_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,x000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
18:16	0x0	**StartLevel**: If restart (i.e., start while WDT is already enabled) is attempted and windowing operation is enabled, restart is rejected if current expiration < StartLevel

Bit	Reset	Description
7:0	0x0	**StartCount**: If restart (i.e., start while WDT is already enabled) is attempted and windowing operation is enabled and current expiration level == StartLevel, restart is rejected if current downcounter > (Period - StartCount). StartCount is the number of cycles where rejection takes place at the start of that level.

TKE_TOP_WDT1_WDTCR_0

```
WDTCR{w} BW + P*{w} + 0    RW    Watchdog Timer Configuration Register
WDTSR{w} BW + P*{w} + 4    RO    Watchdog Timer Status Register
WDTCMDR{w} BW + P*{w} + 8  WO    Watchdog Timer Command Register
WDTUR{w} BW + P*{w} + 12   RW    Watchdog Timer Unlock Register
```

Watchdog Timer Configuration Register

Offset: 0x120000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_WDTSCR1_0
Reset: 0x00700000 (0b0000,xx00,0111,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**WindowedRestartDisableMap**: When WindowedOperationEnable is 1b, restart commands attempted when expiration level has value b are rejected if bit b equals 1b in this field
25	0x0	**DisallowWDTFreeze**: When 1b, the signal watchdog_freeze_request has no effect When 0b, when the signal watchdog_freeze_request is asserted, the WDT stops decrementing CurrentCount if the WDT is enabled
24:23	0x0	**RestartErrorAction**: If a restart action is attempted but fails, indicate which of four actions to take 0x0: do nothing, silently ignore the restart 0x1: immediately jump to expiration level 4 and reload the downcounter 0x2: immediately jump to expiration level 5 and reload the downcounter 0x3: immediately request a system POR When jumping levels, the normal actions configured per level must take place **0** = NOP **1** = JUMP_TO_4 **2** = JUMP_TO_5 **3** = POR_RESET

Bit	Reset	Description
22:20	0x7	**ErrorThreshold**: The WDT logic asserts an error signal to HSM when ExpirationLevel >= ErrorThreshold.
19	0x0	**TscReferenceEnable**: Select as timing reference transitions on a configured TSC bit.
18	0x0	**ChallengeResponseEnable**: Enable the Challenge Response mode of operation. When 1b, the Unlock register operates as a Challenge Response register.
17	0x0	**WindowedOperationEnable**: Enable the windowed mode of operation. When 1b, a restart command is rejected when the current expiration level is identified as disallowed in the WindowedRestartDisableMap or the current time is below the window threshold. The rejection is either silent or performs a reset based on RestartErrorAction
16	0x0	**SystemPOResetEnable**: Enable internal full system reset at fifth expiration of the counter. This reset cannot be intercepted and acts as a Power On Reset **0** = DISABLE **1** = ENABLE
15	0x0	**SystemDebugResetEnable**: Enable systemwide reset assertion at fourth expiration of the counter. This reset can be intercepted by debug logic, e.g., to preserve information
14	0x0	**RemoteInterruptEnable**: Enable normal priority interrupt assertion at third expiration of the counter, connected to LIC for systemwide visibility
13	0x0	**LocalFIQEnable**: Enable high priority interrupt (FIQ) assertion at second expiration of the counter, connected to local interrupt controller
12	0x0	**LocalInterruptEnable**: Enable normal priority interrupt assertion at first expiration of the counter, connected to local interrupt controller
11:4	0x0	**Period**: Measured in periods of the timer selected as source, this is the reload value, so 0 is treated as maximum period
3:0	0x0	**TimerSource**: The timer source interpretation is dependent on the value of TscReferenceEnable: TscReferenceEnable = 0b: timer<TimerSource> is the timing reference TscReferenceEnable = 1b: a transition of TSC[2*TimerSource] is the timing reference **0** = TMR0

TKE_TOP_WDT1_WDTSR_0

Watchdog Timer Status Register

Offset: 0x120004
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TKE_TOP_SCR_WDTSCR1_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,x000,0000,0000,0000)

Bit	Reset	Description
19	0x0	**Frozen**: 1b indicates that the WDT is frozen because of the assertion of the global WDT freeze request (when allowed) or because of the assertion of the global timer freeze request
18	0x0	**ErrorForced**: 1b indicates that the error reported to HSM is forced active for testing purposes
17	0x0	**StickyEnabled**: 1b when the counter is enabled and cannot be disabled, set by StickyStart command, cleared by reset only.
16	0x0	**CurrentError**: Current error reported to HSM, this is the output after combining the functional and forced error paths
14:12	0x0	**CurrentExpirationCount**: Current count of expiration since last start operation, cleared by Start command, saturating at 5. **0** = NO_EXPIRY **1** = LOCAL_IRQ **2** = LOCAL_FIQ **3** = REMOTE_IRQ **4** = DEBUG_RESET **5** = SYSTEM_RESET
11:4	0x0	**CurrentCount**: Current value of the downcounter, reloaded with Period by Start command.
3	0x0	**RemoteInterruptStatus**: Current status of remote interrupt, cleared by Start command.
2	0x0	**LocalFIQStatus**: Current status of FIQ, cleared by Start command.
1	0x0	**LocalInterruptStatus**: Current status of local interrupt, cleared by Start command.

Bit	Reset	Description
0	0x0	**Enabled**: 1b when the counter is active. When Enabled is 1b, the configuration of the WDT is frozen by silently ignoring write attempts to the configuration registers: WDTCR, WDTSCR and WDTFWCR. Set by Start or StickyStart commands, cleared by Disable when StickyEnabled is 0b. **0** = DISABLE **1** = ENABLE

TKE_TOP_WDT1_WDTCMDR_0

Watchdog Timer Command Register

Offset: 0x120008
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_WDTSCR1_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	0x0	**ForceError**: Only working if the Command register is unlocked. 1b sets an internal flop that is ORed with the functional error signal (based on threshold), 0b clears the bit. The state of the flop is shown in the status register.
2	0x0	**StickyStart**: Qualifies the StartCounter bit, if both bits are 1b, the counter is enabled in permanent fashion, i.e., it remains enabled until the next reset, and a disable counter operation has no effect. Ignored if StartCounter is 0b
1	0x0	**DisableCounter**: Only working if the unlock register (WDTUR{w}) has been programmed before with the correct pattern. Writing to the command register always clears the unlock register. If written to 1b, while StartCounter is 0b and the unlock register contains the unlock pattern, the watchdog transitions back to disabled. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
0	0x0	**StartCounter**: When written to 1b, enables the counter operation, loads the counter with Period and starts downcounting, resets the expiration count to 0 and clears all status flags. Enabled is always 1b after this. This is also used as Restart. Writing to StartCounter requires unlocking when ChallengeResponseEnable is 1b, and writing to StartCounter is also restricted to windows of time given by WindowedRestartDisableMap when WindowedOperationEnable is 1b. **0** = DISABLE **1** = ENABLE

TKE_TOP_WDT1_WDTUR_0

Watchdog Timer Unlock Register

Offset: 0x12000c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_WDTSCR1_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UnlockPattern**: The behavior of the Unlock pattern is dependent on the ChallengeResponseEnable configuration bit. When ChallengeResponseEnable is 0b, UnlockPattern must be written with value UNLOCK to allow a write to DisableCounter to take effect. UnlockPattern is reset at each write to the Command register. When ChallengeResponseEnable is 1b, a write to the Command register is only allowed after reading UnlockPattern to get the current value of an LFSR, followed by writing the next value of the LFSR to the UnlockPattern. If the two values match, the unlock is successful. In all cases the LFSR gets (re)initialized with the written value. **50266** = UNLOCK

TKE_TOP_WDT1_WDTSCR_0

Watchdog Timer Skip Configuration Register

Offset: 0x120010
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: TKE_TOP_SCR_WDTSCR1_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,x000,x000,x000)

Bit	Reset	Description
14:12	0x0	**Skip3**: Skip Value at Expiration count 3 At a Downcounter expiration, the expiration level changes as ExpC <= ExpC + 1 + Skip(ExpC), saturating at 5
10:8	0x0	**Skip2**: Skip Value at Expiration count 2
6:4	0x0	**Skip1**: Skip Value at Expiration count 1
2:0	0x0	**Skip0**: Skip Value at Expiration count 0

TKE_TOP_WDT1_WDTFWCR_0

Watchdog Timer Fine Window Configuration Register

Offset: 0x120014
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_WDTSCR1_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,x000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
18:16	0x0	**StartLevel**: If restart (i.e., start while WDT is already enabled) is attempted and windowing operation is enabled, restart is rejected if current expiration < StartLevel
7:0	0x0	**StartCount**: If restart (i.e., start while WDT is already enabled) is attempted and windowing operation is enabled and current expiration level == StartLevel, restart is rejected if current downcounter > (Period - StartCount). StartCount is the number of cycles where rejection takes place at the start of that level.

TKE_TOP_WDT2_WDTCR_0

```
WDTCR{w} BW + P*{w} + 0    RW    Watchdog Timer Configuration Register
WDTSR{w} BW + P*{w} + 4    RO    Watchdog Timer Status Register
```

```
WDTCMDR{w} BW + P*{w} + 8  WO    Watchdog Timer Command Register
WDTUR{w} BW + P*{w} + 12   RW    Watchdog Timer Unlock Register
```

Watchdog Timer Configuration Register

Offset: 0x130000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_WDTSCR2_0
Reset: 0x00700000 (0b0000,xx00,0111,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**WindowedRestartDisableMap**: When WindowedOperationEnable is 1b, restart commands attempted when expiration level has value b are rejected if bit b equals 1b in this field
25	0x0	**DisallowWDTFreeze**: When 1b, the signal watchdog_freeze_request has no effect When 0b, when the signal watchdog_freeze_request is asserted, the WDT stops decrementing CurrentCount if the WDT is enabled
24:23	0x0	**RestartErrorAction**: If a restart action is attempted but fails, indicate which of four actions to take 0x0: do nothing, silently ignore the restart 0x1: immediately jump to expiration level 4 and reload the downcounter 0x2: immediately jump to expiration level 5 and reload the downcounter 0x3: immediately request a system POR When jumping levels, the normal actions configured per level must take place **0** = NOP **1** = JUMP_TO_4 **2** = JUMP_TO_5 **3** = POR_RESET
22:20	0x7	**ErrorThreshold**: The WDT logic asserts an error signal to HSM when ExpirationLevel >= ErrorThreshold.
19	0x0	**TscReferenceEnable**: Select as timing reference transitions on a configured TSC bit.
18	0x0	**ChallengeResponseEnable**: Enable the Challenge Response mode of operation. When 1b, the Unlock register operates as a Challenge Response register.

Bit	Reset	Description
17	0x0	**WindowedOperationEnable**: Enable the windowed mode of operation. When 1b, a restart command is rejected when the current expiration level is identified as disallowed in the WindowedRestartDisableMap or the current time is below the window threshold. The rejection is either silent or performs a reset based on RestartErrorAction
16	0x0	**SystemPOResetEnable**: Enable internal full system reset at fifth expiration of the counter. This reset cannot be intercepted and acts as a Power On Reset **0** = DISABLE **1** = ENABLE
15	0x0	**SystemDebugResetEnable**: Enable systemwide reset assertion at fourth expiration of the counter. This reset can be intercepted by debug logic, e.g., to preserve information
14	0x0	**RemoteInterruptEnable**: Enable normal priority interrupt assertion at third expiration of the counter, connected to LIC for systemwide visibility
13	0x0	**LocalFIQEnable**: Enable high priority interrupt (FIQ) assertion at second expiration of the counter, connected to local interrupt controller
12	0x0	**LocalInterruptEnable**: Enable normal priority interrupt assertion at first expiration of the counter, connected to local interrupt controller
11:4	0x0	**Period**: Measured in periods of the timer selected as source, this is the reload value, so 0 is treated as maximum period
3:0	0x0	**TimerSource**: The timer source interpretation is dependent on the value of TscReferenceEnable: TscReferenceEnable = 0b: timer<TimerSource> is the timing reference TscReferenceEnable = 1b: a transition of TSC[2*TimerSource] is the timing reference **0** = TMR0

TKE_TOP_WDT2_WDTSR_0

Watchdog Timer Status Register

Offset: 0x130004
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: TKE_TOP_SCR_WDTSCR2_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,x000,0000,0000,0000)

Bit	Reset	Description
19	0x0	**Frozen**: 1b indicates that the WDT is frozen because of the assertion of the global WDT freeze request (when allowed) or because of the assertion of the global timer freeze request
18	0x0	**ErrorForced**: 1b indicates that the error reported to HSM is forced active for testing purposes
17	0x0	**StickyEnabled**: 1b when the counter is enabled and cannot be disabled, set by StickyStart command, cleared by reset only.
16	0x0	**CurrentError**: Current error reported to HSM, this is the output after combining the functional and forced error paths
14:12	0x0	**CurrentExpirationCount**: Current count of expiration since last start operation, cleared by Start command, saturating at 5. **0** = NO_EXPIRY **1** = LOCAL_IRQ **2** = LOCAL_FIQ **3** = REMOTE_IRQ **4** = DEBUG_RESET **5** = SYSTEM_RESET
11:4	0x0	**CurrentCount**: Current value of the downcounter, reloaded with Period by Start command.
3	0x0	**RemoteInterruptStatus**: Current status of remote interrupt, cleared by Start command.
2	0x0	**LocalFIQStatus**: Current status of FIQ, cleared by Start command.
1	0x0	**LocalInterruptStatus**: Current status of local interrupt, cleared by Start command.
0	0x0	**Enabled**: 1b when the counter is active. When Enabled is 1b, the configuration of the WDT is frozen by silently ignoring write attempts to the configuration registers: WDTCR, WDTSCR and WDTFWCR. Set by Start or StickyStart commands, cleared by Disable when StickyEnabled is 0b. **0** = DISABLE **1** = ENABLE

TKE_TOP_WDT2_WDTCMDR_0

Watchdog Timer Command Register

Offset: 0x130008
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_WDTSCR2_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	0x0	**ForceError**: Only working if the Command register is unlocked. 1b sets an internal flop that is ORed with the functional error signal (based on threshold), 0b clears the bit. The state of the flop is shown in the status register.
2	0x0	**StickyStart**: Qualifies the StartCounter bit, if both bits are 1b, the counter is enabled in permanent fashion, i.e., it remains enabled until the next reset, and a disable counter operation has no effect. Ignored if StartCounter is 0b
1	0x0	**DisableCounter**: Only working if the unlock register (WDTUR{w}) has been programmed before with the correct pattern. Writing to the command register always clears the unlock register. If written to 1b, while StartCounter is 0b and the unlock register contains the unlock pattern, the watchdog transitions back to disabled. **0** = DISABLE **1** = ENABLE
0	0x0	**StartCounter**: When written to 1b, enables the counter operation, loads the counter with Period and starts downcounting, resets the expiration count to 0 and clears all status flags. Enabled is always 1b after this. This is also used as Restart. Writing to StartCounter requires unlocking when ChallengeResponseEnable is 1b, and writing to StartCounter is also restricted to windows of time given by WindowedRestartDisableMap when WindowedOperationEnable is 1b. **0** = DISABLE **1** = ENABLE

TKE_TOP_WDT2_WDTUR_0

Watchdog Timer Unlock Register

Offset: 0x13000c
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: TKE_TOP_SCR_WDTSCR2_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UnlockPattern**: The behavior of the Unlock pattern is dependent on the ChallengeResponseEnable configuration bit. When ChallengeResponseEnable is 0b, UnlockPattern must be written with value UNLOCK to allow a write to DisableCounter to take effect. UnlockPattern is reset at each write to the Command register. When ChallengeResponseEnable is 1b, a write to the Command register is only allowed after reading UnlockPattern to get the current value of an LFSR, followed by writing the next value of the LFSR to the UnlockPattern. If the two values match, the unlock is successful. In all cases the LFSR gets (re)initialized with the written value. **50266** = UNLOCK

TKE_TOP_WDT2_WDTSCR_0

Watchdog Timer Skip Configuration Register

Offset: 0x130010
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_WDTSCR2_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,x000,x000,x000)

Bit	Reset	Description
14:12	0x0	**Skip3**: Skip Value at Expiration count 3 At a Downcounter expiration, the expiration level changes as ExpC <= ExpC + 1 + Skip(ExpC), saturating at 5
10:8	0x0	**Skip2**: Skip Value at Expiration count 2
6:4	0x0	**Skip1**: Skip Value at Expiration count 1
2:0	0x0	**Skip0**: Skip Value at Expiration count 0

TKE_TOP_WDT2_WDTFWCR_0

Watchdog Timer Fine Window Configuration Register

Offset: 0x130014
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TKE_TOP_SCR_WDTSCR2_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,x000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
18:16	0x0	**StartLevel**: If restart (i.e., start while WDT is already enabled) is attempted and windowing operation is enabled, restart is rejected if current expiration < StartLevel
7:0	0x0	**StartCount**: If restart (i.e., start while WDT is already enabled) is attempted and windowing operation is enabled and current expiration level == StartLevel, restart is rejected if current downcounter > (Period - StartCount). StartCount is the number of cycles where rejection takes place at the start of that level.

8.3.4.3 Real-Time Clock Registers

RTC2_RTCCR_0

RTC Control Register

Offset: 0x0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: RTC2_SCR_CNTSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**HDBG**: Halt-on-debug. Controls whether a Halt-on-debug signal halts the RTC 0 RTC ignores Halt-on-debug. 1 Asserted Halt-on-debug signal halts RTC **0** = ON **1** = OFF

RTC2_RTCBR_0

RTC Busy Register

Offset: 0x4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: RTC2_SCR_RTCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**STATUS**: When 1, this bit indicates that a write is being transferred to the 32,768Hz clock domain. It is set when a write targeting a register in the 32,768Hz clock domain is received on the APB interface and is cleared when the write is fully transferred to the 32,768Hz clock domain. **0** = FREE **1** = BUSY

RTC2_RTCSCR_0

RTC Second Counter Register

Offset: 0x8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: RTC2_SCR_CNTSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CNT**: Part of the time elapsed measured in seconds, read data comes from a register updated in the APB clock domain every eight 32,768Hz clock cycles.

RTC2_RTCSSCR_0

RTC Shadow Second Counter Register

Offset: 0xc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: RTC2_SCR_RTCSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CNT**: Shadow seconds register is updated over to APB side whenever there is a read to milliseconds counter. This allows software to get a consistent full accuracy value by reading the milliseconds counter then the shadow seconds counter.

RTC2_RTCMCR_0

RTC Millisecond Counter Register

Offset: 0x10
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: RTC2_SCR_RTCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9:0	0x0	**CNT**: Part of the time elapsed measured in milliseconds, read data comes from a register updated in the APB clock domain every eight 32,768Hz clock cycles.

RTC2_RTCSAR0_0

RTC Second Alarm Register 0.

Offset: 0x14
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: RTC2_SCR_RTCSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VALUE**: Interrupt is triggered when the seconds counter matches the value in this register.

RTC2_RTCSAR1_0

RTC Second Alarm Register 1.

Offset: 0x18
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: RTC2_SCR_RTCSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VALUE**: Interrupt is triggered when the seconds counter matches the value in this register.

RTC2_RTCMAR_0

RTC Millisecond Alarm Register

Offset: 0x1c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: RTC2_SCR_RTCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9:0	0x0	**VALUE**: Interrupt is triggered when the milliseconds counter matches the value in this register.

RTC2_RTCSCAR_0

RTC Second Countdown Alarm Register

Offset: 0x20
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: RTC2_SCR_RTCSCR_0
Reset: 0x40000000 (0b0100,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: Enable, when set, enables the countdown process.
30	0x1	**PER**: Periodic, when set, the internal counter is automatically reloaded after reaching 0
29:0	0x0	**PTV**: Timer Present Trigger Value, when enabled, an internal counter is loaded with this field and counted down for every second until it reaches 0. If enabled, an interrupt is then triggered when the count reaches zero.

RTC2_RTCMCAR_0

RTC Millisecond Countdown Alarm Register

Offset: 0x24
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: RTC2_SCR_RTCSCR_0
Reset: 0x40000000 (0b0100,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: Enable, when set, enables the countdown process.
30	0x1	**PER**: Periodic, when set, the internal counter is automatically reloaded after reaching 0
29:0	0x0	**PTV**: Timer Present Trigger Value, when enabled, an internal counter is loaded with this field and counted down for every second until it reaches 0. If enabled, an interrupt is then triggered when the count reaches zero.

RTC2_RTCIER_0

RTC Interrupt Enable Register

Offset: 0x28
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: RTC2_SCR_RTCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**MSEC_CDN_ALARM**: Enable bit for interrupt associated with the millisecond countdown alarm
3	0x0	**SEC_CDN_ALARM**: Enable bit for interrupt associated with the second countdown alarm
2	0x0	**MSEC_ALARM**: Enable bit for interrupt associated with the milliseconds alarm
1:0	0x0	**SEC_ALARM**: Enable bitmap for interrupt associated with each of the second alarms

RTC2_RTCISR_0

RTC Interrupt Status Register

Offset: 0x2c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: RTC2_SCR_RTCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**MSEC_CDN_ALARM**
3	0x0	**SEC_CDN_ALARM**
2	0x0	**MSEC_ALARM**
1:0	0x0	**SEC_ALARM**: Same structure as the RTC Interrupt Enable register, shows the current status of the interrupts before processing. The register is RWC, writing a 1 clears the corresponding status bit. The status is taken before applying the enable mask.

RTC2_RTCIVR_0

RTC Interrupt Valid Register

Offset: 0x30
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: RTC2_SCR_RTCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**MSEC_CDN_ALARM**
3	0x0	**SEC_CDN_ALARM**
2	0x0	**MSEC_ALARM**
1:0	0x0	**SEC_ALARM**: Same structure as the RTC Interrupt Enable register, shows the current status of the interrupts after the enable mask. The register is R only and returns Status & Enable.

RTC2_RTCFIR_0

RTC Force Interrupt Register

Offset: 0x34
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: RTC2_SCR_RTCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**MSEC_CDN_ALARM**
3	0x0	**SEC_CDN_ALARM**
2	0x0	**MSEC_ALARM**
1:0	0x0	**SEC_ALARM**: Note that the semantic of the FIR is slightly different than in LIC. In LIC the FIR register is ORed with the incoming interrupt signals, in RTC the force toggles the source bit itself. Also the FIR in RTC acts like the FIR SET register in LIC. Same structure as the RTC Interrupt Enable register. Write only register for test purposes, forces the corresponding interrupt status bit to 1b.

RTC2_RTCRSR_0

RTC Reference Selection Register

Offset: 0x38
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: RTC2_SCR_CNTSCR_0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xxx1)

Bit	Reset	Description
5:4	0x0	**MBS**: MTSC Bit Select, MTSC[9+MBS] edge used as reference when not free running.
0	0x1	**FR**: Free running, controls the RTC TSC lock behavior When 0b, RTC uses a selected bit of MTSC for reference, recommended mode When 1b, RTC counts on each 32,768Hz clock edge

RTC2_RTCDR_0

The values are coded as real value - 1 (offset 1 encoding)
So the division ratio is (1+RTCDR.N) / (1+RTCDR.D) and must be selected
so that the reference frequency * Ratio = 1000Hz
The reset value is correct for the (default) 32,768Hz clock

RTC Divider Register

Offset: 0x3c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: RTC2_SCR_CNTSCR_0
Reset: 0x0fff007c (0b0000,1111,1111,1111,0000,0000,0111,1100)

Bit	Reset	Description
31:16	0xfff	**D**: Denominator
15:0	0x7c	**N**: Numerator

RTC2_RTCBRCR_0

RTC Base Reference Counter Register

Offset: 0x40
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CNT**: Counts rising edges of 32,768Hz clock, not sensitive to HDBG, i.e., never stops counting, and no configuration

RTC2_AOWDTCR_0

Always On Watchdog Timer Configuration Register

Offset: 0x100
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: RTC2_SCR_AOWDTSCR_0
Reset: 0x00700000 (0b0000,xx00,0111,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**WINDOWEDRESTARTDISABLEMAP**: When WindowedOperationEnable is 1b, restart commands attempted when expiration level has value l become NOP if bit b equals 1b.
25	0x0	**DISALLOWWDTFREEZE**: When 1b, the signal watchdog_freeze_request has no effect When 0b, when the signal watchdog_freeze_request is asserted, the WDT stops decrementing CurrentCount if the WDT is enabled
24:23	0x0	**RESTARTERRORACTION**: If a restart action is attempted but fails, indicate which of four actions to take 0x0: do nothing, silently ignore the restart 0x1: immediately jump to expiration level 4 and reload the downcounter 0x2: immediately jump to expiration level 5 and reload the downcounter 0x3: immediately request a system power on reset (POR) When jumping levels, the normal actions configured per level must take place
22:20	0x7	**ERRORTHRESHOLD**: The WDT logic asserts an error signal to HSM when ExpirationLevel > ErrorThreshold.
19	0x0	**TSCREFERENCEENABLE**: Select as timing reference transitions on a configured TSC bit.

Bit	Reset	Description
18	0x0	**CHALLENGERESPONSEENABLE**: Enable the Challenge Response mode of operation.When 1b, the Unlock register operates as a Challenge Response register.
17	0x0	**WINDOWEDOPERATIONENABLE**: Enable the windowed mode of operation. When 1b, a restart command is rejected when the current expiration level is identified as disallowed in the WindowedRestartDisableMap or the current time is below the window threshold. The rejection is either silent or performs a reset based on RestartErrorAction is identified as disallowed in the WindowedRestartDisableMap.
16	0x0	**SYSTEMPORRESETENABLE**: Enable full system reset at fifth expiration of the counter, mostly at the same level as the signal from PMC asserted in case of a Power On Reset
15	0x0	**SYSTEMDEBUGRESETENABLE**: Enable systemwide reset assertion at fourth expiration of the counter. This reset can be intercepted by debug logic, e.g., to preserve information. Debug reset must not be enabled for AO WDT.
14	0x0	**WAKECOLDENABLE**: Redefined, enable the generation of an SC7 wake event plus clearing of PMC scratch register 0 at third expiration of the counter. This bit is RemoteInterruptEnable for a standard WDT
13	0x0	**WAKEENABLE**: Redefined, enable the generation of an SC7 wake event at second expiration of the counter. This bit is LocalFIQEnable for a standard WDT
12	0x0	**LOCALINTERRUPTENABLE**: Enable normal priority interrupt assertion at first expiration of the counter, connected to local interrupt controller
11:4	0x0	**PERIOD**: Measured in periods of the timer selected as source, this is the reload value, so 0 is treated as maximum period
3:0	0x0	**TIMERSOURCE**: The timer source interpretation is dependent on the value of TscReferenceEnable. TscReferenceEnable is 0b, the timing reference is redefined as a transition on an RTC bit If TimerSource is in 0 or 9, bit TimerSource in the Millisecond counter If TimeSource is in 10 or 15, bit TimerSource-10 of the Second counter TscReferenceEnable is 1b, the timing reference is redefined as a transition on TSC[10+2*TimerSource]

RTC2_AOWDTSR_0

Always On Watchdog Timer Status Register

Offset: 0x104
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: RTC2_SCR_AOWDTSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,x000,0000,0000,0000)

Bit	Reset	Description
19	0x0	**FROZEN**: 1b indicates that the WDT is frozen because of the assertion of the global WDT freeze request (when allowed) or because of the assertion of the global timer freeze request
18	0x0	**ERRORFORCED**: 1b indicates that the error reported to HSM is forced active for testing purposes
17	0x0	**STICKYENABLED**: 1b when the counter is enabled and cannot be disabled, set by StickyStart command, cleared by reset only.
16	0x0	**CURRENTERROR**: Current error reported to HSM, this is the output after combining the functional and forced error paths
14:12	0x0	**CURRENTEXPIRATIONCOUNT**: Current expiration level, cleared by Start command, saturating at 5.
11:4	0x0	**CURRENTCOUNT**: Current value of the down counter, reloaded with Period by Start command.
3	0x0	**WAKECOLDSTATUS**: Redefined as the status of the wake cold signal for the AO WDT, RemoteInterruptStatus for a standard WDT, cleared by Start command
2	0x0	**WAKESTATUS**: Redefined as the status of the wake signal for the AO WDT, LocalFIQStatus for a standard WDT, cleared by Start command.
1	0x0	**LOCALINTERRUPTSTATUS**: Current status of local interrupt, cleared by Start command.
0	0x0	**ENABLED**: 1b when the counter is active. When Enabled is 1b, the configuration of the WDT is frozen by silently ignoring write attempts to the configuration registers: WDTCR, WDTSCR and WDTFWCR. Set by Start or StickyStart commands, cleared by Disable when StickyEnabled is 0b. Set by Start command, cleared by Disable.

RTC2_AOWDTCMDR_0

Always On Watchdog Timer Command Register

Offset: 0x108
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: RTC2_SCR_AOWDTSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	0x0	**FORCEERROR**: Only working if the Command register is unlocked. 1b sets an internal flop that is ORed with the functional error signal (based on threshold), 0b clears the bit. The state of the flop is shown in the status register.
2	0x0	**STICKYSTART**: Qualifies the StartCounter bit, if both bits are 1b, the counter is enabled in permanent fashion, i.e., it remains enabled until the next reset, and a disable counter operation has no effect. Ignored if StartCounter is 0b
1	0x0	**DISABLECOUNTER**: Only working if the disable unlock register has been programmed before with the correct pattern. Writing to the command register always clears the disable unlock register. If written to 1b, while StartCounter is 0b and the unlock register contains the unlock pattern, the watchdog transitions back to disabled.
0	0x0	**STARTCOUNTER**: When written to 1b, enable the counter operation, load the counter with Period and starts downcounting, resets the expiration count to 0 and clears all status flags. Enabled is always 1b after this. This is also used as Restart.

RTC2_AOWDTUR_0

Always On Watchdog Timer Unlock Register

Offset: 0x10c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: RTC2_SCR_AOWDTSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UNLOCKPATTERN**: When ChallengeResponseEnable is 0b, Unlock must be written with value 0x0000_C45A to allow a write to DisableCounter to take effect. The UnlockPattern is reset at each write to the Command register. When ChallengeResponseEnable is 1b, read of the Unlock pattern returns the current value of an LFSR, and writing the next value of the LFSR allows a write to the Command register.

RTC2_AOWDTSCOR_0

`Always On Watchdog Timer Skip Configuration Register`

Offset: 0x110
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: RTC2_SCR_AOWDTSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,x000,x000,x000)

Bit	Reset	Description
14:12	0x0	**SKIP3**: Skip Value at Expiration count 3
10:8	0x0	**SKIP2**: Skip Value at Expiration count 2
6:4	0x0	**SKIP1**: Skip Value at Expiration count 1
2:0	0x0	**SKIP0**: Skip Value at Expiration count 0

RTC2_AOWDTFWCR_0

Watchdog Timer Fine Window Configuration Register

Offset: 0x114
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: RTC2_SCR_AOWDTSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,x000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
18:16	0x0	**STARTLEVEL**: If restart (i.e., start while WDT is already enabled) is attempted and windowing operation is enabled, restart is rejected if current expiration < StartLevel
7:0	0x0	**STARTCOUNT**: If restart (i.e., start while WDT is already enabled) is attempted and windowing operation is enabled and current expiration level == StartLevel, restart is rejected if current downcounter > (Period - StartCount). StartCount is the number of cycles where rejection takes place at the start of that level.

RTC2_CLK_OVR_ON_0
Offset: 0x118
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: RTC2_SCR_RTCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**RTC**: SLCG override bit **0** = DISABLE **1** = ENABLE

8.3.4.4 Time Stamp Counter System Control Registers

TSC_SYSCTR0_CNTCR_0

Counter Control Register

Offset: 0x0
Read/Write: See table below
Parity Protection: See table below
Shadow: N
Secure: TrustZone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,xxxx,xx00)

Bit	R/W	Parity Protection	Reset	Description
8	RO	N	0x0	**FCREQ**: Requested frequency modes table entry. This bit is not used since only one frequency for TSC is supported.

Bit	R/W	Parity Protection	Reset	Description
1	RW	Y	0x0	**HDBG**: Halt-on-debug. Controls whether a Halt-on-debug signal halts the system counter 0: System counter ignores Halt-on-debug. 1: Asserted Halt-on-debug signal halts system counter update. **0** = DISABLE **1** = ENABLE
0	RW	Y	0x0	**EN**: Enables the timestamp system counter 0: System counter disabled 1: System counter enabled When this bit changes from '0' to '1', the master TSC is loaded with the CNTCVx registers value, and starts incrementing from that value. Slave TSC may also start counting depending on STSCCR.DIS, a field in a TSC implementation register, see artsc.h When this bit is '0', master and slave TSC do not count and maintain their value. **0** = DISABLE **1** = ENABLE

TSC_SYSCTR0_CNTSR_0

Counter Status Register
Offset: 0x4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: TrustZone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,xxxx,xx0x)

Bit	Reset	Description
8	0x0	**FCACK**: Frequency change acknowledge Always 0 as only one frequency is exposed to the Arm architectural registers.
1	0x0	**HDBG**: Indicates whether the counter is halted because the Halt-on-Debug signal is asserted: 0: Counter is not halted. 1: Counter is halted. **0** = DISABLE **1** = ENABLE

TSC_SYSCTR0_CNTCV0_0

Counter Count Value[31:0] Register.
Offset: 0x8
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: TrustZone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CV**: Counter value[31:0] Read of this provides the slave TSC[31:0] value at the time of read. When CNTCR[EN]=0, then write to this register is used to initialize the master TSC[31:0] value. When CNTCR[EN]=1, then write to this register has unpredictable behavior.

TSC_SYSCTR0_CNTCV1_0

Counter Count Value[63:32] Register
Offset: 0xc
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: TrustZone Protected
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CV**: Counter value[63:32] Read of this provides the slave TSC[63:32] value at the time of read. When CNTCR[EN]=0, then write to this register is used to initialize the master TSC[63:32] value. When CNTCR[EN]=1, then write to this register has unpredictable behavior.

TSC_SYSCTR0_CNTFID0_0

Frequency table entry register
Offset: 0x20
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x01dcd650 (0b0000,0001,1101,1100,1101,0110,0101,0000)

Bit	Reset	Description
31:0	0x1dcd650	**FV**: Counter frequency value in Hz. Default is set to 31.25MHz. Note that this register can be written only once.

TSC_SYSCTR0_CNTFID1_0
Frequency table end marker
Offset: 0x24
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FV**: Counter frequency value end marker (all-0 read-only)

TSC_SYSCTR0_COUNTERID4_0

COUNTERID11-0 registers are read-only, but to be able to initialize them at boot (by ARM7 or main CPU) they are defined as write once. This allows, if wanted, to emulate the ID as defined by Arm.

Offset: 0xfd0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Peripheral ID value

TSC_SYSCTR0_COUNTERID5_0
Offset: 0xfd4
Read/Write: RO

Parity Protection: N
Shadow: N
Secure: TrustZone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**RESERVED**: Reserved, a read gives all '0's

TSC_SYSCTR0_COUNTERID6_0

Offset: 0xfd8
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: TrustZone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**RESERVED**: Reserved, a read gives all '0's

TSC_SYSCTR0_COUNTERID7_0

Offset: 0xfdc
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: TrustZone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**RESERVED**: Reserved, a read gives all '0's

TSC_SYSCTR0_COUNTERID0_0

Offset: 0xfe0
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Peripheral ID value

TSC_SYSCTR0_COUNTERID1_0

Offset: 0xfe4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Peripheral ID value

TSC_SYSCTR0_COUNTERID2_0

Offset: 0xfe8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Peripheral ID value

TSC_SYSCTR0_COUNTERID3_0

Offset: 0xfec
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Peripheral ID value

TSC_SYSCTR0_COUNTERID8_0

Offset: 0xff0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Component ID value

TSC_SYSCTR0_COUNTERID9_0

Offset: 0xff4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Component ID value

TSC_SYSCTR0_COUNTERID10_0

Offset: 0xff8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Component ID value

TSC_SYSCTR0_COUNTERID11_0

Offset: 0xffc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Component ID value

TSC_FSI_SYSCTR0_CNTCR_0

Counter Control Register

Offset: 0x0
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,xxxx,xx00)

Bit	R/W	Parity Protection	Reset	Description
8	RO	N	0x0	**FCREQ**: Requested frequency modes table entry. This bit is not used since only one frequency for TSC is supported.
1	RW	Y	0x0	**HDBG**: Halt-on-debug. Controls whether a Halt-on-debug signal halts the system counter 0: System counter ignores Halt-on-debug. 1: Asserted Halt-on-debug signal halts system counter update. **0** = DISABLE **1** = ENABLE

Bit	R/W	Parity Protection	Reset	Description
0	RW	Y	0x0	**EN**: Enables the timestamp system counter 0: System counter disabled 1: System counter enabled When this bit changes from '0' to '1', the master TSC is loaded with the CNTCVx registers value, and starts incrementing from that value. Slave TSC may also start counting depending on STSCCR.DIS, a field in a TSC implementation register, see artsc.h When this bit is '0', master and slave TSC do not count and maintain their value. **0** = DISABLE **1** = ENABLE

TSC_FSI_SYSCTR0_CNTSR_0

Counter Status Register
Offset: 0x4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,xxxx,xx0x)

Bit	Reset	Description
8	0x0	**FCACK**: Frequency change acknowledge Always 0 as only one frequency is exposed to the Arm architectural registers.
1	0x0	**HDBG**: Indicates whether the counter is halted because the Halt-on-Debug signal is asserted: 0: Counter is not halted. 1: Counter is halted. **0** = DISABLE **1** = ENABLE

TSC_FSI_SYSCTR0_CNTCV0_0

Counter Count Value[31:0] Register.
Offset: 0x8
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CV**: Counter value[31:0] Read of this provides the slave TSC[31:0] value at the time of read. When CNTCR[EN]=0, then write to this register is used to initialize the master TSC[31:0] value. When CNTCR[EN]=1, then write to this register has unpredictable behavior.

TSC_FSI_SYSCTR0_CNTCV1_0

Counter Count Value[63:32] Register
Offset: 0xc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CV**: Counter value[63:32] Read of this provides the slave TSC[63:32] value at the time of read. When CNTCR[EN]=0, then write to this register is used to initialize the master TSC[63:32] value. When CNTCR[EN]=1, then write to this register has unpredictable behavior.

TSC_FSI_SYSCTR0_CNTFID0_0

Frequency table entry register
Offset: 0x20
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x01dcd650 (0b0000,0001,1101,1100,1101,0110,0101,0000)

Bit	Reset	Description
31:0	0x1dcd650	**FV**: Counter frequency value in Hz. Default is set to 31.25MHz. Note that this register can be written only once.

TSC_FSI_SYSCTR0_CNTFID1_0

Frequency table end marker
Offset: 0x24
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FV**: Counter frequency value end marker (all-0 read-only)

TSC_FSI_SYSCTR0_COUNTERID4_0

COUNTERID11-0 registers are read-only, but to be able to initialize them at boot (by ARM7 or main CPU) they are defined as write once. This allows, if wanted, to emulate the ID as defined by Arm.

Offset: 0xfd0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Peripheral ID value

TSC_FSI_SYSCTR0_COUNTERID5_0

Offset: 0xfd4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**RESERVED**: Reserved, a read gives all '0's

TSC_FSI_SYSCTR0_COUNTERID6_0

Offset: 0xfd8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**RESERVED**: Reserved, a read gives all '0's

TSC_FSI_SYSCTR0_COUNTERID7_0

Offset: 0xfdc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**RESERVED**: Reserved, a read giveS all '0's

TSC_FSI_SYSCTR0_COUNTERID0_0

Offset: 0xfe0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Peripheral ID value

TSC_FSI_SYSCTR0_COUNTERID1_0

Offset: 0xfe4

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Peripheral ID value

TSC_FSI_SYSCTR0_COUNTERID2_0

Offset: 0xfe8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Peripheral ID value

TSC_FSI_SYSCTR0_COUNTERID3_0

Offset: 0xfec
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Peripheral ID value

TSC_FSI_SYSCTR0_COUNTERID8_0

Offset: 0xff0
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Component ID value

TSC_FSI_SYSCTR0_COUNTERID9_0

Offset: 0xff4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Component ID value

TSC_FSI_SYSCTR0_COUNTERID10_0

Offset: 0xff8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Component ID value

TSC_FSI_SYSCTR0_COUNTERID11_0

Offset: 0xffc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Component ID value

8.3.4.5 Time Stamp Counter System Control Registers Continued

The status registers are some of control registers readable via CNTReadBase base address by secure or non-secure accesses. These are the same physical registers that are described in the control register section.

TSC_SYSCTR1_CNTCV0_0
Counter Count Value[31:0] Register.
Offset: 0x8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CV**: Counter value[31:0]

TSC_SYSCTR1_CNTCV1_0
Counter Count Value[63:32] Register
Offset: 0xc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CV**: Counter value[63:32]

TSC_SYSCTR1_COUNTERID4_0
Offset: 0xfd0
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Peripheral ID value

TSC_SYSCTR1_COUNTERID5_0

Offset: 0xfd4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**RESERVED**: Reserved, a read gives all '0's

TSC_SYSCTR1_COUNTERID6_0

Offset: 0xfd8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**RESERVED**: Reserved, a read gives all '0's

TSC_SYSCTR1_COUNTERID7_0

Offset: 0xfdc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**RESERVED**: Reserved, a read gives all '0's

TSC_SYSCTR1_COUNTERID0_0

Offset: 0xfe0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Peripheral ID value

TSC_SYSCTR1_COUNTERID1_0

Offset: 0xfe4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Peripheral ID value

TSC_SYSCTR1_COUNTERID2_0

Offset: 0xfe8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Peripheral ID value

TSC_SYSCTR1_COUNTERID3_0
Offset: 0xfec
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Peripheral ID value

TSC_SYSCTR1_COUNTERID8_0
Offset: 0xff0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Component ID value

TSC_SYSCTR1_COUNTERID9_0
Offset: 0xff4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Component ID value

TSC_SYSCTR1_COUNTERID10_0
Offset: 0xff8

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Component ID value

TSC_SYSCTR1_COUNTERID11_0

Offset: 0xffc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Component ID value

TSC_SYSCTR2_CNTFRQ_0

Arm defines a frame of register to discover the presence of MMIO Arm generic timers, called the CNTCTLBase. The SoC device does not have any of these registers but the frame is still required. When there are no target registers like here, the frame only contains read only registers. Like in sysctr1, the counter ID registers are read only and reflect the values programmed in sysctr0. The register below follows the format specified in the Arm Architecture Reference Manual with the name "The CNTCTLBase frame", and combines the CNTCTLBase memory map with the different register definitions in later tables.

Counter-timer Frequency
Offset: 0x0
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x01dcd650 (0b0000,0001,1101,1100,1101,0110,0101,0000)

Bit	Reset	Description
31:0	0x1dcd650	**VAL**: Counter frequency value in Hz, set to 31.25MHz.

TSC_SYSCTR2_CNTNSAR_0

Counter-timer Non-secure Access Register
Offset: 0x4
Read/Write: RO
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VAL**: Non-secure access to frame 0 to 7, coded as RES0 for each as no frame is implemented

TSC_SYSCTR2_CNTTIDR_0

Counter-timer Timer ID Register.
Offset: 0x8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VAL**: Frame features for frame 0 to 7, all coded as 0 as no frame is implemented

CNTACR0-7 are not implemented as their corresponding frames are not implemented.

CNTVOFF0-7 are not implemented as their corresponding frames are not implemented.

TSC_SYSCTR2_COUNTERID0_0

The next register must be at offset 0xFD0 = 4048, the previous was at offset 0x8, so need to skip (4048 - 8) / 4 - 1 positions COUNTERID11-0 registers are read-only, they reflect the values present at the same offset in sysctr0.

Offset: 0xfd0
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**

TSC_SYSCTR2_COUNTERID1_0
Offset: 0xfd4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**

TSC_SYSCTR2_COUNTERID2_0
Offset: 0xfd8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**

TSC_SYSCTR2_COUNTERID3_0
Offset: 0xfdc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**

TSC_SYSCTR2_COUNTERID4_0
Offset: 0xfe0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**

TSC_SYSCTR2_COUNTERID5_0
Offset: 0xfe4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**RESERVED**: Reserved, a read gives all '0's

TSC_SYSCTR2_COUNTERID6_0
Offset: 0xfe8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**RESERVED**: Reserved, a read gives all '0's

TSC_SYSCTR2_COUNTERID7_0
Offset: 0xfec
Read/Write: RO

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**RESERVED**: Reserved, a read gives all '0's

TSC_SYSCTR2_COUNTERID8_0

Offset: 0xff0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**

TSC_SYSCTR2_COUNTERID9_0

Offset: 0xff4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**

TSC_SYSCTR2_COUNTERID10_0

Offset: 0xff8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**

TSC_SYSCTR2_COUNTERID11_0

Offset: 0xffc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**

TSC_FSI_SYSCTR1_CNTCV0_0

Counter Count Value[31:0] Register.
Offset: 0x1008
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CV**: Counter value[31:0]

TSC_FSI_SYSCTR1_CNTCV1_0

Counter Count Value[63:32] Register
Offset: 0x100c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CV**: Counter value[63:32]

TSC_FSI_SYSCTR1_COUNTERID4_0

Offset: 0x1fd0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Peripheral ID value

TSC_FSI_SYSCTR1_COUNTERID5_0

Offset: 0x1fd4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**RESERVED**: Reserved, a read gives all '0's

TSC_FSI_SYSCTR1_COUNTERID6_0

Offset: 0x1fd8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**RESERVED**: Reserved, a read gives all '0's

TSC_FSI_SYSCTR1_COUNTERID7_0

Offset: 0x1fdc

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**RESERVED**: Reserved, a read gives all '0's

TSC_FSI_SYSCTR1_COUNTERID0_0

Offset: 0x1fe0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Peripheral ID value

TSC_FSI_SYSCTR1_COUNTERID1_0

Offset: 0x1fe4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Peripheral ID value

TSC_FSI_SYSCTR1_COUNTERID2_0

Offset: 0x1fe8
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Peripheral ID value

TSC_FSI_SYSCTR1_COUNTERID3_0

Offset: 0x1fec
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Peripheral ID value

TSC_FSI_SYSCTR1_COUNTERID8_0

Offset: 0x1ff0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Component ID value

TSC_FSI_SYSCTR1_COUNTERID9_0

Offset: 0x1ff4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Component ID value

TSC_FSI_SYSCTR1_COUNTERID10_0

Offset: 0x1ff8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Component ID value

TSC_FSI_SYSCTR1_COUNTERID11_0

Offset: 0x1ffc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Component ID value

TSC_FSI_SYSCTR2_CNTFRQ_0

Arm defines a frame of register to discover the presence of MMIO Arm generic timers, called the CNTCTLBase. NVIDIA System on Chip (SoC) devices do not have any of these registers but the frame is still required. When there are no target registers like here, the frame only contains read only registers. Like in sysctr1, the counter ID registers are read only and reflect the values programmed in sysctr0. The register below follows the format specified in the Arm Architecture Reference Manual with name "The CNTCTLBase frame", and combines the CNTCTLBase memory map with the different register definitions in later tables.

Counter-timer Frequency

Offset: 0x2000
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x02625a00 (0b0000,0010,0110,0010,0101,1010,0000,0000)

Bit	Reset	Description
31:0	0x2625a00	**VAL**: Counter frequency value in Hz, set to 40MHz.

TSC_FSI_SYSCTR2_CNTNSAR_0
Counter-timer Non-secure Access Register
Offset: 0x2004
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VAL**: Non-secure access to frame 0 to 7, coded as RES0 for each, as no frame is implemented

TSC_FSI_SYSCTR2_CNTTIDR_0
Counter-timer Timer ID Register.
Offset: 0x2008
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VAL**: Frame features for frame 0 to 7, all coded as 0, as no frame is implemented

CNTACR0-7 are not implemented, as their corresponding frames are not implemented.

CNTVOFF0-7 are not implemented, as their corresponding frames are not implemented.

TSC_FSI_SYSCTR2_COUNTERID0_0

The next register must be at offset 0xFD0 = 4048, the previous was at offset 0x8, so need to skip (4048 - 8) / 4 - 1 positions.

COUNTERID11-0 registers are read-only, they reflect the values present at the same offset in sysctr0.

Offset: 0x2fd0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**

TSC_FSI_SYSCTR2_COUNTERID1_0
Offset: 0x2fd4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**

TSC_FSI_SYSCTR2_COUNTERID2_0
Offset: 0x2fd8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**

TSC_FSI_SYSCTR2_COUNTERID3_0

Offset: 0x2fdc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**

TSC_FSI_SYSCTR2_COUNTERID4_0

Offset: 0x2fe0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**

TSC_FSI_SYSCTR2_COUNTERID5_0

Offset: 0x2fe4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**RESERVED**: Reserved, a read gives all '0's

TSC_FSI_SYSCTR2_COUNTERID6_0

Offset: 0x2fe8
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**RESERVED**: Reserved, a read gives all '0's

TSC_FSI_SYSCTR2_COUNTERID7_0

Offset: 0x2fec
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**RESERVED**: Reserved, a read gives all '0's

TSC_FSI_SYSCTR2_COUNTERID8_0

Offset: 0x2ff0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**

TSC_FSI_SYSCTR2_COUNTERID9_0

Offset: 0x2ff4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**

TSC_FSI_SYSCTR2_COUNTERID10_0

Offset: 0x2ff8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**

TSC_FSI_SYSCTR2_COUNTERID11_0

Offset: 0x2ffc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**

TSC_FSI_FTSCACR_0

Offset: 0x3000
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000002 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,xx10)

Bit	Reset	Description
7:4	0x0	**ABS**: Adjust Bit Select, an edge on MTSC[9+ABS] triggers one adjust operation
1	0x1	**DIR**: Direction of the adjustment, 0 indicates A0 = -1, 1 indicates A0 = +1

Bit	Reset	Description
0	0x0	**EN**: Enables frequency error correction **0** = DISABLE **1** = ENABLE

TSC_FSI_FTSCADR_0

Offset: 0x3004
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,0000,0000,0000,xxxx,0000,0000,0000)

Bit	Reset	Description
27:16	0x0	**D1**: Denominator of the adjust NCO
11:0	0x0	**N1**: Numerator of the adjust NCO

TSC_FSI_FTSCANNR_0

Offset: 0x3008
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00190000 (0bxxxx,0000,0001,1001,xxxx,0000,0000,0000)

Bit	Reset	Description
27:16	0x19	**M0**: Fractional part of the numerator of the adjustable NCO
11:0	0x0	**R0**: Integer part of the numerator of the adjustable NCO

TSC_FSI_FTSCANDR_0

Offset: 0x300c
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: 0
Reset: 0x00000020 (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0010,0000)

Bit	Reset	Description
11:0	0x20	**D0**: The denominator of the adjustable NCO

TSC_FSI_FTSCCNTCV0_0

FSI TSC Counter Count Value Register

Offset: 0x3010
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CV**: Read of this register provides FSI TSC[31:0] value at the time of read (after synchronization to APB clock).

TSC_FSI_FTSCCNTCV1_0

FSI TSC Counter Count Value Register

Offset: 0x3014
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CV**: Read of this register provides {8h0,FSI TSC[55:32]} value at the time of read (after synchronization to APB clock).

TSC_FSI_CLK_OVR_ON_0

TSC SLCG local override

Offset: 0x3018
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**OSC_CLK**: SLCG override bit **0** = DISABLE **1** = ENABLE

USEC Counter

TSC_FSI_USEC_CNTR_USECCVR_0

USECCVR, Microsecond Counter Value Register

Offset: 0x4000
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CNT**: Number of usec pulses since last system reset

TSC_FSI_USEC_CNTR_USECCCR_0

USECCCR, Microsecond Counter Configuration Register

Offset: 0x4004
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000027 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0010,0111)

Bit	Reset	Description
15:8	0x0	**Dividend**: Numerator is Dividend + 1, also known as N
7:0	0x27	**Divisor**: Denominator is Divisor + 1, default value corresponds to 40MHz OSC in FSI.

TSC_FSI_USEC_CNTR_USECCRSR_0

USECCRSR, Microsecond Counter Reference Selection Register

Offset: 0x4010
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xxx1)

Bit	Reset	Description
5:4	0x0	**MBS**: STSC Bit Select, STSC[1+MBS] edge used as reference when not free running.
0	0x1	**FR**: Free running, controls the microsecond counter TSC lock behavior When 0b, the microsecond locks the usec reference pulse to a selected bit of STSC[4:1] using the adjust signal; this is the recommended mode When 1b, Usec Counter counts on each edge of OSC in free running fashion, legacy mode

TSC_FSI_USEC_CNTR_USECCFR_0

USECCFR, Microsecond Counter Freeze Register

Offset: 0x403c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**HDBG**: Halt-on-debug. Controls whether a Halt-on-debug signal halts the microsecond and associated divider logic 0: Microsecond counter ignores Halt-on-debug. 1: Asserted Halt-on-debug signal halts the ration divider, automatically stopping the microsecond counter and the timing reference signals. Care must be taken as timing reference signals are in the inactive state during halt.

TSC_FSI_USEC_CNTR_CLK_OVR_ON_0
Offset: 0x4040
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**OSC**: SLCG override bit **0** = DISABLE **1** = ENABLE

8.3.4.6 Time Stamp Counter Registers

TSC_MTSCACR_0
Offset: 0x0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000002 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,xx10)

Bit	Reset	Description
7:4	0x0	**ABS**: Adjust Bit Select, an edge on MTSC[9+ABS] triggers one adjust operation
1	0x1	**DIR**: Direction of the adjustment, 0 indicates A0 = -1, 1 indicates A0 = +1
0	0x0	**EN**: Enables frequency error correction **0** = DISABLE **1** = ENABLE

TSC_MTSCADR_0

Offset: 0x4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,0000,0000,0000,xxxx,0000,0000,0000)

Bit	Reset	Description
27:16	0x0	**D1**: Parameters defining the adjust frequency for the master TSC NCO
11:0	0x0	**N1**: Parameters defining the adjust frequency for the master TSC NCO

TSC_MTSCANNR_0

Offset: 0x8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x056503b9 (0bxxxx,0101,0110,0101,xxxx,0011,1011,1001)

Bit	Reset	Description
27:16	0x565	**M0**: Parameters defining the numerator of the master TSC NCO
11:0	0x3b9	**R0**: Parameters defining the numerator of the master TSC NCO

TSC_MTSCANDR_0

Offset: 0xc
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000800 (0bxxxx,xxxx,xxxx,xxxx,xxxx,1000,0000,0000)

Bit	Reset	Description
11:0	0x800	**D0**: The denominator of the master TSC NCO

TSC_MTSCCNTCV0_0

Master TSC Counter Count Value Register

Offset: 0x10
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CV**: Read of this register provides Master TSC[31:0] value at the time of read (after synchronization to APB clock).

TSC_MTSCCNTCV1_0

Master TSC Counter Count Value Register

Offset: 0x14
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CV**: Read of this register provides {8h0,Master TSC[55:32]} value at the time of read (after synchronization to APB clock).

TSC_STSCCR_0

Slave TSC Control Register

Offset: 0x100
Read/Write: R/W
Parity Protection: See table below
Shadow: N

SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,xx0x,0000)

Bit	Parity Protection	Reset	Description
8	N	0x0	**FCREQ**: Frequency Change request; note that the bit with the same name in the ARM registers is RO as the notional TSC update frequency is constant. When this bit changes value, it triggers the hardware state machine that switches the clock paths, normally resulting in a change of the frequency of RefClk. The change is completed when STSCSR.FCACK returns the same value as STSCCR.FCREQ.
5	Y	0x0	**FCREQ_DISABLE**: Do not trigger the hardware state machine triggered at a change of STSCCR.FCREQ. This allows for an alternate software based clock change. **0** = DISABLE **1** = ENABLE
3	Y	0x0	**SRC**: If FP is 1b, indicates which RefClk source path is selected.
2	Y	0x0	**FP**: Force the source selection value for RefClk, overriding STSCCR.FCREQ, for debug purposes. The selected path is controlled by the next field.
1	Y	0x0	**DIS**: Disable, combined with the CNTCR.EN field, the slave TSC increments when CNTCR.EN is 1 and STSCCR.DIS is 0.
0	N	0x0	**INIT**: Force the slave TSC to take its value from the master TSC. The transfer from master TSC to slave TSC occurs once for each write of this bit to 1, the value returned is always 0. Software must trigger INIT at least once before enabling the serial distribution of TSC (via TSCDCR.EN), at least five cycles of the 32,768Hz clock after enabling MTSC (i.e., enabling TSC itself) at cold boot, and at least five cycles of the reference clock after the reference is correctly configured.

TSC_STSCRSR_0

Slave TSC Reference Selection Register

Offset: 0x104
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)
PROD: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	PROD	Description
1	0x0	_NONE_	**RANGE**: Select between two lock ranges. Used when FR == 0b. The lock range is proportional to the frequency of adjust request: When 0b, send adjust request at 32,768Hz, this does not work well with the recommended configuration, with lock range about 1.4 ppm When 1b, send adjust request at TSC ref clk frequency divided by 8, lock range is about 160 ppm for recommended configuration **0** = SMALL **1** = LARGE
0	0x0	0x1	**FR**: Free running, controls the slave TSC dynamical lock behavior When 0b, the Slave TSC locks itself on Master TSC, recommended mode When 1b, the Slave TSC operates independently of Master TSC after initialization **0** = LOCKEN **1** = LOCKDIS

TSC_STSCIR0_0

Slave TSC Increment Register0

Offset: 0x108
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0001)

Bit	Reset	Description
3:0	0x1	**K**: TSC is incremented by K at each update clock edge, programming 0 is allowed and freezes the slave TSC without stopping RefClk. One register per path for the RefClk source. This is for osc clock path.

TSC_STSCIR1_0

Slave TSC Increment Register1

Offset: 0x10c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0001)

Bit	Reset	Description
3:0	0x1	**K**: TSC is incremented by K at each update clock edge, programming 0 is allowed and freezes the slave TSC without stopping RefClk. One register per path for the RefClk source. This is for osc clock path.

TSC_STSCSR_0

Slave TSC Status Register

Offset: 0x110
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
8	0x0	**FCACK**: Frequency change acknowledge. This field reflects the current frequency in effect; it is always the value of FCREQ except during the transition period between two frequencies.
7:0	0x0	**OFS**: Offset, difference between master and slave TSC, measured at a change of master TSC. This is calculated as MTSC[7:0] - STSC[7:0] with the result also on 8 bits to get circular arithmetic

TSC_TSCDCR_0

TSC Distribution Control Register

Offset: 0x114
Read/Write: R/W
Parity Protection: See table below
Shadow: N

SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,xxxx,xx00,x000,0000)
PROD: 0x00000011 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1,0001)

Bit	Parity Protection	Reset	PROD	Description
19:16	Y	0x0	_NONE_	**SYNCEP**: Period between synchronization format in exponential notation, a periodic synchronization message is sent when TSC[10+SYNCEP] toggles from 1 to 0. Note that all bits below the selected bit also transition from 1 to 0 at the same time
9	Y	0x0	_NONE_	**RES**: Reserved bit inserted in all synchronization messages.
8	Y	0x0	_NONE_	**HALT_NO_JUMP**: Allows to halt TSC TN without stopping reference to avoid jump at the end of Halt period. If HALT_NO_JUMP=1, future sync msgs are ignored until reset is asserted. **0** = DISABLE **1** = ENABLE
6	Y	0x0	_NONE_	**DISABLE_PERIODIC**: Disable the sending of periodic messages, software can still force the sending of synchronization messages via the SYNC bit. **0** = DISABLE **1** = ENABLE
5	N	0x0	_NONE_	**SYNC**: Force one synchronization message at each write of this bit to 1, the value returned is always 0
4	Y	0x0	0x1	**EN**: Enable the distribution network **0** = DISABLE **1** = ENABLE
3:0	Y	0x0	0x1	**K**: The K value inserted in software triggered synchronization messages; in other cases, the K value in the synchronization message is the currently active K value, with special attention at RefClk source change

TSC_TSCDHCR_0

TSC Distribution Halt Configuration Register The register bits are selected to allow future extensions into up to 16 groups

Offset: 0x118
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
16	0x0	**STOP_CPU_TSC**: Unconditionally stop the CPU terminal nodes
0	0x0	**HDBG_CPU**: Halt CPU terminal nodes in debug mode

TSC_CLK_OVR_ON_0

Offset: 0x11c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**CK32K**: SLCG override bit **0** = DISABLE **1** = ENABLE

TSC_INTERRUPT_STATUS_0

Offset: 0x120
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,xxx0,0000)

Bit	Reset	Description
11	0x0	**GEN3**: RO Alias of GEN{i}_STATUS.INTERRUPT_STATUS, one bit per generator
10	0x0	**GEN2**: RO Alias of GEN{i}_STATUS.INTERRUPT_STATUS, one bit per generator
9	0x0	**GEN1**: RO Alias of GEN{i}_STATUS.INTERRUPT_STATUS, one bit per generator

Bit	Reset	Description
8	0x0	**GEN0**: RO Alias of GEN{i}_STATUS.INTERRUPT_STATUS, one bit per generator
4	0x0	**UNLOCKED**: The hardware locking loop has detected an UNLOCKED_EVENT and is enabled for interrupt generation
3	0x0	**CAPTURED_MISC**
2	0x0	**CAPTURED_MTSC**
1	0x0	**CAPTURED_PTX**
0	0x0	**CAPTURED_GPS**: The corresponding capture branch has information available and is enabled for interrupt generation

TSC_CAPTURE_CONFIGURATION_GPS_0

CAPTURE Registers

Array of CAPTURE Registers for each external reference for tracking.

Signals from AON GPIO used as capture source (mainly used for GPS).

Offset: 0x140
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,xx00,xx00)

Bit	Reset	Description
15:8	0x0	**SRC_SELECT**: Configure the bit selection mux in each branch
5:4	0x0	**EDGE_SELECT**: Bit 0 enable the rising edge, bit 1 the falling edge, all combinations allowed: 00b: no edge, 01b: rising edge only, 10b: falling edge only, 11b: all edges
1	0x0	**INTR_ENABLE**: When set, enable the generation of an interrupt at a new capture, see CAPTURED
0	0x0	**ENABLE**: Fully disable the branch, including the ability to feed the hardware lock mechanism

TSC_CAPTURE_CONTROL_GPS_0

Offset: 0x144
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**UPDATE_VALUE**: At the selected edge, capture the value of TSC in the capture register.

TSC_CAPTURE_STATUS_GPS_0

Offset: 0x148
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,xxx0)

Bit	R/W	Parity Protection	Reset	Description
7:4	RO	N	0x0	**CAPTURE_COUNT**: Can be used for polling purposes, increased modulo 16 at each capture
0	RW	Y	0x0	**CAPTURED**: RW1C, set at any capture, clear by writing 1 to it. An interrupt is generated when CAPTURED is 1b and the corresponding INTR_ENABLE is 1b.

TSC_CAPTURE_VALUE0_GPS_0

Offset: 0x14c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CNT**: 32 bit LSB of last captured TSC value. There is no specific hardware protection to insure atomicity, software can either use consistency techniques to detect errors or suspend then resume the register update while reading the VALUE registers

TSC_CAPTURE_VALUE1_GPS_0

Offset: 0x150
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:0	0x0	**CNT**: 24 bit MSB of last captured TSC value. There is no specific hardware protection to insure atomicity, software can either use consistency techniques to detect errors or suspend then resume the register update while reading the VALUE registers

TSC_CAPTURE_CONFIGURATION_PTX_0

Eth PTP or PCIe PTM signal as capture source.

Offset: 0x15c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,xx00,xx00)

Bit	Reset	Description
15:8	0x0	**SRC_SELECT**: Configure the bit selection mux in each branch
5:4	0x0	**EDGE_SELECT**: Bit 0 enable the rising edge, bit 1 the falling edge, all combinations allowed: 00b: no edge, 01b: rising edge only, 10b: falling edge only, 11b: all edges
1	0x0	**INTR_ENABLE**: When set, enable the generation of an interrupt at a new capture, see CAPTURED

Bit	Reset	Description
0	0x0	**ENABLE**: Fully disable the branch, including the ability to feed the hardware lock mechanism

TSC_CAPTURE_CONTROL_PTX_0

Offset: 0x160
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**UPDATE_VALUE**: At the selected edge, capture the value of TSC in the capture register.

TSC_CAPTURE_STATUS_PTX_0

Offset: 0x164
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,xxx0)

Bit	R/W	Parity Protection	Reset	Description
7:4	RO	N	0x0	**CAPTURE_COUNT**: Can be used for polling purposes, increased modulo 16 at each capture
0	RW	Y	0x0	**CAPTURED**: RW1C, set at any capture, clear by writing 1 to it. An interrupt is generated when CAPTURED is 1b and the corresponding INTR_ENABLE is 1b.

TSC_CAPTURE_VALUE0_PTX_0

Offset: 0x168
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CNT**: 32 bit LSB of last captured TSC value. There is no specific hardware protection to insure atomicity, software can either use consistency techniques to detect errors or suspend then resume the register update while reading the VALUE registers

TSC_CAPTURE_VALUE1_PTX_0

Offset: 0x16c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:0	0x0	**CNT**: 24 bit MSB of last captured TSC value. There is no specific hardware protection to insure atomicity, software can either use consistency techniques to detect errors or suspend then resume the register update while reading the VALUE registers

TSC_CAPTURE_CONFIGURATION_MTSC_0

Traditional MTSC used as capture source.

Offset: 0x178
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,xx00,xx00)

Bit	Reset	Description
15:8	0x0	**SRC_SELECT**: Configure the bit selection mux in each branch
5:4	0x0	**EDGE_SELECT**: Bit 0 enable the rising edge, bit 1 the falling edge, all combinations allowed: 00b: no edge, 01b: rising edge only, 10b: falling edge only, 11b: all edges

Bit	Reset	Description
1	0x0	**INTR_ENABLE**: When set, enable the generation of an interrupt at a new capture, see CAPTURED
0	0x0	**ENABLE**: Fully disable the branch, including the ability to feed the hardware lock mechanism

TSC_CAPTURE_CONTROL_MTSC_0

Offset: 0x17c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**UPDATE_VALUE**: At the selected edge, capture the value of TSC in the capture register.

TSC_CAPTURE_STATUS_MTSC_0

Offset: 0x180
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,xxx0)

Bit	R/W	Parity Protection	Reset	Description
7:4	RO	N	0x0	**CAPTURE_COUNT**: Can be used for polling purposes, increased modulo 16 at each capture
0	RW	Y	0x0	**CAPTURED**: RW1C, set at any capture, clear by writing 1 to it. An interrupt is generated when CAPTURED is 1b and the corresponding INTR_ENABLE is 1b.

TSC_CAPTURE_VALUE0_MTSC_0

Offset: 0x184
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CNT**: 32 bit LSB of last captured TSC value. There is no specific hardware protection to insure atomicity, software can either use consistency techniques to detect errors or suspend then resume the register update while reading the VALUE registers

TSC_CAPTURE_VALUE1_MTSC_0

Offset: 0x188
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:0	0x0	**CNT**: 24 bit MSB of last captured TSC value. There is no specific hardware protection to insure atomicity, software can either use consistency techniques to detect errors or suspend then resume the register update while reading the VALUE registers

TSC_CAPTURE_CONFIGURATION_MISC_0

MISC sources - Only CK32; removed JESD_SYSREF, JESD_LMFC_LEMC_CLK, JESD_COUNTER

Offset: 0x194
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,xx00,xx00)

Bit	Reset	Description
15:8	0x0	**SRC_SELECT**: Configure the bit selection mux in each branch
5:4	0x0	**EDGE_SELECT**: Bit 0 enable the rising edge, bit 1 the falling edge, all combinations allowed: 00b: no edge, 01b: rising edge only, 10b: falling edge only, 11b: all edges

Bit	Reset	Description
1	0x0	**INTR_ENABLE**: When set, enable the generation of an interrupt at a new capture, see CAPTURED
0	0x0	**ENABLE**: Fully disable the branch, including the ability to feed the hardware lock mechanism

TSC_CAPTURE_CONTROL_MISC_0

Offset: 0x198
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**UPDATE_VALUE**: At the selected edge, capture the value of TSC in the capture register.

TSC_CAPTURE_STATUS_MISC_0

Offset: 0x19c
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,xxx0)

Bit	R/W	Parity Protection	Reset	Description
7:4	RO	N	0x0	**CAPTURE_COUNT**: Can be used for polling purposes, increased modulo 16 at each capture
0	RW	Y	0x0	**CAPTURED**: RW1C, set at any capture, clear by writing 1 to it. An interrupt is generated when CAPTURED is 1b and the corresponding INTR_ENABLE is 1b.

TSC_CAPTURE_VALUE0_MISC_0

Offset: 0x1a0
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CNT**: 32 bit LSB of last captured TSC value. There is no specific hardware protection to insure atomicity, software can either use consistency techniques to detect errors or suspend then resume the register update while reading the VALUE registers

TSC_CAPTURE_VALUE1_MISC_0

Offset: 0x1a4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:0	0x0	**CNT**: 24 bit MSB of last captured TSC value. There is no specific hardware protection to insure atomicity, software can either use consistency techniques to detect errors or suspend then resume the register update while reading the VALUE registers

TSC_LOCKING_CONFIGURATION_0

Hardware and Software Locking Registers

Offset: 0x1e8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx00,0000)

Bit	Reset	Description
9:8	0x0	**SRC_SELECT**: Select which of the available capture branches is used as reference for hardware locking. The mapping follows the naming order of the branches, that is 0:GPS, 1:PTX, 2:MTSC, 3:MISC

Bit	Reset	Description
5:4	0x0	**EDGE_SELECT**: Select which edge is used in the locking mechanism, encoded as the edge select for capture purposes.
3	0x0	**SYNC_START_ENABLE**: Starts the counter at the next edge of the reference source
2	0x0	**INVERT_ADJUST_SIGN**: For testing, invert the sign of the measured difference, making the loop unstable
1	0x0	**INT_ENABLE**: Enable the generation of interrupt linked to the hardware lock operation, in particular when losing lock
0	0x0	**ENABLE**: Enable the hardware lock mechanism, i.e., the dynamic adjustment of the frequency and direction of adjust indications sent to the TSC reference clock generator.

TSC_LOCKING_CONTROL_0

Offset: 0x1ec
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**UPDATE_DIFF_DISABLE**: Note that this also disturbs the hardware locking mechanism. As difference values are contained in single registers, the use of this bit is generally not required but allow the atomic reading of a sequence of difference registers by software.
0	0x0	**ALIGN**: When set, and ALIGNED is 0, set the reference value equal to the TSC value at a capture edge; this forces the measured difference to 0. Alignment can be done at any time, but normally only once at start.

TSC_LOCKING_STATUS_0

Offset: 0x1f0
Read/Write: See table below
Parity Protection: See table below
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,x000)

Bit	R/W	Parity Protection	Reset	Description
7:4	RO	N	0x0	**CAPTURE_DIFF_COUNT**: Can be used for polling purposes to snoop the difference register. This is also the value used to define the cycle for the fast convergence algorithm
2	RW	Y	0x0	**UNLOCKED_EVENT**: RW1C, set when LOCKED is 0b, cleared by software writing 1b. An interrupt is generated when UNLOCKED_EVENT is 1b and INT_ENABLE is 1b
1	RO	N	0x0	**LOCKED**: Current lock status, 1 if abs(diff) < locked_threshold
0	RW	Y	0x0	**ALIGNED**: RW1C, see ALIGN for the general description.

TSC_LOCKING_REF_FREQUENCY_CONFIGURATION_0

Offset: 0x1f4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x01dcd650 (0bxxxx,0001,1101,1100,1101,0110,0101,0000)

Bit	Reset	Description
27:0	0x1dcd650	**INCREMENT**: The reset value corresponds to a one second period, the maximum reference period supported is a little less than 8.6 seconds.

TSC_LOCKING_DIFF_CONFIGURATION_0

Offset: 0x1f8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x0000001f (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0001,1111)

Bit	Reset	Description
15:0	0x1f	**LOCKED_THRESHOLD**: Lock is deemed lost when the absolute value of DIFF_0 exceeds the threshold, the reset value corresponds to a deviation of about one microsecond.

TSC_LOCKING_DIFF_STATUS_0_0

Offset: 0x1fc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:0	0x0	**CNT**: Signed difference between captured and reference value, this corresponds to measured phase error. The hardware lock attempts to make the difference 0. The different registers form a shift register, with _0 the most recent difference. The hardware locking mechanism requires at least the two last differences, but four are provided to allow for more sophisticated software algorithms if wanted. The maximum phase error is about 270ms, with no protection against overflow.

TSC_LOCKING_DIFF_STATUS_1_0

Offset: 0x200
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:0	0x0	**CNT**: Signed difference between captured and reference value, this corresponds to measured phase error. The hardware lock attempts to make the difference 0. The different registers form a shift register, with _0 the most recent difference. The hardware locking mechanism requires at least the two last differences, but four are provided to allow for more sophisticated software algorithms if wanted. The maximum phase error is about 270ms, with no protection against overflow.

TSC_LOCKING_DIFF_STATUS_2_0

Offset: 0x204
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:0	0x0	**CNT**: Signed difference between captured and reference value, this corresponds to measured phase error. The hardware lock attempts to make the difference 0. The different registers form a shift register, with _0 the most recent difference. The hardware locking mechanism requires at least the two last differences, but four are provided to allow for more sophisticated software algorithms if wanted. The maximum phase error is about 270ms, with no protection against overflow.

TSC_LOCKING_DIFF_STATUS_3_0

Offset: 0x208
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:0	0x0	**CNT**: Signed difference between captured and reference value, this corresponds to measured phase error. The hardware lock attempts to make the difference 0. The different registers form a shift register, with _0 the most recent difference. The hardware locking mechanism requires at least the two last differences, but four are provided to allow for more sophisticated software algorithms if wanted. The maximum phase error is about 270ms, with no protection against overflow.

TSC_LOCKING_ADJUST_CONFIGURATION_0

Offset: 0x20c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x001fffff (0bxxxx,xxxx,xxx1,1111,1111,1111,1111,1111)

Bit	Reset	Description
20:0	0x1fffff	**MAX_NUM**: Maximum unsigned value allowed for the absolute value of the numerator. This must always be smaller or equal than DEN/8 = 224/8 = 221 as the TSC CAR logic requires at least eight clock cycles between two adjust requests.

TSC_LOCKING_FAST_ADJUST_CONFIGURATION_0

Offset: 0x210
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,xx00,xxx0)

Bit	Reset	Description
31:16	0x0	**THRESHOLD**: Only apply the fast convergence algorithm if the current magnitude of the difference exceeds the threshold.
15:8	0x0	**K_INT**: Integer part of the factor used to calculate the delta to apply to NUM when the fast convergence algorithm is enabled.
5:4	0x0	**M**: Number of right shift used in conjunction with K_INT
0	0x0	**FAST_ENABLE**: Enable the fast convergence algorithm, subject to ENABLE being 1

TSC_LOCKING_ADJUST_NUM_CONTROL_0

Offset: 0x214
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0b0xxx,xxxx,xx00,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**TRANSFER_TO_HW**: When TRANSFER_TO_HW is 1b, the initial values are loaded in the underlying hardware logic. If a software controlled loop is implemented, TRANSFER_TO_HW can be kept at 1 and the other field modified based on the software based locking algorithm.

Bit	Reset	Description
21:0	0x0	**NUM**: Initial signed value of the numerator controlling the adjust frequency

TSC_LOCKING_ADJUST_DELTA_CONTROL_0
Offset: 0x218
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxx0,0000,0000,0000,0000,0000)

Bit	Reset	Description
20:0	0x0	**DELTA_NUM**: Unsigned value to add or remove from the current NUM when hardware locking is enabled and a slow update is required.

TSC_LOCKING_ADJUST_STATUS_0
Offset: 0x21c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xx00,0000,0000,0000,0000,0000)

Bit	Reset	Description
21:0	0x0	**NUM**: Current signed value of the numerator controlling the adjust frequency, the value may change when hardware lock is enabled

TSC Waveform Generators

These sets of registers allow for generation of software controlled waveforms.

TSC_GEN0_CTRL_0

Offset: 0x380
Read/Write: WO
Parity Protection: Y
Shadow: N

SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**INITIAL_VAL**: Start value for current Generator
0	0x0	**ENABLE**: When ENABLE goes from 0->1, the output level is unconditionally set to INITIAL_VAL. The FSM goes into the WAITING state and starts processing EDGE registers when TSC becomes equal or greater than the value programmed in TSC_START. When ENABLE is 0, the FSM is stopped in an IDLE state (WAITING and RUNNING both 0b)

TSC_GEN0_START0_0

Offset: 0x384
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LSB_VAL**: Indicates the start of TSC value [31:0] for current Generator function

TSC_GEN0_START1_0

Offset: 0x388
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:0	0x0	**MSB_VAL**: Indicates the start of TSC value [55:32] for current Generator function

TSC_GEN0_STATUS_0

Offset: 0x38c

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000,0000)

Bit	R/W	Reset	Description
6	RW	0x0	**INTERRUPT_STATUS**: Indicates current Interrupt level; Write 1 to Clear interrupt bit
5	RO	0x0	**VALUE**: Current output value
4:2	RO	0x0	**EDGE_ID**: EDGE reg currently processing
1	RO	0x0	**RUNNING**: When set, FSM is active and processing EDGE regs. Yet to encounter a STOP condition.
0	RO	0x0	**WAITING**: When set, generator is active and is waiting for TSC to reach programmed TSC_START value.

TSC_GEN0_EDGE_0

Array of seven identical EDGE registers containing fields to create user defined waves EDGE0,EDGE1....EDGE7

This is an array of eight identical register entries; the register fields below apply to each entry.
Full register list is: TSC_GEN0_EDGE_<i>, among which <i> belongs to <0..7>.
Offset: 0x398,..,0x3b4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**INTERRUPT_EN**: Enable software Interrupt after executing present EDGE reg
30	0x0	**STOP**: Generator is stopped after executing present EDGE reg Toggle ENABLE to restart FSM
29	0x0	**TOGGLE**: If enabled, toggle current value

Bit	Reset	Description
28	0x0	**LOOP**: Loop back to EDGE0 after current EDGE reg, allowing to define periodic patterns. If LOOP=0, the next EDGE register is used in a circular fashion, i.e., EDGE0, EDGE1..,EDGE0, etc. STOP has precedence over LOOP.
27:0	0x0	**OFFSET**: Offset (in TSC units) from the previous change at which to apply the actions defined by the register

TSC_GEN1_CTRL_0

Offset: 0x400
Read/Write: WO
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**INITIAL_VAL**: Start value for current Generator
0	0x0	**ENABLE**: When ENABLE goes from 0->1, the output level is unconditionally set to INITIAL_VAL. The FSM goes into the WAITING state and starts processing EDGE registers when TSC becomes equal or greater than the value programmed in TSC_START. When ENABLE is 0, the FSM is stopped in an IDLE state (WAITING and RUNNING both 0b)

TSC_GEN1_START0_0

Offset: 0x404
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LSB_VAL**: Indicates the start of TSC value [31:0] for current Generator function

TSC_GEN1_START1_0

Offset: 0x408
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:0	0x0	**MSB_VAL**: Indicates the start of TSC value [55:32] for current Generator function

TSC_GEN1_STATUS_0

Offset: 0x40c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000,0000)

Bit	R/W	Reset	Description
6	RW	0x0	**INTERRUPT_STATUS**: Indicates current Interrupt level; Write 1 to Clear interrupt bit
5	RO	0x0	**VALUE**: Current output value
4:2	RO	0x0	**EDGE_ID**: EDGE reg currently processing
1	RO	0x0	**RUNNING**: When set, FSM is active and processing EDGE regs. Yet to encounter a STOP condition.
0	RO	0x0	**WAITING**: When set, generator is active and is waiting for TSC to reach programmed TSC_START value.

TSC_GEN1_EDGE_0

Array of seven identical EDGE registers containing fields to create user defined waves EDGE0,EDGE1....EDGE7

This is an array of eight identical register entries; the register fields below apply to each entry.
Full register list is: TSC_GEN1_EDGE_<i>, among which <i> belongs to <0..7>.
Offset: 0x418,..,0x434
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**INTERRUPT_EN**: Enable software Interrupt after executing present EDGE reg
30	0x0	**STOP**: Generator is stopped after executing present EDGE reg Toggle ENABLE to restart FSM
29	0x0	**TOGGLE**: If enabled, toggle current value
28	0x0	**LOOP**: Loop back to EDGE0 after current EDGE reg, allowing to define periodic patterns. If LOOP=0, the next EDGE register is used in a circular fashion, i.e., EDGE0, EDGE1..,EDGE0, etc. STOP has precedence over LOOP.
27:0	0x0	**OFFSET**: Offset (in TSC units) from the previous change at which to apply the actions defined by the register

TSC_GEN2_CTRL_0

Offset: 0x480
Read/Write: WO
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**INITIAL_VAL**: Start value for current Generator
0	0x0	**ENABLE**: When ENABLE goes from 0->1, the output level is unconditionally set to INITIAL_VAL. The FSM goes into the WAITING state and starts processing EDGE registers when TSC becomes equal or greater than the value programmed in TSC_START. When ENABLE is 0, the FSM is stopped in an IDLE state (WAITING and RUNNING both 0b)

TSC_GEN2_START0_0

Offset: 0x484
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LSB_VAL**: Indicates the start of TSC value [31:0] for current Generator function

TSC_GEN2_START1_0

Offset: 0x488
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:0	0x0	**MSB_VAL**: Indicates the start of TSC value [55:32] for current Generator function

TSC_GEN2_STATUS_0

Offset: 0x48c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000,0000)

Bit	R/W	Reset	Description
6	RW	0x0	**INTERRUPT_STATUS**: Indicates current Interrupt level; Write 1 to Clear interrupt bit
5	RO	0x0	**VALUE**: Current output value
4:2	RO	0x0	**EDGE_ID**: EDGE reg currently processing

Bit	R/W	Reset	Description
1	RO	0x0	**RUNNING**: When set, FSM is active and processing EDGE regs. Yet to encounter a STOP condition.
0	RO	0x0	**WAITING**: When set, generator is active and is waiting for TSC to reach programmed TSC_START value.

TSC_GEN2_EDGE_0

Array of seven identical EDGE registers containing fields to create user defined waves EDGE0,EDGE1....EDGE7

This is an array of eight identical register entries; the register fields below apply to each entry.
Full register list is: TSC_GEN2_EDGE_<i>, among which <i> belongs to <0..7>.
Offset: 0x498,..,0x4b4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**INTERRUPT_EN**: Enable software Interrupt after executing present EDGE reg
30	0x0	**STOP**: Generator is stopped after executing present EDGE reg Toggle ENABLE to restart FSM
29	0x0	**TOGGLE**: If enabled, toggle current value
28	0x0	**LOOP**: Loop back to EDGE0 after current EDGE reg, allowing to define periodic patterns. If LOOP=0, the next EDGE register is used in a circular fashion, i.e., EDGE0, EDGE1..,EDGE0, etc. STOP has precedence over LOOP.
27:0	0x0	**OFFSET**: Offset (in TSC units) from the previous change at which to apply the actions defined by the register

TSC_GEN3_CTRL_0

Offset: 0x500
Read/Write: WO
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**INITIAL_VAL**: Start value for current Generator
0	0x0	**ENABLE**: When ENABLE goes from 0->1, the output level is unconditionally set to INITIAL_VAL. The FSM goes into the WAITING state and starts processing EDGE registers when TSC becomes equal or greater than the value programmed in TSC_START. When ENABLE is 0, the FSM is stopped in an IDLE state (WAITING and RUNNING both 0b)

TSC_GEN3_START0_0

Offset: 0x504
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LSB_VAL**: Indicates the start of TSC value [31:0] for current Generator function

TSC_GEN3_START1_0

Offset: 0x508
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:0	0x0	**MSB_VAL**: Indicates the start of TSC value [55:32] for current Generator function

TSC_GEN3_STATUS_0

Offset: 0x50c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000,0000)

Bit	R/W	Reset	Description
6	RW	0x0	**INTERRUPT_STATUS**: Indicates current Interrupt level; Write 1 to Clear interrupt bit
5	RO	0x0	**VALUE**: Current output value
4:2	RO	0x0	**EDGE_ID**: EDGE reg currently processing
1	RO	0x0	**RUNNING**: When set, FSM is active and processing EDGE regs. Yet to encounter a STOP condition.
0	RO	0x0	**WAITING**: When set, generator is active and is waiting for TSC to reach programmed TSC_START value.

TSC_GEN3_EDGE_0

Array of seven identical EDGE registers containing fields to create user defined waves EDGE0,EDGE1....EDGE7

This is an array of eight identical register entries; the register fields below apply to each entry.
Full register list is: TSC_GEN3_EDGE_<i>, among which <i> belongs to <0..7>.
Offset: 0x518,..,0x534
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: TSC_SCR_TSCSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**INTERRUPT_EN**: Enable software Interrupt after executing present EDGE reg
30	0x0	**STOP**: Generator is stopped after executing present EDGE reg Toggle ENABLE to restart FSM

Bit	Reset	Description
29	0x0	**TOGGLE**: If enabled, toggle current value
28	0x0	**LOOP**: Loop back to EDGE0 after current EDGE reg, allowing to define periodic patterns. If LOOP=0, the next EDGE register is used in a circular fashion, i.e., EDGE0, EDGE1..,EDGE0, etc. STOP has precedence over LOOP.
27:0	0x0	**OFFSET**: Offset (in TSC units) from the previous change at which to apply the actions defined by the register

8.3.4.7 Microsecond Counter Registers

USEC_CNTR_USECCVR_0

USECCVR, Microsecond Counter Value Register

Offset: 0x0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: USEC_CNTR_SCR_USECSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CNT**: Number of usec pulses since last system reset

USEC_CNTR_USECCCR_0

USECCCR, Microsecond Counter Configuration Register

Offset: 0x4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: USEC_CNTR_SCR_USECSCR_0
Reset: 0x000004bf (0bxxxx,xxxx,xxxx,xxxx,0000,0100,1011,1111)

Bit	Reset	Description
15:8	0x4	**Dividend**: Numerator is Dividend + 1, also known as N

Bit	Reset	Description
7:0	0xbf	**Divisor**: Denominator is Divisor + 1, default value corresponds to 38.4MHz OSC.

USEC_CNTR_USECCRSR_0

USECCRSR, Microsecond Counter Reference Selection Register

Offset: 0x10
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: USEC_CNTR_SCR_USECSCR_0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xxx1)

Bit	Reset	Description
5:4	0x0	**MBS**: STSC Bit Select, STSC[1+MBS] edge used as reference when not free running.
0	0x1	**FR**: Free running, controls the microsecond counter TSC lock behavior When 0b, the microsecond locks the usec reference pulse to a selected bit of STSC[4:1] using the adjust signal; this is the recommended mode When 1b, Usec Counter counts on each edge of OSC in free running fashion, legacy mode

USEC_CNTR_USECCFR_0

USECCFR, Microsecond Counter Freeze Register

Offset: 0x3c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: USEC_CNTR_SCR_USECSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**HDBG**: Halt-on-debug. Controls whether a Halt-on-debug signal halts the microsecond and associated divider logic 0: Microsecond counter ignores Halt-on-debug. 1: Asserted Halt-on-debug signal halts the ration divider, automatically stopping the microsecond counter and the timing reference signals. Care must be taken as timing reference signals are in the inactive state during halt.

USEC_CNTR_CLK_OVR_ON_0

Offset: 0x40
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: USEC_CNTR_SCR_USECSCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**OSC**: SLCG override bit **0** = DISABLE **1** = ENABLE

8.3.4.8 Generic Timestamping Engine (GTE) APS Registers

GTE_APS_TECTRL_0

TECTRL, timestamp engine control register

Offset: 0x0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_APS_SCR_TESCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,xx00,x000)

Bit	Reset	Description
15:8	0x0	**OCCUPANCYTHRESHOLD**: Generate an interrupt if the FIFO occupancy is equal to or larger than this limit. Not all bits physically present depending on FIFO capacity
5:4	0x0	**AUTOADVADDR**: When auto advance (AUTOADVENABLE) is enabled, advance the FIFO when reading address 0x08 + 4 * AutoAdvAddr.

2	0x0	**AUTOADVENABLE**: Enable automatic advance of the FIFO for DMA purpose. When enabled, AutoAdvAddr indicates which register advance the FIFO. When disabled the FIFO advance is via the command register to avoid read-sensitive addresses. **0** = DISABLE **1** = ENABLE
1	0x0	**INTERRUPTENABLE**: Enable generation of interrupt based on FIFO occupancy **0** = DISABLE **1** = ENABLE
0	0x0	**ENABLE**: Disable this timestamp engine **0** = DISABLE **1** = ENABLE

GTE_APS_TETSCH_0

TETSCH, timestamp engine TSC high register

Offset: 0x4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: GTE_APS_SCR_TESCR_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:0	0x0	**VAL**: TSC[55:32] for the captured event at the FIFO head, 0b if the FIFO is empty

GTE_APS_TETSCL_0

TETSCL, timestamp engine TSC low register

Offset: 0x8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: GTE_APS_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VAL**: TSC[31:0] for the captured event at the FIFO head, 0b if the FIFO is empty

GTE_APS_TESRC_0

TESRC, timestamp engine source register

Offset: 0xc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: GTE_APS_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**MODULE**: Identifies the specific timestamp engine as the source for this event. The value must be unique across the SoC. The value is a constant, i.e., it remains non zero when the FIFO is empty.
23:16	0x0	**SLICE**: Indicate the slice for the captured event at the FIFO head, 0b if the FIFO is empty. Not all bits physically present depending on number of slices present.
15:0	0x0	**TAG**: Always 0, no tag defined

GTE_APS_TECCV_0

TECCV, timestamp engine current captured value register

Offset: 0x10
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: GTE_APS_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VAL**: The value of the interrupt signals at the current capture time, 0b if the FIFO is empty.

GTE_APS_TEPCV_0

TEPCV, timestamp engine previous captured value register

Offset: 0x14
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: GTE_APS_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VAL**: The value of the interrupt signals at the previous capture time, 0b if the FIFO is empty.

GTE_APS_TEENCV_0

TEENCV, timestamp engine encode with valid register

Offset: 0x18
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: GTE_APS_SCR_TESCR_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxx0,xxx0,0000,0000,0000)

Bit	Reset	Description
31	0x0	**INVALID**: Indicates that the other field do not carry any valid information
16	0x0	**FALLING**: Direction of the change for the identified bit, 0b when all bits have been encoded
12:5	0x0	**SLICE**: Indicate the slice for the captured event at the FIFO head, 0b if the FIFO is empty, and 0 if all bits have been encoded. Not all bits physically present depending on number of slices present.
4:0	0x0	**BITINDEX**: Bit index in the slice of bit that flipped between the current and previous value

GTE_APS_TECMD_0

TECMD, timestamp engine command register

Offset: 0x1c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_APS_SCR_TESCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	0x0	**CMD**: Command to execute, reads NOP when the command has finished, 0: NOP, 1: POP - advance the FIFO, the FIFO may become empty **0** = NOP **1** = POP

GTE_APS_TESTATUS_0

TESTATUS, timestamp engine status register

Offset: 0x20
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: GTE_APS_SCR_TESCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,xxxx,xx0x)

Bit	Reset	Description
15:8	0x0	**OCCUPANCY**: Current FIFO occupancy, 0 indicates empty. Not all bits physically present depending on FIFO capacity
1	0x0	**INTERRUPT**: Current interrupt status, same as the outgoing signal, so after the interrupt enable

GTE_APS_CLK_OVR_ON_0

Offset: 0x24
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: GTE_APS_SCR_TESCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**GTE**: SLCG override bit **0** = DISABLE **1** = ENABLE

GTE_APS_SLICE0_TETEN_0

Offset: 0x40
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_APS_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of enabled events. Enable map is under control of masters allowed to read the timestamp

GTE_APS_SLICE0_TETDIS_0

Offset: 0x44
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_APS_SCR_TEDSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of disabled events Disable map is assumed to be frozen during system configuration to enforce visibility rules, i.e., taking the security model into account to hide certain signals from certain masters

GTE_APS_SLICE1_TETEN_0

Offset: 0x60
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_APS_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of enabled events. Enable map is under control of masters allowed to read the timestamp

GTE_APS_SLICE1_TETDIS_0
Offset: 0x64
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_APS_SCR_TEDSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of disabled events Disable map is assumed to be frozen during system configuration to enforce visibility rules, i.e., taking the security model into account to hide certain signals from certain masters

8.3.4.9 Generic Timestamping Engine (GTE) GPIO Registers

GTE_GPIO_TECTRL_0

TECTRL, timestamp engine control register

Offset: 0x0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_GPIO_SCR_TESCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,xx00,x000)

Bit	Reset	Description
15:8	0x0	**OCCUPANCYTHRESHOLD**: Generate an interrupt if the FIFO occupancy is equal to or larger than this limit. Not all bits physically present depending on FIFO capacity

5:4	0x0	**AUTOADVADDR**: When auto advance (AUTOADVENABLE) is enabled, advance the FIFO when reading address 0x08 + 4 * AutoAdvAddr.
2	0x0	**AUTOADVENABLE**: Enable automatic advance of the FIFO for DMA purpose. When enabled, AutoAdvAddr indicates which register advance the FIFO. When disabled the FIFO advance is via the command register to avoid read-sensitive addresses. **0** = DISABLE **1** = ENABLE
1	0x0	**INTERRUPTENABLE**: Enable generation of interrupt based on FIFO occupancy **0** = DISABLE **1** = ENABLE
0	0x0	**ENABLE**: Disable this timestamp engine **0** = DISABLE **1** = ENABLE

GTE_GPIO_TETSCH_0

TETSCH, timestamp engine TSC high register

Offset: 0x4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: GTE_GPIO_SCR_TESCR_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:0	0x0	**VAL**: TSC[55:32] for the captured event at the FIFO head, 0b if the FIFO is empty

GTE_GPIO_TETSCL_0

TETSCL, timestamp engine TSC low register

Offset: 0x8
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: GTE_GPIO_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VAL**: TSC[31:0] for the captured event at the FIFO head, 0b if the FIFO is empty

GTE_GPIO_TESRC_0

TESRC, timestamp engine source register

Offset: 0xc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: GTE_GPIO_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**MODULE**: Identifies the specific timestamp engine as the source for this event. The value must be unique across the SoC. The value is a constant, i.e., it remains non zero when the FIFO is empty.
23:16	0x0	**SLICE**: Indicate the slice for the captured event at the FIFO head, 0b if the FIFO is empty. Not all bits physically present depending on number of slices present.
15:0	0x0	**TAG**: Always 0, no tag defined

GTE_GPIO_TECCV_0

TECCV, timestamp engine current captured value register

Offset: 0x10
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: GTE_GPIO_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description

31:0	0x0	**VAL**: The value of the interrupt signals at the current capture time, 0b if the FIFO is empty.

GTE_GPIO_TEPCV_0

TEPCV, timestamp engine previous captured value register

Offset: 0x14
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: GTE_GPIO_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VAL**: The value of the interrupt signals at the previous capture time, 0b if the FIFO is empty.

GTE_GPIO_TEENCV_0

TEENCV, timestamp engine encode with valid register

Offset: 0x18
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: GTE_GPIO_SCR_TESCR_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxx0,xxx0,0000,0000,0000)

Bit	Reset	Description
31	0x0	**INVALID**: Indicates that the other field do not carry any valid information
16	0x0	**FALLING**: Direction of the change for the identified bit, 0b when all bits have been encoded
12:5	0x0	**SLICE**: Indicate the slice for the captured event at the FIFO head, 0b if the FIFO is empty, and 0 if all bits have been encoded. Not all bits physically present depending on number of slices present.
4:0	0x0	**BITINDEX**: Bit index in the slice of bit that flipped between the current and previous value

GTE_GPIO_TECMD_0

TECMD, timestamp engine command register

Offset: 0x1c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_GPIO_SCR_TESCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	0x0	**CMD**: Command to execute, reads NOP when the command has finished, 0: NOP, 1: POP - advance the FIFO, the FIFO may become empty **0** = NOP **1** = POP

GTE_GPIO_TESTATUS_0

TESTATUS, timestamp engine status register

Offset: 0x20
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: GTE_GPIO_SCR_TESCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,xxxx,xx0x)

Bit	Reset	Description
15:8	0x0	**OCCUPANCY**: Current FIFO occupancy, 0 indicates empty. Not all bits physically present depending on FIFO capacity
1	0x0	**INTERRUPT**: Current interrupt status, same as the outgoing signal, so after the interrupt enable

GTE_GPIO_CLK_OVR_ON_0

Offset: 0x24
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: GTE_GPIO_SCR_TESCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**GTE**: SLCG override bit **0** = DISABLE **1** = ENABLE

GTE_GPIO_SLICE0_TETEN_0

Offset: 0x40
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_GPIO_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of enabled events. Enable map is under control of masters allowed to read the timestamp

GTE_GPIO_SLICE0_TETDIS_0

Offset: 0x44
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_GPIO_SCR_TEDSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of disabled events Disable map is assumed to be frozen during system configuration to enforce visibility rules, i.e., taking the security model into account to hide certain signals from certain masters

GTE_GPIO_SLICE1_TETEN_0

Offset: 0x60
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_GPIO_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of enabled events. Enable map is under control of masters allowed to read the timestamp

GTE_GPIO_SLICE1_TETDIS_0

Offset: 0x64
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_GPIO_SCR_TEDSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of disabled events Disable map is assumed to be frozen during system configuration to enforce visibility rules, i.e., taking the security model into account to hide certain signals from certain masters

GTE_GPIO_SLICE2_TETEN_0

Offset: 0x80
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_GPIO_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of enabled events. Enable map is under control of masters allowed to read the timestamp

GTE_GPIO_SLICE2_TETDIS_0
Offset: 0x84
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_GPIO_SCR_TEDSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of disabled events Disable map is assumed to be frozen during system configuration to enforce visibility rules, i.e., taking the security model into account to hide certain signals from certain masters

8.3.4.10 Generic Timestamping Engine (GTE) Interrupt Controller Registers

GTE_INTR_CTLR_TECTRL_0

TECTRL, timestamp engine control register

Offset: 0x0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TESCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,xx00,x000)

Bit	Reset	Description
15:8	0x0	**OCCUPANCYTHRESHOLD**: Generate an interrupt if the FIFO occupancy is equal to or larger than this limit. Not all bits physically present depending on FIFO capacity
5:4	0x0	**AUTOADVADDR**: When auto advance (AUTOADVENABLE) is enabled, advance the FIFO when reading address 0x08 + 4 * AutoAdvAddr.
2	0x0	**AUTOADVENABLE**: Enable automatic advance of the FIFO for DMA purpose. When enabled, AutoAdvAddr indicates which register advance the FIFO. When disabled the FIFO advance is via the command register to avoid read-sensitive addresses. **0** = DISABLE **1** = ENABLE

1	0x0	**INTERRUPTENABLE**: Enable generation of interrupt based on FIFO occupancy **0** = DISABLE **1** = ENABLE
0	0x0	**ENABLE**: Disable this timestamp engine **0** = DISABLE **1** = ENABLE

GTE_INTR_CTLR_TETSCH_0

TETSCH, timestamp engine TSC high register

Offset: 0x4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TESCR_0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:0	0x0	**VAL**: TSC[55:32] for the captured event at the FIFO head, 0b if the FIFO is empty

GTE_INTR_CTLR_TETSCL_0

TETSCL, timestamp engine TSC low register

Offset: 0x8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VAL**: TSC[31:0] for the captured event at the FIFO head, 0b if the FIFO is empty

GTE_INTR_CTLR_TESRC_0

TESRC, timestamp engine source register

Offset: 0xc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**MODULE**: Identifies the specific timestamp engine as the source for this event. The value must be unique across the SoC. The value is a constant, i.e., it remains non zero when the FIFO is empty.
23:16	0x0	**SLICE**: Indicate the slice for the captured event at the FIFO head, 0b if the FIFO is empty. Not all bits physically present depending on number of slices present.
15:0	0x0	**TAG**: Always 0, no tag defined

GTE_INTR_CTLR_TECCV_0

TECCV, timestamp engine current captured value register

Offset: 0x10
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VAL**: The value of the interrupt signals at the current capture time, 0b if the FIFO is empty.

GTE_INTR_CTLR_TEPCV_0

TEPCV, timestamp engine previous captured value register

Offset: 0x14
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VAL**: The value of the interrupt signals at the previous capture time, 0b if the FIFO is empty.

GTE_INTR_CTLR_TEENCV_0

TEENCV, timestamp engine encode with valid register

Offset: 0x18
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TESCR_0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxx0,xxx0,0000,0000,0000)

Bit	Reset	Description
31	0x0	**INVALID**: Indicates that the other field do not carry any valid information
16	0x0	**FALLING**: Direction of the change for the identified bit, 0b when all bits have been encoded
12:5	0x0	**SLICE**: Indicate the slice for the captured event at the FIFO head, 0b if the FIFO is empty, and 0 if all bits have been encoded. Not all bits physically present depending on number of slices present.
4:0	0x0	**BITINDEX**: Bit index in the slice of bit that flipped between the current and previous value

GTE_INTR_CTLR_TECMD_0

TECMD, timestamp engine command register

Offset: 0x1c
Read/Write: R/W

Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TESCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	0x0	**CMD**: Command to execute, reads NOP when the command has finished, 0: NOP, 1: POP - advance the FIFO, the FIFO may become empty **0** = NOP **1** = POP

GTE_INTR_CTLR_TESTATUS_0

TESTATUS, timestamp engine status register

Offset: 0x20
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TESCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,xxxx,xx0x)

Bit	Reset	Description
15:8	0x0	**OCCUPANCY**: Current FIFO occupancy, 0 indicates empty. Not all bits physically present depending on FIFO capacity
1	0x0	**INTERRUPT**: Current interrupt status, same as the outgoing signal, so after the interrupt enable

GTE_INTR_CTLR_CLK_OVR_ON_0

Offset: 0x24
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TESCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description

0	0x0	**GTE**: SLCG override bit **0** = DISABLE **1** = ENABLE

GTE_INTR_CTLR_SLICE0_TETEN_0

Offset: 0x40
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of enabled events. Enable map is under control of masters allowed to read the timestamp

GTE_INTR_CTLR_SLICE0_TETDIS_0

Offset: 0x44
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TEDSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of disabled events Disable map is assumed to be frozen during system configuration to enforce visibility rules, i.e., taking the security model into account to hide certain signals from certain masters

GTE_INTR_CTLR_SLICE1_TETEN_0

Offset: 0x60
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: GTE_INTR_CTLR_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of enabled events. Enable map is under control of masters allowed to read the timestamp

GTE_INTR_CTLR_SLICE1_TETDIS_0

Offset: 0x64
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TEDSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of disabled events Disable map is assumed to be frozen during system configuration to enforce visibility rules, i.e., taking the security model into account to hide certain signals from certain masters

GTE_INTR_CTLR_SLICE2_TETEN_0

Offset: 0x80
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of enabled events. Enable map is under control of masters allowed to read the timestamp

GTE_INTR_CTLR_SLICE2_TETDIS_0

Offset: 0x84
Read/Write: R/W

Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TEDSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of disabled events Disable map is assumed to be frozen during system configuration to enforce visibility rules, i.e., taking the security model into account to hide certain signals from certain masters

GTE_INTR_CTLR_SLICE3_TETEN_0

Offset: 0xa0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of enabled events. Enable map is under control of masters allowed to read the timestamp

GTE_INTR_CTLR_SLICE3_TETDIS_0

Offset: 0xa4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TEDSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of disabled events Disable map is assumed to be frozen during system configuration to enforce visibility rules, i.e., taking the security model into account to hide certain signals from certain masters

GTE_INTR_CTLR_SLICE4_TETEN_0

Offset: 0xc0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of enabled events. Enable map is under control of masters allowed to read the timestamp

GTE_INTR_CTLR_SLICE4_TETDIS_0

Offset: 0xc4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TEDSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of disabled events Disable map is assumed to be frozen during system configuration to enforce visibility rules, i.e., taking the security model into account to hide certain signals from certain masters

GTE_INTR_CTLR_SLICE5_TETEN_0

Offset: 0xe0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of enabled events. Enable map is under control of masters allowed to read the timestamp

GTE_INTR_CTLR_SLICE5_TETDIS_0

Offset: 0xe4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TEDSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of disabled events Disable map is assumed to be frozen during system configuration to enforce visibility rules, i.e., taking the security model into account to hide certain signals from certain masters

GTE_INTR_CTLR_SLICE6_TETEN_0

Offset: 0x100
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of enabled events. Enable map is under control of masters allowed to read the timestamp

GTE_INTR_CTLR_SLICE6_TETDIS_0

Offset: 0x104
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TEDSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of disabled events Disable map is assumed to be frozen during system configuration to enforce visibility rules, i.e., taking the security model into account to hide certain signals from certain masters

GTE_INTR_CTLR_SLICE7_TETEN_0

Offset: 0x120
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of enabled events. Enable map is under control of masters allowed to read the timestamp

GTE_INTR_CTLR_SLICE7_TETDIS_0

Offset: 0x124
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TEDSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of disabled events Disable map is assumed to be frozen during system configuration to enforce visibility rules, i.e., taking the security model into account to hide certain signals from certain masters

GTE_INTR_CTLR_SLICE8_TETEN_0

Offset: 0x140
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of enabled events. Enable map is under control of masters allowed to read the timestamp

GTE_INTR_CTLR_SLICE8_TETDIS_0

Offset: 0x144
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TEDSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of disabled events Disable map is assumed to be frozen during system configuration to enforce visibility rules, i.e., taking the security model into account to hide certain signals from certain masters

GTE_INTR_CTLR_SLICE9_TETEN_0

Offset: 0x160
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of enabled events. Enable map is under control of masters allowed to read the timestamp

GTE_INTR_CTLR_SLICE9_TETDIS_0

Offset: 0x164

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TEDSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of disabled events Disable map is assumed to be frozen during system configuration to enforce visibility rules, i.e., taking the security model into account to hide certain signals from certain masters

GTE_INTR_CTLR_SLICE10_TETEN_0

Offset: 0x180
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of enabled events. Enable map is under control of masters allowed to read the timestamp

GTE_INTR_CTLR_SLICE10_TETDIS_0

Offset: 0x184
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TEDSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of disabled events Disable map is assumed to be frozen during system configuration to enforce visibility rules, i.e., taking the security model into account to hide certain signals from certain masters

GTE_INTR_CTLR_SLICE11_TETEN_0

Offset: 0x1a0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of enabled events. Enable map is under control of masters allowed to read the timestamp

GTE_INTR_CTLR_SLICE11_TETDIS_0

Offset: 0x1a4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TEDSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of disabled events Disable map is assumed to be frozen during system configuration to enforce visibility rules, i.e., taking the security model into account to hide certain signals from certain masters

GTE_INTR_CTLR_SLICE12_TETEN_0

Offset: 0x1c0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of enabled events. Enable map is under control of masters allowed to read the timestamp

GTE_INTR_CTLR_SLICE12_TETDIS_0

Offset: 0x1c4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TEDSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of disabled events Disable map is assumed to be frozen during system configuration to enforce visibility rules, i.e., taking the security model into account to hide certain signals from certain masters

GTE_INTR_CTLR_SLICE13_TETEN_0

Offset: 0x1e0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of enabled events. Enable map is under control of masters allowed to read the timestamp

GTE_INTR_CTLR_SLICE13_TETDIS_0

Offset: 0x1e4
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: GTE_INTR_CTLR_SCR_TEDSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of disabled events Disable map is assumed to be frozen during system configuration to enforce visibility rules, i.e., taking the security model into account to hide certain signals from certain masters

GTE_INTR_CTLR_SLICE14_TETEN_0

Offset: 0x200
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of enabled events. Enable map is under control of masters allowed to read the timestamp

GTE_INTR_CTLR_SLICE14_TETDIS_0

Offset: 0x204
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TEDSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of disabled events Disable map is assumed to be frozen during system configuration to enforce visibility rules, i.e., taking the security model into account to hide certain signals from certain masters

GTE_INTR_CTLR_SLICE15_TETEN_0

Offset: 0x220
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of enabled events. Enable map is under control of masters allowed to read the timestamp

GTE_INTR_CTLR_SLICE15_TETDIS_0

Offset: 0x224
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TEDSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of disabled events Disable map is assumed to be frozen during system configuration to enforce visibility rules, i.e., taking the security model into account to hide certain signals from certain masters

GTE_INTR_CTLR_SLICE16_TETEN_0

Offset: 0x240
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TESCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of enabled events. Enable map is under control of masters allowed to read the timestamp

GTE_INTR_CTLR_SLICE16_TETDIS_0
Offset: 0x244
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GTE_INTR_CTLR_SCR_TEDSCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MAP**: Represents the set of disabled events Disable map is assumed to be frozen during system configuration to enforce visibility rules, i.e., taking the security model into account to hide certain signals from certain masters

8.4 Multi-Purpose I/O Pins and Pin Multiplexing (PinMux)

8.4.1 Overview

This chapter discusses the features and capabilities of the Multi-purpose I/O (MPIO) pads, and how to configure their multiplexing so that the same group of MPIO's can serve multiple functions for different applications, hence Pin Multiplexing (PinMux). These are part of the pad-ring, which is a compilation of pads, custom cells, and related test logic along the periphery of the die. The following key words are used in the description of the PinMux:

- **Pad -** A Pad is typically a circuit logic block with single-ended signaling circuitry to interface with the external world.
- **Brick -** A Brick is typically a high-speed I/O circuit block with differential signaling circuitry to interface with the external world, such as the PLL pins. A Brick can also be a collection of single-end driver/receiver pads, such as those used for the EMMC interface.
- **Pin -** A Pin is a BGA ball in the SoC package that is the Pad's/brick's connection to the external world. It is synonymous to and used interchangeably with the term "I/O" in the descriptions of this chapter.

Scope of This Chapter:

- MPIO pad types
- MPIO pad controls
- Pad control units
- MPIO pad features low power, back-drive

- Low-power features of MPIO pads
- PinMux scheme
- Generic programming considerations of pad controls
- Default pin states and pull-up/down selection
- Misc system arch aspects applicable to MPIO pads
- Registers for PinMux selection, I/O capabilities including electrical control settings.

PinMux allows a single MPIO pad or a group of MPIO pads to be used by multiple low-speed I/O controllers, including the General-Purpose Input/Output pad (GPIO) controller. Depending on the platform use case, the PinMux logic can be programmed/configured to allow any specific controller the exclusive use of the assigned MPIOs. The configuration is static and determined by Orin system platform requirements. With Pin Multiplexing, the total number of I/Os required for Orin can thus be optimized.

The PinMux scheme includes multiple groups of MPIO pads, each of which is shared by a number of low-speed I/O controllers, including the GPIO controller. A Pad control unit with configuration registers is associated with each of these MPIO pads groups to achieve the following:

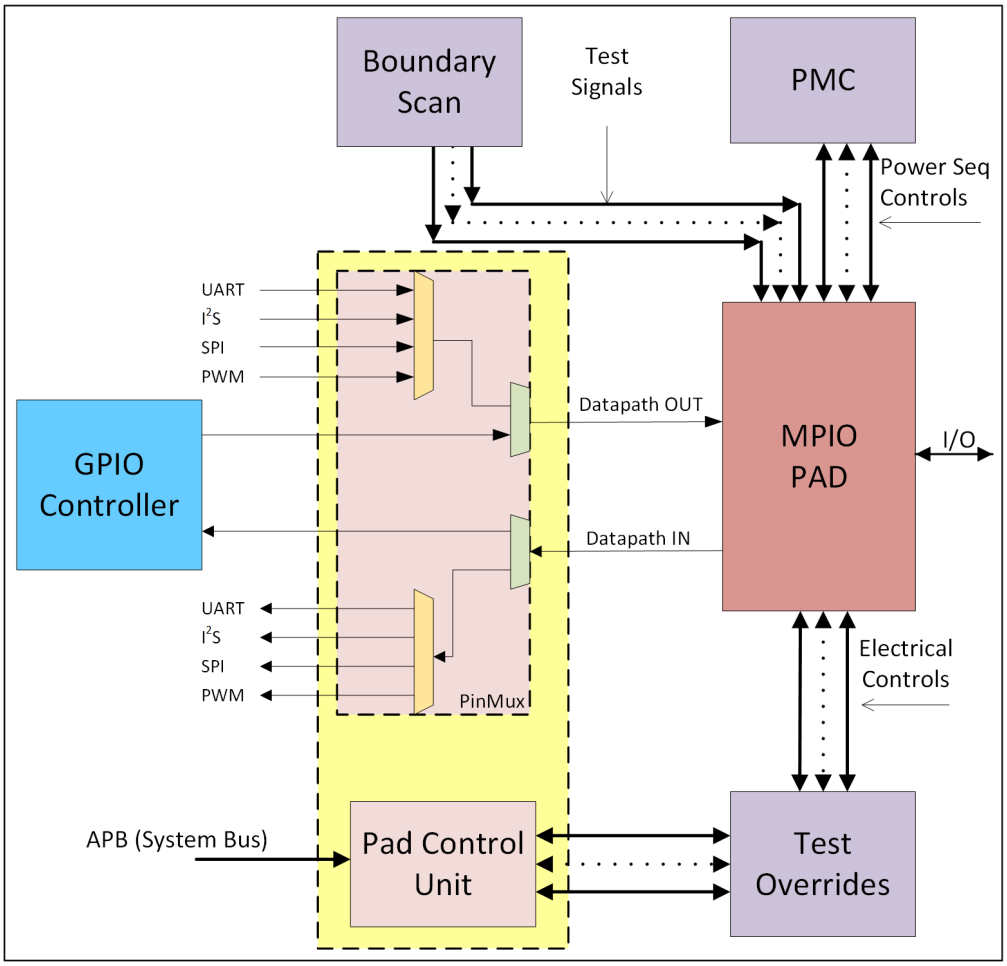
- PinMux selection (selecting specific controller for its exclusive use of the group of MPIO pads)
- I/O capabilities of the MPIO pads group
- Electrical settings of the MPIO pads group

Notes:

1. Details of the pad/brick circuit design are not discussed except for their usage and features.
2. Refer to the corresponding chapters of the various Low-speed Interfaces which are connected to the PinMux logic for specific programming sequences.
3. High-Speed I/O (HSIO) interfaces such as PCIe, USB3.x, SATA, and UFS do not use MPIO pads, do not support PinMuxing, and are not discussed in this chapter. Refer to the HSIO Cluster chapter and related sections for features and programming guidelines.
4. PinMux details are provided by NVIDIA in the form of a spreadsheet simply referred to just as *Pinmux*. Please refer to the NVIDIA support web-sites, or ask your NVIDIA representative for details.

The following diagram illustrates how PinMux multiplexes different Orin I/O signals (Such as UART, I^2S, SPI, and PWM as an example) to use one common MPIO pad (group).

Figure 8.32 PinMux Illustration



8.4.1.1 Standard and Compatibility

The MPIO pads used in Pin Multiplexing support various low-speed interfaces as well as the control sideband signals for high-speed interfaces. Each of the MPIO pad types is designed to comply with the interface standards including their electrical specifications. The table below lists the MPIO pad types and the interfaces they support.

Table 8.31 MPIO Pad Types and Supported Interfaces

MPIO Pad Type	MPIO Pad Name	Supported Interfaces/Devices
ST	BDPGLP	UART, SPI, I^2S, Various peripheral sideband GPIOs.
OD	BDPGLPHVIN	I^2C, PCIe sidebands, SPI, UART, DP-AUX sidebands, HDMI CEC, USB VBUS controls.

MPIO Pad Type	MPIO Pad Name	Supported Interfaces/Devices
CZ	BDSDMEM	SD, SDIO, EMMC, I^2S, CAN.
LV_CZ	BDSDMEMLV	QSPI, RGMII, UFS sidebands.
JT_RST	BILPRST	JTAG.

For details of how these interfaces are implemented in Orin, refer to the relevant LSIO and HSIO chapters in this TRM.

8.4.1.2 Glossary

Note that different names may end up in the same acronym when the entire Orin TRM is put in perspective. The acronyms listed here are within the context of the Multi-Purpose I/Os and Pin Multiplexing chapter.

Term	Definition
AON I/O Rails	I/O rails VDDIO_SYS, VDDIO_AO, and VDDIO_AO_HV serving as AON cluster logic are referred as AON I/O rails
CV	Computer Vision
CZ	Controlled Impedance Pad
DD	Dual Driver CMOS Pad
DFT	Design For Test
DPD	Deep Power Down (Pad Low-Power Mode)
GPIO	General Purpose I/O
HPD	Hot Plug Detect
HSIO	High-Speed I/O
IST	In-System Test
LSIO	Low-Speed I/O
LSM	Latent fault mechanism
LV	Low-Voltage Pad
MB	Micro-Boot
MPIO	Multi Purpose I/O
OBS	Observation Bus
OD	Open Drain

Term	Definition
PAD Group	Collection of pads sharing the same I/O supply/power ball
Padlet	Logical group of pads sharing I/O voltage along with pad control registers, PinMux logic, test logic such as - boundary scan, etc.
PD	Pull-Down
PMIC	Power Management IC (a board-level component primarily with the voltage regulation functions)
PU	Pull-Up
PWM	Pulse Width Modulation
SFIO	Special Functional I/O (the specific I/O meant for some functional interface connected with hardware controller unit)
ST	Standard CMOS Pad
TMC	Test Mode Controller (spreading across multiple units with its overrides driven from the DFT logic) miniTMC is the Test Mode Controller part of the DFT logic.
VR	Voltage Regulator

8.4.1.3 Relevant Chapters in the TRM

- Address Map
- Controller Area Network (CAN)
- Clock Controller and Reset (CAR)
- Functional safety Island (FSI)
- GPIO Controllers
- Hardware Safety Manager (HSM)
- High-Speed I/O Cluster (HSIO)
- I^2C Controller
- MIPI M-PHY (MPHY)
- Memory Sub-System (MSS)
- PCIe Complex (PCIe)
- Power Management Controller (PMC)
- Platform Security Controller (PSC)
- Pulse Width Modulation (PWM)
- Quad Serial Peripheral Interface (QSPI)
- Serial Peripheral Interface (SPI)
- SD and MMC Controller (SDMMC)

- Universal Asynchronous Receiver/Transmitter (UART)
- Universal Flash Storage Complex (UFS)
- USB Complex (USB)

8.4.1.4 MPIO Pad Features

Each MPIO pad consists of:

- An output driver with
  - Tri-state capability,
  - Drive strength controls,
  - Push-Pull mode, Open-Drain mode, or both.
- An input receiver with
  - Schmitt mode, CMOS mode, or both.
- A weak pull-up and a weak pull-down.

The Orin SoC has five types of MPIO pads as summarized in the following table.

Table 8.32 MPIO Pad Types

Pad Name (Type)	IO Swing	Input Buffer	Output Buffer	Nominal Pull Strength	Slew Rate Control	Drive Strength Control
BDPGLP_* (ST [1])	1.8 V	Schmitt (300 mV) & CMOS	Push-Pull & Open Drain (emulation mode)	50 KΩ +/- 15%	No	5 bits (up and down)
BDPGLPHVIN_* (OD [2])	1.8 V	Schmitt (200/190 mV) [5] & CMOS	Push-Pull & Open Drain	50 KΩ +/- 15%	Yes	5 bits (up and down)
BDSDMEM_* (CZ [3])	1.8/3.3 V	Schmitt & CMOS	Push-Pull & Open Drain (emulation mode)	20 KΩ +/- 35%	2 bits (up and down)	7 bits (up and down)
BDSDMEMLV_* (LV_CZ [4])	1.2/1.8 V	Schmitt & CMOS	Push-Pull	18 KΩ +/- 30%	2 bits (up and down)	5 bits (up and down)
BILPRST_* (ST type input only)	1.8 V	Schmitt & CMOS	Push-Pull & Open Drain (emulation mode)	50 KΩ +/- 15%	No	No

NOTES:

[1] The **ST** (Standard) MPIO pad is the most common pad on the chip. It is used for typical General Purpose I/Os (GPIOs) and various other slow-speed (< 100 MHz) interfaces.

[2] The **OD** (Open-Drain) MPIO pad is similar to the ST pad with the additional provision of tolerating I/O voltage level up to 3.3 V when the pad's output driver is set to Open-Drain mode for Interfaces like I^2C, USB VBUS, PCIe side bands, HDMI CEC, and Display HPD signals. Special power-sequencing steps must be taken when using this feature to ensure that the pad operates without any damage or reliability concerns.

[3] The **CZ** (Controlled output Impedance) MPIO pad is optimized for use in applications requiring tightly controlled output impedance for high I/O frequencies (100 ~ 300 MHz). It is similar to the ST pad with the additional Dual-Driver (1.8 V or 3.3 V I/O swing) support and changes in drive strength circuitry of the weak pull-up/down.

[4] The **LV_CZ** (Low Voltage Controlled Impedance) pad is a variation of the CZ pad with dual voltage selection between 1.8 V or 1.2 V I/O swing.

[5] (E_IO_HV == 0), 200 mV & (E_IO_HV == 1), 190 mV. (See the corresponding PinMux registers.)

8.4.1.4.1 Special Purpose Pad Types

Beyond pad types covered so far, we have some special purpose pads used for targeted purpose. Note that these special purpose MPIO pads are different from PHYs used for high-speed differential interfaces like USB, Display, etc.

Table 8.33 Special Purpose Pad Types

Pad	I/O Rail Voltage	Functional Name	Brief Description	Pad Name
CZ_COMP	1.8/3.3 V	SDMMC Calibration Pads	Used for generating PVT compensated impedance calibration code for the SDMEM pads (typically used for interfacing SDMMC devices). It is connected to GND external reference resistor of 50 Ω with ± 1% variation. Circuitry within the Orin SoC continuously matches the output impedance of the CZ pad to the on-board pull-up resistors attached to the **CZ_COMP** pad.	BDSDMEMCOMP_*
LV_CZ_COMP	1.8 V	eMMC Calibration Pads	Same as MEM_COMP but can operated at lower I/O voltage. Circuitry within the Orin SoC continuously matches the output impedance of the LV-CZ pad to the on-board pull-up resistors attached to the **LV_CZ_COMP** pad.	BSDMEMLVCOMP_*

Pad	I/O Rail Voltage	Functional Name	Brief Description	Pad Name
JT_RST	1.8 V	JTAG and Reset Pads (Input Only)	Used for JTAG (except JTAG TDO) and Reset interface. Has a special mode of operation called VDD only mode where in the pad becomes operational without having the I/O rail up. This mode of operation is needed for Reset pads which must propagate the Reset before I/O rails are up. This is achieved by Logic0 tie-off to RCVR_SEL pin of this pad. Beyond SYS_RESET_N pin, JTAG pin, JTAG mode selection pins also follow similar usage with tie-off to RCVR_SEL pad input.	BILPRST_*

8.4.1.4.2 MPIO Pad Variations

Within each standard MPIO pad types there are variations that are mainly associated with the way how pads behave during DPD mode and availability some specific features in DPD. There are three distinct types of variations in each type. Based on the functional and power choices appropriate pad type shall be chosen for each pin.

Table 8.34 MPIO Pad Variations

S.No	Input Receiver in DPD	Output Values	PU/PD Availability During DPD [1]	Pad Type Extension	Usage
1	ENABLED	DYNAMIC	YES	VDP1P1P1 **[2]**	Used for pins that need to be active when core supply (VDD_SOC == OFF).
2	ENABLED	STATIC	YES	VDVXP1P1P1	Used for pins that need to be active during chip DPD power states. Since the input receive is enabled, can be used for pins used Wake events. Lesser leakage power than *VDP1P1P1 type pads since only input receive logic is active.
3	DISABLED	STATIC	YES	VXVDP1P1P1 **[3]**	Since receiver is DISABLED can't be used for signals acting as Wake events. Consumes less leakage power of all pad variants as even input receiver circuits are also in disabled state.
4	N/A	N/A	N/A	R90	Pad physical orientation is North-South, default pad type orientation is West-East.

NOTES:

[1] PU is not available in DPD by nature of the pad.

[2] This variation is available only in BDPGLP_*, BDPGLPHVIN_*, BILPRST_* pads. These pads do have the DPD latching circuit like the VDVXP1P1P1 pad type and support entering in the DPD (Deep Power Down) mode with SEL_DPD/E_DPD sequencing. This mechanism (in lieu of removing the DPD latching and circuitry) is chosen for the flexibility to park in a state during DPD. Hence, the VDP1P1P1 and VDVXP1P1P1 pad are the same structurally except for their power domains. The BILPRST_* pad does not support the DPD mode or Low-Power mode. (In Low-Power mode, the pad drives output state based on pad controls latched before entering in the Low-Power mode.).

[3] The pad type BDSDMEM_* and BDSDMEMLV_ deviates from the general MPIO pad type variation. These types of pads have single VXVD pad type with wake capability selection through an additional pin.

When additional input pin “WAKE_PAD” is tied to 1, it behaves like the VDVX pad with input receiver enabled for wake functionality.

When additional input pin “WAKE_PAD” is tied to 0, it behaves like th VXVD pad with receiver disabled.

8.4.1.4.3 MPIO Pad Low-Power Modes and Features

The following table summarizes the different power modes of the MPIO pads.

Table 8.35 MPIO Pad Low-Power Modes and Features

Mode	Rail Status and DPD Control Setting	Description	Usage	Remarks
OFF	I/O Rail (OFF) VDD_RTC (OFF) VAUX_CORE (OFF) Control Inputs (Don’t-Care)	Pad circuits are fully OFF and non-functional and consuming zero power.	System OFF state	
STANDBY (NO RETENTION)	I/O Rail (OFF) VDD_RTC (ON) VAUX_CORE (OFF) (E_DPD = 1) (SEL_DPD = 1)	Pads circuits are OFF except for the VDD_RTC portion of the circuit. Consumes minimal leakage in the always-on [AO] portion of the circuit operating in VDD_RTC.	SC7 State	Specific Interface’s I/O rail can be turned OFF and no I/O state is retained. Since the core rail is OFF, E_DPD and SEL_DPD should be at Logic1.
STANDBY	I/O Rail (ON) VDD_RTC (ON) VAUX_CORE (OFF) (E_DPD = 1) (SEL_DPD = 1)	Pads are holding the I/O state but core rail is OFF and hence consuming minimal power.	SC7 State	I/O states are retained through holding latches inside the pads

Mode	Rail Status and DPD Control Setting	Description	Usage	Remarks
LOGICAL DPD	I/O Rail (ON) VDD_RTC (ON) VAUX_CORE (ON) (E_DPD = 0) (SEL_DPD = 1)	Pads are holding the I/O state but core rail is not OFF and hence consuming normal power.	Transitory state and I/O values are retained.	No power benefit in this mode since core rail is ON. However, software can possibly go through this state as and when the interface is becoming IDLE to reach the STANDBY state It happens with software sequences for power saving to DPD state.
LOGICAL DPD (I/O OFF)	I/O Rail (OFF) VDD_RTC (ON) VAUX_CORE (ON) (E_DPD = 0) (SEL_DPD = 1)	Pads are not holding the I/O state but core rail is not OFF and hence consuming normal power.	Transitory state and I/O Values are not retained.	Used by the software to transit from STANDBY-NO RETENTION to Functional Status Inactive state.
INACTIVE	I/O Rail (OFF) VDD_RTC (ON) VAUX_CORE (ON) (E_DPD = 0) (SEL_DPD = 0)	Pads are functional from Core side but I/O rails are kept OFF	Transitory state needed to for transitioning to Low-Power states.	This happens when I/O rails are powering up time windows during system active-LOW power states.
FUNCTIONAL (ACTIVE)	I/O Rail (ON) VDD_RTC (ON) VAUX_CORE (ON) (E_DPD = 0) (SEL_DPD = 0)	Normal mode and pads consume both Static and Dynamic power	Normal functional usage	

8.4.1.4.4 MPIO Pad Electrical Control Features

General purpose pads support features to cater to the electrical requirements of multiple slow speed interfaces and GPIO pins. The following list provides the features common to all MPIO pads. A further complete list of electrical features by pad type is covered in the Electrical Controls section.

- Weak Pull-up: I/O driver pulls up the line irrespective of pull-up resistor on board.
  - Maps to PUPD bit field in pad control register.
- Weak Pull-down: I/O driver pulls down the line irrespective of pull-up resister on board.
  - Maps to PUPD bit field in pad control register.
- Programmable Output pull-up/pull-down drive strength: Enables to program different drive strengths required by different external bridges, codecs, and such devices in the system.
  - Maps to CFG_CAL_DRVUP/DN fields in pad control register.
- Schmitt receiver: Enables Schmitt receiver in input receiver circuit. Enabling Schmitt provides better noise margin characteristics for the input and depending on driver's logic threshold levels this can be enabled.
  - Maps to E_SCHMT bit field in pad control register.

- Input receiver: Controls whether the input path (receiver portion of I/O driver) is enabled or not. Generally disabled for most of the pads except those needed for boot.
  - Maps to E_INPUT bit field in pad control register.
- Pad output state: Top Level override to float Pad Output irrespective of the actual PinMux Control. Useful to get the correct default values and while switching PinMux to GPIO mode of operation (though such switching is not recommended in general).
  - Maps to TRISTATE bit field in pad control register.

TRISTATE = 0x0:

Pad I/O driver is DRIVEN (Some of the critical pads, typically PMIC interface pads, clock pads, etc. The Default status are driven to ensure correct values are driven to the PMIC or external device).

TRISTATE = 0x1:

Pad I/O drive is FLOATED (actual value on the pad output depends on the Weak Pull-up/Pull-down control in the respective PUPD fields.

Boot Loader software enables this field to driven state before starts using the pad in output mode. Same applies to peripheral interface drivers which shall configure pad control fields during the interface initialization. Other than the PMIC Interface related pads, The Power-On Reset default is FLOATED so that every unused pin is in the OUTPUT FLOATED/Weak-PU/PD state.

8.4.1.4.5 Electrical Controls

The following table summarizes the electrical controls of MPIO pads. All these electrical controls are configurable through pad control registers. For more details of electrical controls or more pad types, refer to pad control registers.

Table 8.36 BDSDMEM Electrical Controls (BDSDMEM Variants)

Signal Name	In/ Out	Power Domain [1]	Description
E_INPUT	In	VDD_SOC/VDD_RTC	Enables receiver path I/O → ZI/TZI.
E_SCHMT	In	VDD_SOC/VDD_RTC	Enables Schmitt trigger in the receiver.
E_LPBK	In	VDD_SOC/VDD_RTC	Enables loopback mode A → ZI/TZI.
DRV_TYPE_[0]	In	VDD_SOC/VDD_RTC	1x/2x I/O driver selection.
DRV_TYPE_[1]	In	VDD_SOC/VDD_RTC	Drive codes selection: COMP pad driven codes or pad default codes.
DRVUP_DEF_[1:0]	In	VDD_SOC/VDD_RTC	Back-up default driver output pull-up control code. Applied Drive up/down codes (0, 1, DRVUP_DEF[1], DRVUP_DEF[0], 0).

Signal Name	In/ Out	Power Domain [1]	Description
DRVDN_DEF_[1:0]	In	VDD_SOC/VDD_RTC	Back-up default driver output pull-down control code. Applied Drive up/down codes (0, 1, DRVDN_DEF[1], DRVDN_DEF[0], 0).
E_PULLU	In	VDD_SOC/VDD_RTC	Enables weak pull-up (20 KΩ ± 35%).
E_PULLD	In	VDD_SOC/VDD_RTC	Enables weak pull-down (20 KΩ ± 35%).
E_PBIAS_BUF	In	VDD_SOC/VDD_RTC	Enables internally generated bias for driver PMOS in 3.3 V mode. Selects strong driver with internally generated 0.5*VDDP when VDDP is LOW and VCLAMP is HIGH. Wrong setting combination with VREF_SEL may cause electrical stress resulting in pad damage.
VREF_SEL_[3:0]	In	VDD_SOC/VDD_RTC	Select different bias levels for driver PMOS, when (E_PBIAS_BUF == 1)
SCHMIT_[1:0] [2]	In	VDD_SOC/VDD_RTC	Schmitt trigger hysteresis window control.

Notes:

[1] The VDD_SOC/ VDD_RTC power domain depends on VXVD (or VDVX)/VD* variant respectively.

[2] Schmitt trigger hysteresis controls are as follows:

SCHMIT_[1:0]	(E33V == 1)			(E33V == 0)		
	V_{IL}	V_{IH}	window	V_{IL}	V_{IH}	window
00	1.13	1.15	0.02	0.89	0.90	0.01
01	1.09	1.18	0.09	0.84	0.93	0.09
10	1.06	1.22	0.16	0.80	0.97	0.17
10	1.02	1.28	0.26	0.76	1.01	0.25

Table 8.37 BDSDMEMLV Electrical Controls (BDSDMEMLV Variants)

Signal Name	In/ Out	Power Domain	Description
E_INPUT	In	VDD_SOC/VDD_RTC	Enables receiver path I/O → ZI/TZI
E_SCHMT	In	VDD_SOC/VDD_RTC	Enables Schmitt trigger in the receiver
E_HSM	In	VDD_SOC/VDD_RTC	Reserved.
IO_RESET_N	In	VDD_SOC/VDD_RTC	Reserved.

Signal Name	In/ Out	Power Domain	Description
E_PREEMP	In	VDD_SOC/VDD_RTC	Reserved.
E_LPBK	In	VDD_SOC/VDD_RTC	Enables loopback mode
DRV_TYPE_[0]	In	VDD_SOC/VDD_RTC	1x/2x I/O driver selection
DRV_TYPE_[1]	In	VDD_SOC/VDD_RTC	Drive codes selection: COMP pad driven codes or default codes.
E_PULLU	In	VDD_SOC/VDD_RTC	Enables weak pull-up (~18 KΩ ± 30%).
E_PULLD	In	VDD_SOC/VDD_RTC	Enables weak pull-down (~18 KΩ ± 30%).
DRVDN_[7:0]	In	VDD_SOC/VDD_RTC	I/O Driver Output Pull-Down drive strength code
DRVUP_[7:0]	In	VDD_SOC/VDD_RTC	I/O Driver Output Pull-Up drive strength code

Note: VDD_SOC/VDD_RTC power domain depends on VXVD (or VDVX)/VD* variant respectively.

Table 8.38 BDPGLP, BDPGLPHVIN Electrical Controls (BDPGLP, BDPGLPHVIN Variants)

Signal Name	In/Out	Power Domain	Description
E_INPUT	In	VDD_SOC/VDD_RTC	Enables receiver path I/O → ZI/TZI
E_SCHMT	In	VDD_SOC/VDD_RTC	Enables Schmitt trigger in the receiver
E_LPBK	In	VDD_SOC/VDD_RTC	Enables loopback mode A -> ZI/TZI
E_PULLU	In	VDD_SOC/VDD_RTC	Enables weak pull-up (~100 KΩ ± 15%)
E_PULLD	In	VDD_SOC/VDD_RTC	Enables weak pull-down (~100 KΩ ± 15%)
DRVDN[4:0]	In	VDD_SOC/VDD_RTC	I/O Driver Output Pull-Down drive strength code
DRVUP[4:0]	In	VDD_SOC/VDD_RTC	I/O Driver Output Pull-Up drive strength code
E_LPDR	In	VDD_SOC/VDD_RTC	Disables most base driver fingers, leaving only minimal base driver finger. Enabled for I^2C interface to meet slow rise/fall time spec of low-speed modes.
E_IO_HV	In	VDD_SOC/VDD_RTC	Enables open-drain pull-up capability to 3.3 V. Enables 3.3 V Receiver. Relevant for BDPGLPHVIN* variant pad only.

Note: VDD_SOC/VDD_RTC power domain depends on VXVD(or VDVX)/VD* variant respectively.

Table 8.39 BILPRST Electrical Controls (BILPRST_VD Variants)

Signal Name	In/ Out	Power Domain	Description
RCVR_SEL	In	VDD_RTC	Selects between VDD-ONLY or VDD-AND-VDDP receivers. 0: Selects VDD-ONLY Receiver 1: Selects VDD-AND-VDDP Receiver

8.4.1.4.6 Power Sequence Controls

Power sequence controls are the specific controls to handle sequencing in and out of different power states a MPIO pad supports.

Table 8.40 BDSDMEM Power Sequence Controls (BDSDMEMVD*, BDSDMEMVXVD)

Signal Name	In/ Out	Power Domain	Description
E_33V	In	VDD_RTC	Selection of 1.8/3.3 V I/O output voltage swing.
SEL_DPD	In	VDD_TRC	Latches state of I/O set by E_33V/E_PULLU/E_PULLD and data level shifters.
E_DPD	In	VDD_RTC	Places pad in DPD mode by deactivating bias, clamping settings for DPD mode.

Table 8.41 BDSDMEMLV Power Sequence Controls (BDSDMEMLV_VXVD)

Signal Name	In/ Out	Power Domain	Description
E_18V	In	VDD_RTC	Selection of 1.8/1.2 V I/O output voltage swing.
SEL_DPD	In	VDD_TRC	Latches state of I/O set by E_18V/E_PULLU/E_PULLD and data level shifters.
E_DPD	In	VDD_RTC	Places pad in DPD mode by deactivating bias, clamping settings for DPD mode.

Table 8.42 EMMCIO_BRICK Power Sequence Controls (BDEMMC_IOBRICK)

Signal Name	In/ Out	Power Domain	Description
SEL_DPD	In	VDD_TRC	Latches state of I/O set by E_18V/E_PULLU/E_PULLD and data level shifters. Disable Rx.
E_DPD	In	VDD_RTC	Places pad in DPD mode by deactivating bias, clamping settings for DPD mode, and gating-out inputs from core.

Table 8.43 BDPGLP Power Sequence Controls (BDPGLP_VD, BDPGLP_VDVX, BDPGLP_VXVD)

Signal Name	In/ Out	Power Domain	Description
SEL_DPD	In	VDD_TRC	Latches state of I/O set by E_18V/E_PULLU/E_PULLD and data level shifters.
E_DPD	In	VDD_RTC	Places pad in DPD mode by deactivating bias, clamping settings for DPD mode.

Notes:

- For VD variants, all the pad inputs and outputs are in VDD_RTC power domain.
- VDD_RTC/VDD_SOC power domain depends on VDVX/VXVD variant respectively.

Table 8.44 BDPGLPHVIN Power Sequence Controls (BDPGLPHVIN_VD, BDPGLP_HVIN_VDVX, BDPGLP_HVIN_VXVD)

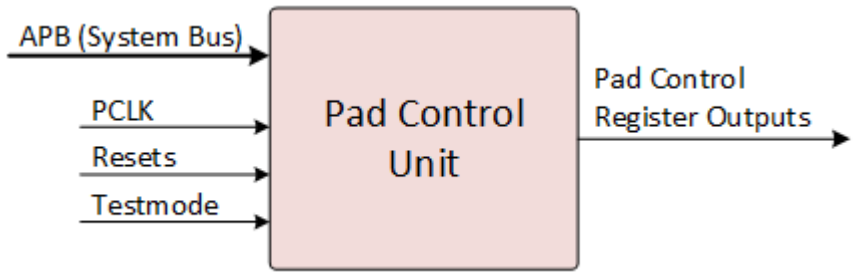
Signal Name	In/ Out	Power Domain	Description
SEL_DPD	In	VDD_TRC	Latches state of I/O set by E_18V/E_PULLU/E_PULLD and data level shifters.
E_DPD	In	VDD_RTC	Places pad in DPD mode by deactivating bias, clamping settings for DPD mode.

Note: VDD_RTC/VDD_SOC power domain depends on VDVX/VXVD variant respectively.

8.4.1.5 Pad Control Unit

Pad control units are the APB clients with register banks for pad controls and PinMux selection. The pad controls provide option to program the pad electrical characteristics depending on platform design and/or connected devices. Each pad control units consists of registers for multiple pads of a PAD group. The following diagram shows the I/O of a Pad Control Unit.

Figure 8.33 Pad Control Unit and Its I/O



8.4.1.6 Pad Control Grouping

The Orin Pad control units are grouped by functions as listed in the following table.

Table 8.45 Orin Pad Control Groups

Pad Control Group	GPIO Support	Control Registers Base Address Block
AO	Yes	PADCTL_A14
AO_HV	Yes	PADCTL_A15
DEBUG	Yes	PADCTL_A5
EDP	Yes	PADCTL_A16
EMMC	No	PADCTL_A6
EQOS	Yes	PADCTL_A21
PEX_CTL	Yes	PADCTL_A7
PEX_CTL_2	Yes	PADCTL_A20
PEX_CTL_3	Yes	PADCTL_A25
QSPI	Yes	PADCTL_A11
SDMM1_HV	Yes	PADCTL_A8
SYS	Yes	PADCTL_A12
UFS	Yes	PADCTL_A17
G2	Yes	PADCTL_A13
G3	Yes	PADCTL_A0
G4	Yes	PADCTL_A4
G5	Yes	PADCTL_A2
G7	Yes	PADCTL_A24

Pad Control Group	GPIO Support	Control Registers Base Address Block
G8	No	PSC_PADCTL
G9	Yes	FSI_PADCTL_A0
G10	Yes	FSI_PADCTL_A1
G11	Yes	FSI_PADCTL_A2

8.4.1.7 PinMux

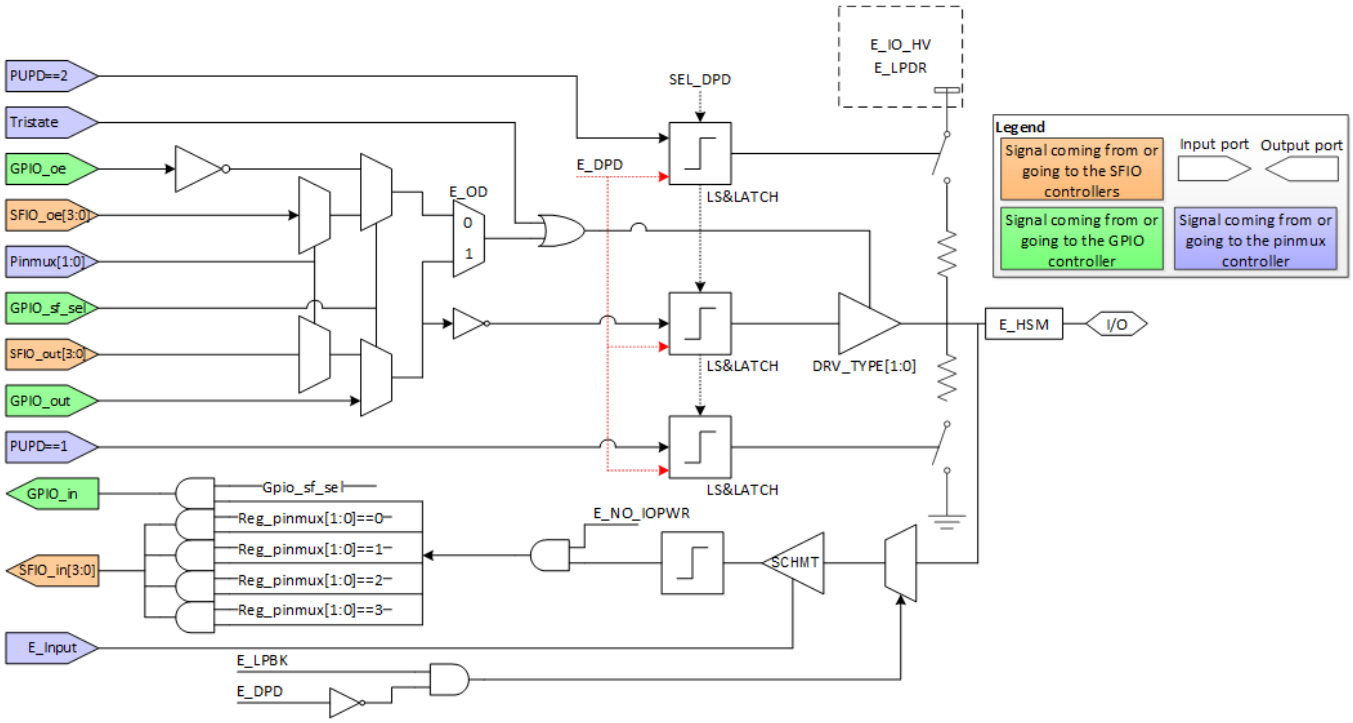
The concept of PinMux logic is to share the same pin for multiple interfaces which aren't active at the same time in every platform. Sharing of pins using PinMux optimizes the number of pins required in SoC thus reducing the cost of die. Orin connects to various peripherals on different platforms, but not all peripherals in single platform. Based on targeted market segment platform goals, we arrive at PinMuxing selection. Respective platform software shall configure the pins at boot or post-boot stages prior to communication with related peripheral devices.

PinMux option per pin consists of:

1. Sharing the pin with four different Special Functional I/Os (SFIOs) and one GPIO, *i.e.* a pin that is not required for SFIO can be used by GPIO-driven software.
2. Override provision to float the pad outputs irrespective of the driving SFIO controller.

The PinMux Controller includes the logic and registers to select a pin for one of the four SFIO. The following diagram shows the PinMux logic associated with a single MPIO pad. The diagram also shows interface with various units and polarity of the signals at the interface level. At pad level when (ENB == 1), the output drivers are disabled.

Figure 8.34 PinMux Logic



8.4.1.7.1 PinMuxed Interfaces

The following table provides a high-level list of interfaces connected with PinMux logic. Note that this also covers the cases of interfaces which have only one SFIO usage (*i.e.* only one PinMux option) and possible to use as GPIO pin. For signal level muxing details, refer to the Orin *Pinmux* document.

Table 8.46 Interfaces Connected Using PinMux Logic

S. No	Interface Name/ IO Rail	Brief Description	Pin-out Spec Reference
1	SDMMC	For Interfacing with WiFi modem etc. or for the 2nd Removable SD card.	SDMMC1
2	EAVB	For Automotive Ethernet Audio Video Bridge	EQOS
3	UART	For connecting various peripherals like BT via UART and Audio codec/FM etc. via I2S interfaces	UART
4	I2S	For various Audio Codec and Digital Speakers	I2S1-I2S8
5	DMIC, DSPKR, IQC	Digital Mike, Codec, DAB Interface for automotive	DMIC1- DMIC5 DSPKR0, DSPKR1

S. No	Interface Name/ IO Rail	Brief Description	Pin-out Spec Reference
6	I^2C	Connection to various sensors, VRs, GMSL aggregators, etc.	I2C1, I2C2, I2C3, I2C4, I2C5, I2C6, I2C7, I2C8, I2C9, I2C10, I2C15
7	QSPI	QSPI interface (automotive boot media)	QSP0*, QSPI1*
8	CAN	Interface to the CAN PHY	CAN0*, CAN1*
9	DIRECTDC	Debug Interface	DIRECTDC1*
10	SPI	Touch device in Mobile, MCU connection in Automotive, Serial flash for BIOS in Chrome platform etc.	SPI1*, SPI2*, SPI3*, SPI4*, SPI5*
11	PWM	Connection to high-current power regulators and FAN.	PWM1 ~ PWM8
12	DP-AUX	Display interface sideband channel.	DP_AUX_CH0*

8.4.2 Functional Description

8.4.2.1 Programmable Drive Strength Capabilities

This section summarizes the programmable drive strength capabilities of various pads in the granularity of pad groups.

1. SDMEM pads which do not have calibration pad, do not have programmable drive strength registers. The DRVUP_VCLAMP [6:0]/DRVDN_VCLAMP [6:0] codes are hard wired to corresponding power group VCLAMP and GND respectively. Pins in the following pad group have a 2-bit back-up drive strength control:

- AO_HV, AUDIO_HV

2. SDMEMLV pads without calibration pad have programmable drive strength registers as in the following pad group:

- UFS, EQOS

3. BDPGLP, BDPGLPHVIN pads without calibration pad have programmable drive strength registers as in the following pad groups:

- All CONN, CAM, and MPIO pad groups as listed in the Orin *Pinmux* document.

4. SDMEM pads with calibration pad have their drive strength controlled by the COMP pad and their overriding and programmable drive strength registers as in the following pad groups:

- SDMMC1_HV, SDMMC3_HV

5. SDMEMLV pads with calibration pad have their drive strength controlled by the COMP pad and their overriding and programmable drive strength registers as in the following pad group:

- QSPI

The typical drive impedance is 50 Ω for all the cases (**1.** through **5.**) above.

For all the pads that do not have corresponding calibration pads (*i.e.* **1.** through **3.** above) the drive strength must be programmed. The Power-On Reset default value in such cases typically works fine as the low interface speed and trace design does not need higher drive codes.

For all the pads with the drive strength calibration feature and hardware drive strength calibration logic, their interface initialization needs to include such drive strength calibration. Software must follow the individual unit's interface initialization guides to ensure that drive strength calibration is taken care of

- one-time during power-up, or
- periodically by programming the drive codes to compensate for PVT variations.

8.4.2.2 Pin-states and Default Pad Controls

Each of the Orin I/O's default pin-state is defined based on primary interface bus idle signal level requirements. Other than functional interface pins, GPIO pins default pin-state is pull-down. There are few exceptions like GPIOs used for button matrix require default pin-state to be pull-up to save external strong pull-ups and platform cost. Ultimately, the Power-On Reset field in the Orin *Pinmux* document captures this information per pin including all the exceptions based on platform consideration.

Either during cold-boot or warm-boot power cycling, pin-states defaults to the Reset values captured in the POR field of the Orin *Pinmux* document. The following provides the connection to Reset types which has implications to pin-states of pad controls.

Further, the following per pin fields as captured in the Orin *Pinmux* document are used to derive the pad control field Reset default value for each pin.

- POR – Pin state at Reset/Power-On
  - Z (Pin Tri-stated).
  - PU/1 (Pin Pulled-Up).
  - PD/0 (Pin Pulled-Down).
- Tristate_en
  - Passthrough (Pin active, *i.e.* drive value for output; transparent for input).
  - Tristate (Pin Tri-stated).
- PU/PD
  - Pad control setting Pull-Up/Down is enabled for pad internal weak Pull-Down or Pull-Up.
- E_INPUT

    - Pad control setting to enable input receiver path which is irrespective of pad I/O direction.
- E_IO_HV
    - Pad control setting to select for 3.3 V tolerance. Applicable for BDPGLPHVIN, open-drain capable pads.
- E_LPDR
    - Pad control setting to turn of drivers inside pad. Applicable for BDPGLP and BDPGLPHVIN pad types.

Table 8.47 Resets Impacting Pin-state

Pins	Reset connection
All pads on I/O rails excluding VDDIO_PWR_CTL	L2 warm Reset
Pads on VDDIO_PWR_CTL	L1 warm Reset

For the pins in VDDIO_PWR_CTL all the pad control registers, associated GPIO logic works on L1 warm Reset unlike all MPIO pads in various other groups.

8.4.2.2.1 GPIO Pin-state, Pad Controls Managed by software

Apart from the pin-states, pad controls coming up as Reset defaults, following two cases where software manages the pin-state, pad control based on platform configuration.

- GPIO/special purpose pins which are part of I/O rails that are active during SC7 power state transitions shall be managed by software.
- GPIO/special purpose pins which are not part of Always-on I/O rails but require different pin-state vs the default pin-state are managed by software as per customer PinMux configuration in Customer PinMux.

8.4.2.3 I/O Electrical Capabilities and Special Handling

The Orin I/Os are typically connected to various external devices powered by regulators which may be shared between Orin and other system components. Given this board-level design nature, and to satisfy the specifics of power-up, power-down, and transitioning in and out of SC7 (or equivalent) Low-Power states, the following system-level exposure scenarios are possible:

- **Voltage Tolerance -** I/Os can be exposed to the maximum tolerance voltage, when the I/O power-rails are on.
- **Fail-Safe -** I/Os can be exposed to the maximum fail-safe voltage resulting in high-leakage current without I/O damage, when the I/O power-rails are off. Given that the Fail-safe situation could result in high-leakage current, the leakage current calculation for reference is given as follows:

  - Leakage current (estimated) = [External device voltage (1.8 or 3.3 V) – Orin OFF VDDP voltage] / External device drive strength.
- **Back-Drive -** I/Os can be pulled HIGH without back-driving to off-power-rail, even when I/O power-rails are off.

This section provides an overview of the Orin I/O electrical capabilities with respect to these different system-level scenarios.

8.4.2.3.1 Back-Drive Tolerant GPIO/LSIO Interfaces

Interfaces or pins that are expected to be driven by the external components with relevant pad power rails switched OFF require a special pad type with back-drive support. In Orin, BDPGLPHVIN_* & BILPRST_* pad types are built for such purposes.

I/O with Back-Drive Capability implies that:

- 1.8 V tolerant, 1.8 V fail-safe, and 1.8 V Back-Drive capable (*i.e.* when Orin is powered off, it is okay for these I/Os to be pulled HIGH to 1.8 V without any Back-Drive current).
- Not 3.3 V fail-safe (*i.e.* when Orin is powered off, I/O cannot be pulled HIGH to 3.3 V to avoid over-voltage damage of the transistors).

In Orin, such back-drive capable pads are used for a sub-set of low-speed interface ports:

- UART
- I^2C
- SPI
- GPIOs for Orin power sequencing and management.

All of the high-speed pads are not back-drive capable, but instead have the capability for voltage tolerance or Fail-safe. A summary of I/O pads with back-drive capabilities along with voltage tolerance and fail-safe capabilities is provided in the following table.

Table 8.48 I/O Voltage Tolerance, Fail-safe, Back Drive Capability Overall Summary

I/O Name	IO Supply Voltage	IO Voltage Tolerance	Fail-Safe Voltage Tolerance	Back-Drive Capability	Back-Drive Current
BILPRST_* (Both variants Core-Only, and Core and IO)	1.8 V	1.8 V	1.8 V	Yes	< 20 µA
BDPGLP_*, BDSDMEMLV_*	1.8 V	1.8 V	1.8 V	No	N/A
BDPGLHVIN_*	1.8 V	1.8v for push-pull, 3.3.v for open-drain	1.8 V	Yes	< 20 µA

I/O Name	IO Supply Voltage	IO Voltage Tolerance	Fail-Safe Voltage Tolerance	Back-Drive Capability	Back-Drive Current
BDSDMEM*	1.8/3.3 V	3.3 V	1.8 V	No	N/A
BDUSB2*	1.8/3.3 V	3.3 V	3.3 V (Provided there is 1 KΩ in-series resistor)	No	~ 2 mA
BDDPAUX*	1.8 V	3.3 V	1.8 V	Yes	< 20 µA
BOPEXCLK*	1.8 V	1.8 V	1.8 V	No	N/A
UPHYDS*	0.975/1.35 V	1.35 V	No	No	N/A
DP_SINGLE*	0.95 V	0.95 V	0.95 V	No	N/A
BDMIPI16X*	1.2 V	1.2 V	1.2 V	No	N/A
LP5_*	1.1/0.5/0.6 V	1.1/0.5/0.6 V	No	No	N/A

8.4.2.3.2 Pads and Bricks Support for Uncontrolled Shutdown and I/O

The Orin pads and bricks are specially designed to properly handle uncontrolled shutdown, where one or all of the core power rails (VDD_SOC, VDD_RTC) is powered down with the I/O power rails still active which can cause permanent pad damage from electrical stress if without proper circuit designs.

8.4.2.3.3 Programmable Drive Strength Capabilities

This section summarizes the programmable drive strength capabilities of various pad groups.

1. BDSDMEM_* Pads

Controlled by programmable drive strength registers.
The DRVUP_VCLAMP [6:0]/DRVDN_VCLAMP [6:0] codes are hard-wired to the corresponding power group VCLAMP and GND respectively.
Pins in the following pad groups have a 2-bit back-up drive strength control using this pad variant.

- AO_HV

2. BDSDMEMLV_* Pads

Controlled by programmable drive strength registers.
Pins in the following pad groups use this pad variant:

- UFS, EQOS

3. BDPGLP_*, BDPGLPHVIN_* Pads (Common/GPIO/MPIO pad type)

Controlled by programmable drive strength registers.
Pins in the following pad groups use this pad variant:

- Many LSIO pad groups as listed in the Orin *Pinmux* document.
- Few pads in DEBUG.

4. BDSDMEMCOMP_* Pad

This is a calibration pad for drive strength calibration to override the programming drive strength register settings if selected. This pad variant works for both 1.8 and 3.3 V I/O signaling levels.
Pins in the following pad group use this pad variant:

- SDMMC1_HV

5. BDSDMEMLVCOMP_* Pad

This is a calibration pad for drive strength calibration to override the programming drive strength register settings if selected. This pad variant works for both 1.2 and 1.8 V I/O signaling levels.
Pins in the following pad groups use this pad variant.

- QSPI, DEBUG, EQOS

The typical design goal for the 5 pad types above is to have a drive impedance of 50 Ω. For interfaces using pads that do not have the corresponding calibration features, *i.e.* type **1.** through **3.** above, the drive strength must be programmed.

The power-on default values by the drive strength registers are expected to meet the typical low-speed interface requirements and general-purpose sidebands electrical specs. Based on the silicon characteristics, software needs to make certain to handle such interface initialization properly.

For pads type **4.** and **5.** above, their interface initialization triggers hardware drive strength calibration. Software must follow the individual unit's interface initialization guides to ensure that drive strength calibration is taken care of

- one-time during power-up, or
- periodically by programming the drive codes to compensate for PVT variations.

8.4.2.4 External Wake Events

External wake events are I/O state change(s), which are defined as triggers to notify the Orin SoC power management logic to indicate some action is needed requiring the SoC to exit low power state. For example, HDMI interfaces define a pin called Hot plug detect (HPD), and wake event associated with it. Upon a hot plug-in connection of HDMI, the event is notified as pin-state value

change on HPD pin. We define number of such external wake events to cater multiple platform use cases. Upon I/O state change, “wake engine” logic in PMC registers the event and triggers power manager hardware. For more details on wake event programming model, wake processing capabilities refer to PMC section.

Orin I/Os capable as “Wake pins” are indicated with wake event numbering in the Orin *Pinmux* document.

8.4.2.5 GPIO Controller to Ports Mapping

The mapping of GPIO controllers to GPIO Ports and Pad Control Groups is discussed here.

- GPIOs in the SYS, AO, and AO_HV pin group are on the VDD_RTC (Ungated Always-ON core power supply).
- GPIOs in G9, G10 and G11 groups are on the ungated VDD_FSI core power supply.
- GPIOs in the rest of the pin groups are on the ungated VDD_SOC core power supply.

Table 8.49 GPIO Controllers for Different Pad Control Groups

GPIO Controller	GPIO Ports	Pad Control Groups	Padlet Group / Voltage Domain
GPIO_CTL0	A (8)*	G5	VDDIO_G5
	B (1)*	DEBUG	VDDIO_DEBUG
	AC (8)* AD (4)*	G7	VDDIO_G7
GPIO_CTL1	X (8)* Y (8)* Z (8)*	G2	VDDIO_G2
GPIO_CTL2	M (8)* N (8)*	EDP	VDDIO_EDP
	P (8)* Q (8)* R (6)*	G3	VDDIO_G3
GPIO_CTL3	K (8)* L (4)*	PEX_CTL	VDDIO_PEX_CTL
	AE (2)*	UFS	VDDIO_UFS
	AF (4)*	PEX_CTL_2	VDDIO_PEX_CTL_2
	AG (8)*	PEX_CTL_3	VDDIO_PEX_CTL_3
GPIO_CTL4	G (8)* H (8)* I (7)*	G4	VDDIO_G4

GPIO Controller	GPIO Ports	Pad Control Groups	Padlet Group / Voltage Domain
GPIO_CTL5	C (8)* D (4)*	QSPI	VDDIO_QSPI
	E (8)* F (6)*	EQOS	VDDIO_EQOS
	J (6)*	SDMMC1_HV	VDDIO_SDMMC1_HV
GPIO_AON	AA (8)* BB (4)*	AO_HV	VDDIO_AO_HV
	CC (8)* DD (3)*	AO	VDDIO_AO
	EE (8)* GG (1)*	SYS	VDDIO_SYS
GPIO_FSI_CTL0	S (8)* T (2)*	G9	VDDIO_G9
	W (2)*	G11	VDDIO_G11
GPIO_FSI_CTL1	U (8)* V (1)*	G10	VDDIO_G10

* Number of GPIO Pins in the given GPIO Port.

8.4.2.6 DFD OBS (Observation Bus) Signals

In DFD mode, the selected unit's related signals are routed through certain pins for debug purpose. For more information on the observation bus selection and usage in DFD mode, please refer to the DFD chapter of this TRM. There are a total of 32 pins as summarized in the following table for the Observation Bus (OBS) to aid the SoC debug.

Table 8.50 DFD OBS Pins

Pin Name	Pins	Pin Count	Max Freq	IO Rail Group	Normal Usage / Comment
UART5	TX, RX, RTS, CTS	4	250 MHz	VDDIO_G2	Serial Port
UART2	TX, RX, RTS, CTS	4	250 MHz	VDDIO_G2	Serial Port
GEN2_I2C	CLK, DAT	2	250 MHz	VDDIO_AO	I^2C Control Port
GEN8_I2C	CLK, DAT	2	250 MHz	VDDIO_AO	I^2C Control Port
GEN3_I2C	CLK, DAT	2	250 MHz	VDDIO_G3	I^2C Control Port
SOC_GPIO23	GPIO	1	250 MHz	VDDIO_G3	Camera GPIO
SPI5	DOUT, DIN, CSO, SCK	4	250 MHz	VDDIO_G7	SPI5 port

Pin Name	Pins	Pin Count	Max Freq	IO Rail Group	Normal Usage / Comment
SOC_GPIO57	GPIO	1		VDDIO_G7	
SOC_GPIO45 **SOC_GPIO46** **SOC_GPIO47** **SOC_GPIO48**	DAP1 pins (SCLK, DOUT, DIN, FS)	4	250 MHz	VDDIO_G7	Digital Audio Port
DAP4	SCLK, DOUT, DIN, FS	4	250 MHz	VDDIO_G5	Digital Audio Port
DAP6	SCLK, DOUT, DIN, FS	4	250 MHz	VDDIO_G5	Digital Audio Port

8.4.2.7 DPD

DPD refers to the SoC pads Deep Power Down state which may be the case during SC7 and/or such low-power states of the SoC. In SC7 scenario, SoC power, *i.e.* VDD_SOC (same as VAUXC_CORE/ VDD_SOC) rail is cut off. Pad DPD control signal SEL_DPD assertion enable the latches inside the pad to store pad A/EN/E_PULLU/E_PULLD pad controls. If I/O rail is on, latched values are driven out during SC7. So, the SoC's I/O state prior to entering to SC7/low power state is retained while the pad is in DPD mode.

Essentially, I/O DPD sequencing on/off consists of:

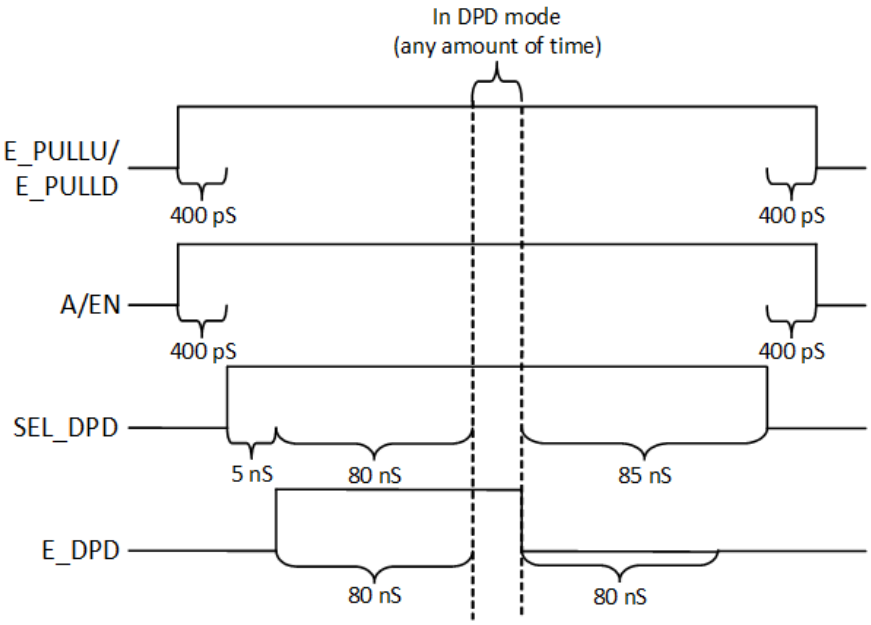
- Pads DPD mode is handled by the unit/PMC hardware managed.
- I/O or core rail power on/off is managed by software in combination with platform power tree design.

SC7 is a Low-Power/Standby state in which Orin maintains much of its I/O state while most of the core logic is powered off. It is possible that pads are transitioned in to DPD mode in many other scenarios beyond Orin SC7 and shall not be inferred as one-to-one correspondence. For example, some pads in the DPD mode may be completely powered down including I/O rails while others are still powered. External wakes and platform back-drive usage cases are platform variant.

8.4.2.7.1 MPIO Pad DPD Sequence

The following diagram shows the sequencing and timing relations of power sequence signals that are necessary to transition a MPIO pad transition in and out of DPD mode. All MPIO pads follow same DPD sequence driven by PMC logic.

Figure 8.35 MPIO Pad DPD Sequence Diagram



8.4.2.7.2 DPD Control Mechanisms

The LSIO and MPIO pads can transition to Low-Power mode (DPD) for power saving even when the corresponding interfaces are still active.

Apart from the SC7 hardware sequencer, software is allowed to transition the pads into such Low-Power state by writing to the DPD registers in PMC, often referred as the DPD knobs. This section provides an overview of the DPD control mechanisms of Orin I/Os with respect to the following:

- low-power state transition or otherwise,
- sequenced by the PMC SC7 hardware sequencer or otherwise,
- DPD control per-pin or group-wise, and
- specific logic driving the DPD controls of the pads.

Given that the DPD control groups cover all Orin I/Os, it is worth noting that the PMC hardware (in VDD_RTC and Always-On core power) controls the DPD of a subset of Orin I/Os as specified in the PMC IMPL Registers.

The DPD control mechanisms are classified in the following groups.

- **Group A** (I/Os with no DPD mode support at all)
  These pads are associated with the PMIC for system initialization and Reset, and do not enter the DPD mode during SC7. Specifically, these are inputs only, and are always needed across SoC power states.

- **Group B** (I/Os with their DPD controls tied to 1'b0 with JTAG overrides)
  These pads must not go into DPD state because of their critical functional nature and implications to power sequencing and chip functioning during operation or IST mode. Unlike those in Group A, the pads in this group are bi-directional in functions.
- **Group C** (I/Os with DPD knob per pin)
  These pads are in the Always-On I/O rails, or need to remain active for chip observation during IST.
  The I/Os on the Always-On I/O rails have per-pin DPD override masks to enable their exclusion from SC7 entry/exit and PMC global DPD sequencing. Refer to the PMC IMPL Registers for a list of such DPD knobs.
- **Group D** (I/Os with DPD controls not by the PMC hardware but the registers in the individual cluster for security and safety considerations)
  Software must properly handle DPD programming for power saving or disabling before accessing these I/Os. Refer to the PSC and FSI chapter in this TRM for more details.
- **Group E** (I/Os transitionable to DPD state by group-level DPD controls by the PMC hardware for low-speed interfaces)
  These pads must go through PMC global DPD sequencing at SC7 entry/exit regardless of the software DPD control programming. Refer to the PMC IMPL Registers for a list of such DPD knobs.
- **Group F** (I/Os transitionable to DPD state by group-level DPD controls by the PMC hardware for non-Group-G high-speed interfaces)
  These pads have DPD mode just like other Orin MPIO/GPIO pads. The global PMC Finite State Machine manages the DPD control at power on/off, while the DPD overrides are by the individual unit's AO (Always On) logic. Refer to the PMC IMPL Registers for a list of DPD knobs corresponding to the USB2 bricks.
- **Group G** (I/Os transitionable to DPD state by group-level DPD controls by the PMC hardware for DDR, Display, and HSIO high-speed interfaces)
  These I/Os' DPD mode is not controlled by the PMC hardware at SC7 entry/exit. Instead, the cluster/unit's logic owning the interface must provide appropriate clamps to manage to put the pad in Low-Power state. Refer to the DDR, Display, and HSIO chapters in this TRM for such details.

PCIe Sidebands and Clock Out DPD Controls

This section provides details of the PCIe sidebands I/Os including PCIe clock outs.

The PCIe sidebands as part of 3 pin groups are controlled by the following PMC registers bit fields:

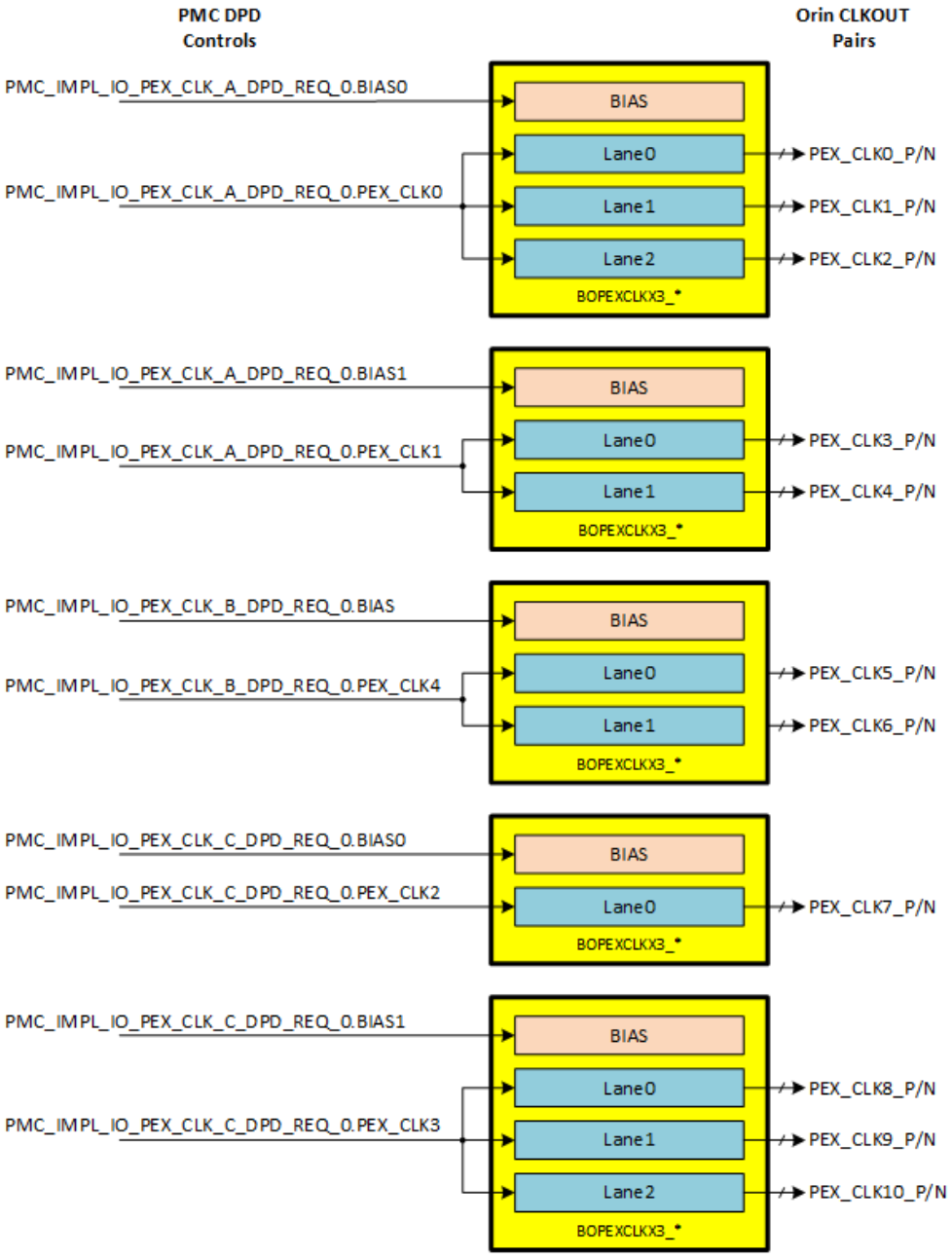
- PMC_IMPL_IO_PEX_CTL_A_DPD_REQ_0.PEX_CTL
  - controls the PCIe sidebands of PCIe controllers C0, C1, C2, C3, and C4, which are mapped to HSIO UPHY.
- PMC_IMPL_IO_PEX_CTL_B_DPD_REQ_0.PEX_CTL_2
  - controls the PCIe sidebands of PCIe controllers C5 and C6, which are mapped to NVHS UPHY.

- PMC_IMPL_IO_PEX_CTL_C_DPD_REQ_0.PEX_CTL_3
  - controls the PCIe sidebands of PCIe controllers C7, C8, C9, and C10, which are mapped to GBE UPHY.

For details of these registers, please refer to the PMC IMPL Registers in the PMC chapter.

Each BOPEXCLKX3_* has a single DPD control to manage 3 clock pairs and a separate DPD control for BIAS. The following diagram provides the overview of the DPD controls for all the PCIe clock-outs.

Figure 8.36 PEX CLKOUT Pads Software DPD Control Mapping



8.4.2.8 IST and I/O Considerations

The Orin In-System Test (IST) is the SoC LSM (latent Safety Mechanism) mechanism to detect latent faults as per functional safety requirements. During IST, the DFT runs the logic and memory BIST patterns then logs their results to the embedded storage device (eMMC). The boot-up

software and algorithms at the next boot-up analyze then make appropriate decisions including notifications to users to improve the system reliability.

During such IST, various Orin I/Os are at static state except for the few needed in the IST function. Based on their states during IST and as described in the sub-sections blow, all the pins are classified into the following categories:

- Functional
- Interrupt
- Toggle
- Static

8.4.2.8.1 Functional

This category of I/O interface signals (pins) is needed for IST to function properly. All SoC logic related to these signals need to be:

- isolated from or totally bypassing the BIST logic, and
- driven or processed (as input I/Os) by the IST DFT logic.

This Functional category I/Os are summarized in the following table.

Table 8.51 IST Functional Category I/Os

Signal/Pin Names	IST Pin-state	IST Description
EMMC_CLK	Functional	DFT seed is stored in eMMC.
EMMC_CMD	Functional	DFT seed is stored in eMMC.
EMMC_DAT0	Functional	DFT seed is stored in eMMC.
EMMC_DAT1	Functional	DFT seed is stored in eMMC.
EMMC_DAT2	Functional	DFT seed is stored in eMMC.
EMMC_DAT3	Functional	DFT seed is stored in eMMC.
EMMC_DAT4	Functional	DFT seed is stored in eMMC.
EMMC_DAT5	Functional	DFT seed is stored in eMMC.
EMMC_DAT6	Functional	DFT seed is stored in eMMC.
EMMC_DAT7	Functional	DFT seed is stored in eMMC.
EMMC_DQS	Functional	DFT seed is stored in eMMC.
EMMC_COMP	Functional	DFT seed is stored in eMMC.
XTAL_IN	Functional	Crystal Clock input and output, required by the hardware function of IST.

Signal/Pin Names	IST Pin-state	IST Description
XTAL_OUT	Functional	Crystal Clock input and output, required by the hardware function of IST.
CLK_32K_IN	Functional	32 K clock input, required by the hardware function of IST.
SYS_RESET_N	Functional	PMIC drives it depending on the power state of the system.
AO_RETENTION_N	Functional	Used as IST_DONE notification to MCU at the normal/partial (due to abrupt thermal event) completion of IST. More details are covered in the **Special-purpose GPIOs Actively Managed during IST** sub-section below.
THERM_DP	Functional	Thermal sensor connection for thermal management.
THERM_DN	Functional	
JTAG_TRST_	Functional	JTAG path interface connectivity for IST debug.
JTAG_TDO	Functional	
JTAG_TMS	Functional	
JTAG_TCK	Functional	
JTAG_TDI	Functional	
NVJTAG_SEL	Functional	ARM JTAG (default) or NVJTAG selection.
NVDBG_SEL	Functional	Used for switching to MIPI debug modes in closed-box system.
Various DFD I/Os	Functional	DFD Debug Observation Bus (OBS) signals are listed in the DFD OBS Pins Table in the DFD OBS Signals section above.

The GPIO pads used for DFD must not be transitioned in to DPD state. The DFD observation signals are routed through the pad's test mode inputs over TM/TASEL, etc. Hence, these DFD observation signals do not need any of the following:

- PinMux overrides.
- Scan isolation.
- IST hardware management.

8.4.2.8.2 Interrupts

This category of I/O (input only) interface signals (pins) is needed to channel Interrupts from aggregated platform events to the DFT master logic to cause shutdown of the IST sequence during IST.

This Interrupts category I/Os are summarized in the following table.

Table 8.52 IST Interrupt Category I/Os

Signal/Pin Names	IST Pin-state	IST Description
SOC_GPIO14	Interrupt (functional)	Notification on any of these pins causes IST abrupt shutdown depending on mask selection. These Interrupt lines are of fixed “Active HIGH” polarity convention. The DFT master has provision to enable/mask for I/Os used as aggregated platform level triggers. Register definition of such IST Interrupt filtering controls details can be found in the IST Manuals. The IST retains them in the DPD mode, while the VD/VDVX pad type I/O core side input (ZI_WAKE) is propagated into the core DFT logic.

8.4.2.8.3 Toggle

This category of I/O interface signals (pins) is meant to be used by the various PWM-based Voltage Regulator (VR) controls during IST.

This Toggle category I/Os are summarized in the following table.

Table 8.53 IST Toggle Category I/Os

Signal/Pin Names	IST Pin-state	IST Description
SOC_GPIO07 (pin-mux option GP_PWM6)	Toggle	PWM-based CPU-VR.
TOUCH_CLK (pin-mux option GP_PWM4)	Toggle	PWM-based SOC-VR.
GP_PWM2	Toggle	PWM-based GPU-VR.
GP_PWM3	Toggle	PWM for fan control. Must be turned on during IST to set the fan at some fixed speed.
SOC_GPIO21/GP_PWM5	Toggle	PWM for 2^{nd} fan control (optional feature).
SOC_GPIO19/GP_PWM7	Toggle	PWM based CV-VR.

NOTE:

- PWM controllers as listed in the table above together with related SoC logic need to be isolated from the logic BIST.
- PWM outputs as listed in the table above need PinMux/pad control “JTAG_override”, since this logic portion is not scan-isolated. Specifically, PWM1 with “JTAG_override” provision is not intended to be used to control and config VRs or alike.

The requirements for the PWM interfaces used for High-current VRs in the auto-platform are:

- Voltage must be set at a level required to safely complete the IST sequence.
- PWM ports for VR must keep toggling with outputs not to be sequenced out (IDLE or inactive) by BPMP software towards IST initialization, or by system software during

- IST, and
- shutdown sequence in auto platforms (that are IST relevant).

- An un-used PWM port must be regarded as a Static category I/O (details in the sub-section below) for that platform, with software transitioning it to the DPD state to keep it in safe I/O state.

8.4.2.8.4 Static

This category of I/O interface signals (pins) is expected to maintain at a pre-defined level (pin-state) during IST, including

- certain PWM output signals expected to be parked at certain pre-defined “static” state (depending on platform GPIO usage), and
- any un-used PWM port (to be programmed by software prior to IST like any other Static signals in order to be excluded from DFT boundary scan coverage).

To achieve Static” state,

- the BPMP IST software selects a Toggle I/O as GPIO mode to drive the output state to HIGH or Low, and
- the BPMP IST software includes the selected I/O together with other similar I/Os that need to transition to the DPD state to be parked safe during IST.

Another approach to achieve safe Static state during IST is using the JTAG_override mechanism. However, this involves additional coding in the IST µCode to read the Pad controls (including the PM field for JTAG_override), is therefore not a preferred option.

It is important to note that transition I/Os to the pre-defined pin state is not limited only to the Static category I/Os, but all I/Os that are not active to perform the IST

This Static category I/Os are summarized in the following table.

Table 8.54 IST Static Category I/Os

Signal/Pin Names	IST Pin-state	Description of Chip I/O State & Requirements
SOC_PWR_REQ	Static	Need to be parked at Logic-HIGH state in the system during IST. This ensures that chip's not-Always-On core voltage request is asserted to the PMIC. This is achieved by ▪ JTAG override programming to set up in “tri-state” or “Logic-HIGH” as part of the IST µCode (as described in the **Drive SOC_PWR_REQ HIGH during IST** sub-section below, since the I/O does not support DPD control, thus not covered by DPD control programming), and ▪ a board-level pull-up resistor.
CPU_PWR_REQ	Static	Need to be parked at Logic-HIGH state in the system during IST. This ensures that the CPU core power request is asserted to the PMIC. BPMP firmware needs to drive the GPIO accordingly before IST mode is entered. Subsequently, DPD mode of the pad maintains the I/O state.

Signal/Pin Names	IST Pin-state	Description of Chip I/O State & Requirements
GPU_PWR_REQ	Static	Need to be parked at Logic-HIGH state in the system during IST. This ensures that the GPU core power request is asserted to the PMIC. BPMP firmware required to drive the GPIO accordingly before IST mode is entered. Subsequently, DPD mode of the pad maintains the I/O state.
CV_PWR_REQ	Static	Need to be parked at Logic-HIGH state in the system during IST. This ensures that chip CV core power request is asserted to the PMIC. BPMP firmware needs to drive the GPIO accordingly before IST mode is entered. Subsequently, DPD mode of the pad maintains the I/O state.
SOC_GPIO28 (DGPU_PWR_EN)	Static	Need to be parked at Logic-Low state in the system during IST. This also requires turning off the GPU power rails. BPMP firmware needs to drive the GPIO accordingly before IST mode is entered. Subsequently, DPD mode of the pad maintains the I/O state.
SOC_ERROR_N	Static	Need to be parked at Logic-HIGH state in the system during IST. This ensure that Orin does not flag a fault notification to the MCU, while IST is being performed. This is achieved by the board-level Pull-up resistor as the default state of this I/O driven by HSM logic is “tri-state” (open-drain connection). Subsequently, DPD mode of the pad maintains the I/O state.
BOOTV_CTL_N	Static	Need to be parked at Logic-Low state in the system during IST. This ensures that safe voltage level (Vboot) for VDD_SOC power is supplied by the power regulator powering Orin after IST. This is achieved by the board-level Pull-down resistor as the default pin is tri-state (open-drain selection). Subsequently, DPD mode of the pad maintains the I/O state.
SHUTDOWN_N	Static	Need to be parked at Logic-HIGH state in the system during IST. This ensures that Orin does not trigger shutdown during IST. This is achieved by ▪ JTAG override programming to set up in “tri-state” as part of IST µCode (as described in the **Tri-state SHUTDOWN_N during IST** sub-section below, since the I/O does not support DPD control, thus not covered by DPD control programming), and ▪ a board-level pull-up resistor.

8.4.2.8.5 I/O Safe State Considerations

It is required that all unused I/Os are transitioned to the DPD state and safely parked in a quiescent state prior to IST. The quiescent state of each GPIO can vary from platform to platform. Software must ensure that the required IDLE state is achieved on these I/O interfaces or GPIOs prior to entering IST, followed by DPD/power-down as applicable. The **Transitioning I/Os to DPD State** sub-section below provides details and references to DPD control programming of these inactive or non-operational I/Os during IST.

System software is responsible for keeping the following in a safe and quiescent state as per DPD/power-down sequence programming guidelines:

- all unused MPIO I/O pad groups (with PMC DPD control per group).
- all unused high-speed bricks (with DPD control).

The I/O safe state I/Os are:

1. All the I/Os listed as Interrupt and Static category.
2. Other I/Os not covered by the IST categories. This means all I/Os with no IST implication need to be at IDLE state and managed per functional IDLE state requirements.

The DPD control settings during IST provide the full summary of DPD/power-down controls and settings required by system software transitioning prior to entering IST.

For all such unused I/Os, miniTMC must sample and hold the DPD controls when entering the IST mode and keep them at safe and quiescent state.

8.4.2.8.6 Special-purpose GPIOs Actively Managed during IST

The signals as summarized in the table below need to be

- toggled at the start/end of IST sequence, or
- actively driven to a “Static” state by the DFT hardware.

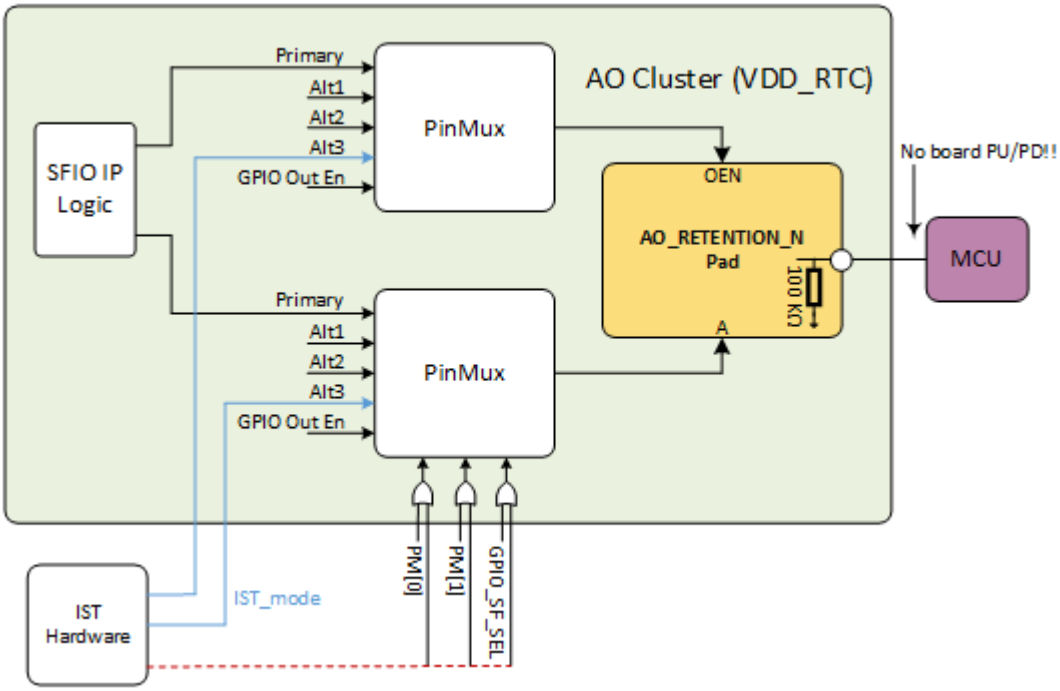
These signals are presented here because they need

- pad inputs (*i.e.* A and OEN as shown in the **IST_DONE Signaling to MCU** diagram below) overrides in the functional path, or
- actively driving a desired Pin-state by the IST firmware or hardware as described below.

Ball Name	Platform Symbol Name	GPIO Register	Usage	Driven by
AO_RETENTION_N	GP02	GPIO_AON Register: PIO_EE_OUTPUT_VALUE_02_0	Output SFIO: Notification to MCU. (IST_DONE)	IST hardware. POR default: Low. IST_DONE_N is used as handshake notification to MCU as described below. **HIGH**: by BPMP-IST at completion of voltage settings. **LOW**: by IST hardware at completion of IST.
SOC_GPIO34	GP186	GPIO_CTL3 register: GPIO_L_INPUT_03_0	Input GPIO: IST request from MCU.	MCU

The following diagram shows the scheme for one of such I/Os, IST_DONE, notification to the external MCU.

Figure 8.37 IST_DONE Signaling to MCU



Software Sequence for IST_DONE_N Output GPIO Driving HIGH

The software programming steps to assert IST_DONE(AO_RETENTION_N) HIGH are as follows:

- Configure Pad control to select the pin function as GPIO including electrical control settings:
  - Write (PADCTL_SYS_AO_RETENTION_N_0.GPIO_SF_SEL = 1'b0)
  - Write (PADCTL_SYS_AO_RETENTION_N_0.E_IO_HV = 1'b0)
  - Write (PADCTL_SYS_AO_RETENTION_N_0.TRISTATE = 1'b0)
- Configure per pin GPIO registers to enable GPIO o/p mode of operation:
  - Write (GPIO_EE_ENABLE_CONFIG_02_0.INOUT = OUT(1'b1)) to select the GPIO pin usage as output.
  - Write (GPIO_EE_OUTPUT_CONTROL_02_0.GPIO_OUT_CONTROL = DRIVEN(1'b0)) to have it driven by GPIO logic, not floated.
  - Write (GPIO_EE_ENABLE_CONFIG_02_0.GPIO_ENABLE = ENABLE(1'b1)) to enable GPIO function.
  - Write (GPIO_EE_OUTPUT_VALUE_02_0.GPIO_OUT_VAL = 1'b1)) to drive GPIO to HIGH value.

Software Sequence for IST_REQ_N Input GPIO

The software programming steps to initialize IST_REQ_N (SOC_GPIO34) HIGH are as follows:

- Configure Pad control to select the pin function as GPIO including electrical control settings:

  - Write (PADCTL_PEX_CTL_SOC_GPIO34_0.GPIO_SF_SEL=1'b0)
  - Write (PADCTL_PEX_CTL_SOC_GPIO34_0.E_IO_HV=1'b0)
  - Write (PADCTL_PEX_CTL_SOC_GPIO34_0.TRISTATE =1'b0)
- Configure per pin GPIO registers to enable GPIO input mode of operation:
  - Write (GPIO_L_ENABLE_CONFIG_03_0.INOUT = IN(1'b0)) to select the GPIO pin usage as output.
  - Write (GPIO_L_OUTPUT_CONTROL_03_0. GPIO_OUT_CONTROL = FLOATED(1'b1)) to float the GPIO (not driven by GPIO logic).
  - Write (GPIO_L_ENABLE_CONFIG_03_0.GPIO_ENABLE = ENABLE(1'b1)) to enable GPIO function.
  - Read GPIO_L_INPUT_03_0.GPIO_IN to check the GPIO pin status.

8.4.3 Programming Guidelines

Some controller instances make their set of signals available on two or more sets of MPIO's. To say it another way, such controllers have more than one "interface."

Note: Before using any controller, make sure that the PinMux registers are programmed to bring out the controller's signals on a maximum of one interface.

8.4.3.1 PinMux Programming

PinMuxing configurations are static. PinMuxing provides the flexibility to choose a pin for different function across different platforms, but not to change them dynamically.

In specific cases, based on use case requirement it is needed to change a pin from SFIO to GPIO or vice versa. Although PinMux selection dynamic changing is not supported, SFIO to GPIO or vice versa changes are allowed. Subsection outlines the PinMux programming, SFIO/GPIO switching programming and such pad control programming guidelines.

8.4.3.1.1 Changing a PinMux Selection

PinMux programming to select the pin function is part of respective I/O initialization. Below sequence are a reference and confines to general purpose pad I/Os which are shared by multiple low-speed interface controllers (SFIO).

- Ensure that all the interface controllers sharing the pins are in in-active state, so that possible glitches do not result corruption in downstream logic.
- Change the PADCTL_<padctl_group>_<ball_name> .PM value to desired I/O function enumerated per pin.
- Initialize the respective SFIO controller and follow required SFIO interface programming guide.

8.4.3.1.2 Driving Source Change from SFIO to GPIO

Pads are by-default owned by SFIO Primary function or Alternative options can be switched as software driven GPIO as follows:

- Configure the pad's GPIO specific registers for correct value for output mode with the correct value as specified in GPIO section. This ensures the new source is ready with the correct value.
- Change the pad control register, *i.e.* PADCTL_<padctl_group>_<Ball_Name>.GPIO_SF_SEL bit field register corresponding a pin to 1'b0 for GPIO selection.
- Follow the GPIO controller programming model depending on desired GPIO function as specified in GPIO section (Input or output to a value or input with Interrupt etc.).

8.4.3.1.3 Driving Source Change from GPIO to SFIO

Pads currently set up in GPIO mode may be switched back to SFIO function as follows:

- Change the pad control register, *i.e.* PADCTL_<padctl_group>_<Ball_Name>.GPIO_SF_SEL bit field register corresponding a pin to "1" for SFIO selection.
- Initialize the respective SFIO controller and follow required SFIO interface programming guide.

8.4.3.1.4 Programming Sequence for Input Mode

After cold boot based on platform configuration (typically obtained from the Boot Loader) different LSIO functional controllers must be mapped to specific I/O pin. The following is the programming sequence for that:

- Set PM field as per PinMux selection in pad control register PADCTL_<padctl_group>_<ball_name>_*.PM =
  - 0x0: Primary
  - 0x1: Alt1
  - 0x2: Alt2
  - 0x3: Alt3
  - Set SFIO/GPIO selection to SFIO selection, (PADCTL_<padctl_group>_<ball_name>_*.GPIO_SFIO_SEL = HSIO) (nothing but SFIO).
  - Set (PADCTL_<padctl_group>_<ball_name>_* .TRISTATE = TRISTATE) to disable SoC functional logic driving the I/O.
  - Set (PADCTL_<padctl_group>_<ball_name>_* .E_INPUT= ENABLE) to turn on input receiver.
  - The specific I/O controllers driving the I/Os is configured and activated.

8.4.3.1.5 Programming Sequence for Output Mode and Bi-directional Mode

Normal Push-pull Pad

- Set PM field as per PinMux selection in pad control register PADCTL_<padctl_group>_<ball_name>_*.PM =
  - 0x0: Primary
  - 0x1: Alt1
  - 0x2: Alt2
  - 0x3: Alt3
  - Set SFIO/GPIO selection to SFIO selection, (PADCTL_<padctl_group>_<ball_name>_*.GPIO_SFIO_SEL = HSIO) (nothing but SFIO).
  - Set (PADCTL_<padctl_group>_<ball_name>_* .TRISTATE = PASSTHROUGH) to enable selected SFIO functional logic driving the I/Os.
  - Set (PADCTL_<padctl_group>_<ball_name>_* .E_INPUT = ENABLE) for bidirectional configuration, or
    Set (PADCTL_<padctl_group>_<ball_name>_* .E_INPUT = DISABALE) for output only configuration
  - The specific I/O controllers driving I/O is thus configured and activated.

Open-drain Pad

Open drain pads are used to emulate Wired-AND functionality, where in multiple drivers can be shorted on the board with an external pull-up. These pads don't drive active HIGH, instead it drives active LOW or float the pin. Platform pull-up shall ensure correct value on the node, *i.e.* if all the drivers are floating it is at Logic1 and if any of the drivers are driving active LOW, bus state is at Logic0. Such pads require a different programming sequence. So, any logic (both functional or software based GPIO) shall never the program/drive pads to Active 1. Both the ST pads and DD pads able to operate in open drain Mode. But the DD pads have specific control to tolerate the I/O swing of 3.3 V.

3.3 V Tolerance and Open-drain Operation

- Configure the pad for 3.3 V swing tolerance by enabling the E_IO_HV pad control, *i.e.* (PADCTL_<padctl_group>_<ball_name>_*.E_IO_HV = 1). Applicable to DD pads when it has to tolerate 3.3 V. Typically, other than the PMIC interface, which is guaranteed to operate at 1.8 V for all the other DD pads pad control field E_IO_HV is set to 1 as Reset default to ensure that they come with 3.3 V tolerance mode. E_IO_HV pad control must be set to 1 before I/O can be pulled to > 1.8 V.
- Set (PADCTL_<pactcl_group>_<ball_ame>_*.TRISTATE = TRISTATE). Even when the functional logic driving Logic 1, the pad is in open drain driver mode and it is floated through an external pull-up. Enabling internal pull-up is ignored by the pad when the E_IO_HV is set though this is not the recommended.

1.8 V Tolerance and Open-drain Operation

- To emulate open drain in 1.8V mode of operation:
  - have the platform design float the pad with external pull-up, or
  - program (PADCTL_<padctl_group>_<ball_name>.PUPD = PULLUP).
- To configure the Pad for 1.8V swing tolerance:
  - setting (PADCTL_<padctl_group>_<ball_name>.E_IO_HV = 0).
- To drive high Impedance into the pad set:
  - (PADCTL_<pactcl_group>_<ball_ame>_*.TRISTATE = TRISTATE).

1.8 V Tolerance/Driving and Push-Pull Operation

Since DD pads can be operated in normal push pull mode as well as open drain mode to emulate the push-pull mode the following configuration is followed.

To put the open-drain pad in 1.8 V push-pull mode operation:

- Set (PADCTL_<padctl_group>_<ball_name>.E_IO_HV = DISABLE).
- Set (PADCTL_<padctl_group>_<ball_name>_* .TRISTATE = PASSTHROUGH) to enable functional logic driving/sampling the I/O.
- Set (PADCTL_<padctl_group>_<ball_name>_* .E_INPUT = ENABLE) for bi-directional mode of operation.
- The specific I/O controllers driving the I/O is configured and activated.

8.4.3.2 Pad Control Programming

8.4.3.2.1 Set Pad to High Impedance State

To select the pad for drive high impedance state:

- Set (PADCTL_<padctl_group>_<ball_name>_* .TRISTATE = TRISTATE).
- Ensure (PADCTL_<padctl_group>_<ball_name>_* .PUPD == 0) to confirm that internal pull-up/pull-down is disabled.

8.4.3.2.2 E_PBIAS_BUF and VREF_SEL Pad Controls

Programming E_PBIAS_BUF and VREF_SEL pad controls to BDSDMEM_* pads incorrectly may result in electrical damage.

E_PBIAS_BUF and VREF_SEL should be programmed in the following cases:

- VDD33 and VDD18 in the same direction
- HIGH VDD33 along with LOW VDD18

To prevent overstress, use the following guidelines when programming E_PBIAS_BUF and VREF_SEL:

1. When (VDD33 - 0.91 * VDD18 > 1.98 V), E_PBIAS_BUF must be 0.
2. When ((VDD33 - 0.83 * VDD18 > 1.98 V) && (E_PBIAS_BUF == 1)), VREF_SEL must be less than 1.
3. When ((VDD33 - 0.77 * VDD18 > 1.98 V) && (E_PBIAS_BUF == 1)), VREF_SEL must be less than 2.
4. When ((VDD33 - 0.71 * VDD18 > 1.98 V) && (E_PBIAS_BUF == 1)), VREF_SEL must be less than 3.

8.4.3.2.3 Pad Controls Sequencing Requirements

1. To avoid remote chances of leakage and long-term reliability issues, E_33V needs to be set before supplying 3.3 V to the 1.8/3.3 V dual-voltage pads.
   a. SDMMC1/SDMMC3 for removable card usage is covered by the SD card voltage switching sequence.
   b. For cases where the I/O voltage usage is limited to 1.8 V, it is done as part PinMux/pad control initialization.
2. (E_IO_HV = 0) needs to be programmed by the PinMux/pad control initialization BDPGLPHVIN pads which does not need to operate open-drain mode. Otherwise, the pad remains in open-drain mode and cannot function as a push-pull driver.

8.4.3.2.4 Pad Control Settings per PinMux Selection

The required Pad control settings for each of the interface varies per their electrical specs. This implies that the Power-on default settings may need to be re-programmed as part of the PinMux/Pad Control initialization during MB1 (Micro-Boot stage1) by MB1 software.

8.4.3.3 I/O Safe State Programming during IST

8.4.3.3.1 Transitioning I/Os to DPD state

As part of the IST Safe state consideration, every unused I/O must be parked in a Static state. This means that BPMP and IST software need to transition every unused pad to the DPD/power-down state following DPD sequencing. Furthermore, software drivers as well as or firmware managing the I/Os needs to transition the interfaces to IDLE state prior to the DPD sequencing such that the I/Os Static state resembles the chip power-on state to avoid any complications with the connected devices.

- For all GPIO/MPIO-speed I/Os and few other bricks (for which DPD controls are driven out from PMC) typical DPD programming sequence is generated by the PMC hardware and SC7 Finite State Machine following the pad DPD sequencing requirement.

- For all GPIO/MPIO-speed I/Os in PSC and FSI, the pad's DPD state is achieved by programming the DPD controls (SEL_DPD, E_DPD) mapped registers in the PSC_AON and FSI PADCTL registers, respectively. A time delay of 1 ms is required between SEL_DPD enabling and E_DPD enabling.
- For all remaining high-speed bricks, like DDR and UPHY bricks, special sequence/steps is required to park them in safe state. Such sequence is not the scope of this chapter. Refer to HSIO and MSS chapter of this TRM for Safe State Parking Programming Sequences.

8.4.3.3.2 Tri-state SHUTDOWN_N during IST

As part of the IST scheme and I/O safe state considerations, the SHUTDOWN_N output pad must be tri-stated during IST by programming the JTAG overrides of the SHUTDOWN_N pad controls:

- Program the JTAG overrides of SHUTDOWN_N pad controls which are used as electrical configuration and to tri-state this I/O:
  - (PADCTL_SYS_SHUTDOWN_N_0.PUPD = NONE).
  - (PADCTL_SYS_SHUTDOWN_N_0.E_IO_HV = ENABLE).
- Set up the JTAG override register bits corresponding to this to drive logic High.

8.4.3.3.3 Drive SOC_PWR_REQ HIGH during IST

As part of the IST scheme and I/O safe state considerations, the SOC_PWR_REQ pad must be parked at the asserted HIGH level by programming the following:

- Set up the JTAG override register bits corresponding to this to drive logic High.

8.4.3.4 Unused I/O Programming

Unused general purpose I/Os in a platform may be left unconnected however pad controls must be set per below guideline to avoid leakage and/or noise especially when I/O power supply is ON. Programming steps for different I/Os is summarized as below.

Table 8.55 Unused I/O Programming with I/O Power Supply ON

PHY/BRICK	PAD Control Programming requirement when I/O is unused but I/O power supply is ON
MPIO pads	Following is done by platform Orin Customer PinMux. So required pad control programming is automatically done through PinMux driver in MB1. No further software steps are needed. ▪ TRI-STATE enable, (PADCTL_<power_group>_<IO>_0.TRISTATE = 1) ▪ Input path disable, (PADCTL_<power_group>_<IO>_0.E_INPUT = 1) ▪ Weak pull-down enable, (PADCTL_<power_group>_<IO>_0.PUPD = 1)
DDR I/O Brick	Park the unused bricks in DPD mode as per the EMC programming guide.
DDR COMP Pads	Park the unused bricks in DPD mode as per the EMC programming guide.

PHY/BRICK	PAD Control Programming requirement when I/O is unused but I/O power supply is ON
DDR_RESET_PAD	Park the unused bricks in DPD mode as per the EMC programming guide.
HSIO UPHY	Park the unused brick lanes in IDDQ mode. ▪ Set IDDQ and RESET to asserted via per lane UPHY PADCTL registers. ▪ Enable pull-downs in the UPHY PADCTL registers.
NVHS UPHY	Park the unused brick in IDDQ mode. ▪ Set IDDQ and RESET to asserted via UPHY PADCTL register. ▪ Enable pull-downs in the UPHY PADCTL registers.
USB2 pads	Park unused bricks in DPD mode. ▪ Set PD* to 1 via XUSB PADCTL register XUSB_PADCTL_USB2_OTG_PAD1_CTL_0_0 Enable weak pull-down by programming following bit fields. ▪ (XUSB_AO_UTMIP_SLEEPWALK_CFG_0. MASTER_ENABLE = 1) ▪ (XUSB_AO_UTMIP_SLEEPWALK_0. USBOP_RPD_A = 1) ▪ (XUSB_AO_UTMIP_SLEEPWALK_0. USBON_RPD_A = 1)
USB2 BIAS pad	Power down the BIAS pad if all the USB2 bricks are not used (less likely) ▪ Set (XUSB_PADCTL_USB2_BIAS_PAD_CTL_0_0.PD = 1).
MIPI CSI/DSI PHY	Park the unused bricks in DPD mode to keep the PAD in quiescent state. ▪ Program fields corresponding to CSI/DSI bricks in PMC registers, *i.e.* PMC_IMPL_IO_CSI_DPD_REQ_0. Enable weak pull-downs in the pad. ▪ (NVCSI_PHY_0_NVCSI_CIL_<A-H>_PAD_CONFIG_0.SPARE_CLK_<A-H> = 1) ▪ (NVCSI_PHY_0_NVCSI_CIL_<A-H>_PAD_CONFIG_0.SPARE_IO1_<A-H> = 1) ▪ (NVCSI_PHY_0_NVCSI_CIL_<A-H>_PAD_CONFIG_0.SPARE_IO0_<A-H> = 1)
MIPI BIAS pad	Park the BIAS pad in DPD mode if none of the CSI ports are used. ▪ Program the MIPI_BIAS bit field in PMC to enable DPD
DP/HDMI Combo PHY	Power down different elements in the PHY following the sequence below. Set override for power down signals from register: SOR_NV_PDISP_SOR_PLL2_0_AUX2_OVERRIDE_POWERDOWN SOR_NV_PDISP_SOR_PLL2_0_AUX1_SEQ_PLLCAPPD_OVERRIDE Disable the Tx lanes, CAL, and wait for > 5 µs ▪ Set PDCAL to 1 through SOR reg: SOR_NV_PDISP_SOR_DP_PADCTL0_0_PAD_CAL_PD_POWERDOWN ▪ Set PD_TX* to 1 through SOR reg: SOR_NV_PDISP_SOR_DP_PADCTL0_0_PD_TXD_0..3_YES ▪ Set PDPORT to 1 through SOR reg: SOR_NV_PDISP_SOR_PLL2_0_AUX7_PORT_POWERDOWN_ENABLE Disable the PLL / charge-pump / VCO, and wait for > 5 µs. ▪ Set PLLCAPPD to 1 through SOR reg: SOR_NV_PDISP_SOR_PLL2_0_AUX8_SEQ_PLLCAPPD_ENFORCE_ENABLE ▪ Set PLLVCOPD to 1 through SOR reg: SOR_NV_PDISP_SOR_PLL2_0_VCOPD_YES ▪ Set PDPLL to 1 through SOR reg: SOR_NV_PDISP_SOR_PLL0_0_PWR_OFF Assert PDBG to disable bandgap, and wait for > 5 µs. ▪ Set PDBG to 1 through SOR reg: SOR_NV_PDISP_SOR_PLL2_0_AUX6_BANDGAP_POWERDOWN_ENABLE Assert DPD master power down control. ▪ Set E_DPD to 1 through PMC reg: PMC_IMPL_IO_DPD_REQ_0_HDMI_DP0..3_ON

PHY/BRICK	PAD Control Programming requirement when I/O is unused but I/O power supply is ON
DP AUX pads	Park the unused bricks in DPD/Power down state. ▪ Set DPAUX_HYBRID_SPARE_0_PAD_PWR_POWERDOWN to `1' corresponding to the DP-AUX channel.
PEX CLK out pads	Park the unused lanes in DPD mode. ▪ Set corresponding bit fields in PMC register PMC_IMPL_IO_DPD_REQ_0

8.4.3.5 DPD Sequencing

When certain interface is not required to be active, group of pads associated with the interface may be put in power down (DPD) state to save power. Based on MPIO pad capabilities, and functional requirements:

- All non-AO groups of MPIO pads enter in to DPD state as part of SC7 entry and exit DPD state as part of SC7 entry by PMC hardware.
- All AO groups of MPIO pads do not enter in to DPD state as part of SC7 entry and do exit DPD state as part of SC7 exit by PMC hardware.

Below is a reference guide for generic MPIO pad DPD sequencing during system active, PMC “Pad controls” sections software programming sequence shall be referred for precise programming steps. Steps outline for typical MPIO interface DPD entry and exit:

DPD Entry:

- Software: Bring the I/O interface to Bus IDLE state. I/O controllers are brought into IDLE state and subsequently the values they are driving towards the pads are stable.
- Software: Program the PMC_IMPL_SEL_DPD_TIM_0 register as per SEL_DPD and E_DPD timing requirements. The pad control inputs E_DPD and SEL_DPD are driven by PMC H/W meeting the timing requirement.
- Software: Set PMC PMC_IMPL_IO_DPD[DPD2/DPD7/DPD8]_REQ_0 register bit field(s) corresponding to the group of pads or interface. DPD controls and mechanisms lists all such PMC DPD software knobs.
- Hardware: PMC runs the state machine to drive the DPD sequence control pins to enter i in DPD state.
  - SEL_DPD is asserted which triggers the pad to latch the pad inputs.
  - E_DPD is asserted based on PMC_IMPL_SEL_DPD_TIM_0 configuration.
- Hardware: I/O state prior to entering to DPD is maintained while in DPD.

DPD Exit:

- Software: Enable clocks and initialize respective I/O controller.

- Software: Clear PMC PMC_IMPL_IO_DPD[DPD2/DPD7/DPD8] _REQ_0 register bit field(s) corresponding to the group of pads or interface. DPD controls and mechanisms lists all such PMC DPD software knobs.
- Hardware: PMC runs the state machine to drive the DPD sequence control pins to exit the DPD state.
  - SEL_DPD is de-asserted which triggers the pad to latch the pad inputs.
  - E_DPD is de-asserted based on PMC_IMPL_SEL_DPD_TIM_0 configuration.
- Software: Proceeds with I/O interface programming for required interface operation.

8.4.3.5.1 MPIO Pad DPD State Output Tweaking

In such cases the following sequence should be performed:

- SC7-entry-init: Configure the pad's GPIO specific registers for desired pin state.
- SC7-entry-init: Change the pin to GPIO (as on output).
- SC7-entry-init: Drive the desired value.
- SC7-entry-init: Trigger to DPD mode by setting bit, *i.e.* PMC_IMPL_DPDx_REQ in PMC to sample the correct value.
- Proceed with SC7-entry sequence.
- SC7-exit: Do the PinMux recovery.
- SC7-exit: Change the pin from GPIO to SFIO and associate with the respective controller.
- SC7-exit Ensure the controller is configured to drive correct value what is getting driven in the pad so that there is no glitch when the pad is coming out of DPD.
- SC7-exit: Bring the pads out of DPD.

PHY/Brick SC7 Entry/Exit

Most of the unused PHY/Bricks I/O states are parked to tri-state and hence they are put into Low-Power state during SC7 and brought out of power down modes during SC7 exit. Refer to respective HSIO/DDR architecture documents for the same.

Bringing I/O Rails Down During SC7

To realize lowest possible leakage power, it's possible to power down I/O rails in conjunction with Orin Low-Power states such as SC7. Orin I/O rails may be shared with the interfacing device and hence the possibility of powering off I/O rails goes case by case based on interface/platform.

Discrete Pad's I/O Rails in SC7

Normally the I/O rails of the discrete pads are ON during SC7 to support wakes etc. However, to aggressively reduce system power it may be turned OFF in specific platforms depending on use case requirements. I/O rail power on/off has special considerations with regards to power

sequencing and duly meeting MPIO pad functional requirements as listed in DPD controls handling and Low-Power state aspects.

Important note with regards to turning ON I/O rails:

- MPIO pads VDDP supply must not be powered ON while the pad is in DPD mode. Deep sleep exit turning ON I/O rails has modification in Orin. Reset is asserted to exit the DPD mode (rather to say ensure MPIO pad is not in DPD mode) before I/O rails are powered ON. See the Power Management Controller chapter for more details.

PHY/Bricks I/O Rail in SC7

Bricks are put in their respective Low-Power mode and the control mechanism differs for each brick. The Low-Power mode mechanisms of different bricks are captured in Low-Power Modes for PHYs as a reference. Respective software drivers for the brick like Display Driver, Camera Driver puts the bricks in Low-Power mode when interface is IDLE. Low-Power/DPD programming shall refer to respective HSIO/DDR architecture documents.

8.4.3.6 Wake Events Configuration

To enable wake triggering when wake capable pad is in active state, the E_INPUT pad control needs to be programmed HIGH to enable receiver path in pad.

However, such wake capable pad is in DPD state (E_DPD = 1, SEL_DPD = 1) enables receiver path to trigger wake irrespective of E_INPUT pad control setting. To avoid corner case scenarios of missing wake events during DPD and active states, it's recommended to set E_INPUT = 1 setting for wake capable pads. It is required to setup E_INPUT = 1 (Enable) by software post-boot for all the wake event supported pads relevant for a platform.

Most of the mechanical buttons act as SC7 wake source which means their key press to be de-bounced during SC7. Beyond which the wake events must be protected from noise spikes. So, wake sensing logic in PMC does glitch filtering and de-bounce capability. Wake programming consists of various signaling, routing and masking register configuration in PMC. The following is a highlight of wake signal conditioning capabilities are supported.

De-bounce selection:

- DEBOUNCE_CLK = 1 kHz: Any wake events width less than 31 µs is ignored.
- DEBOUNCE_CLK = 32 kHz: Any wake events width less than 31 µs is ignored.

De-glitch:

- If enabled, Glitches <= 2 ns are filtered out.

For more details. refer to the PMC section titled Wake Engine Programming.

8.4.4 G3 PAD Control Registers

NOTE:

The G3 PAD Control Registers are collectively called PADCTL_A0 in the System Address Map. Use this Block Name (PADCTL_A0) to find the Base Address of these registers in the System Address Map.

NOTE:

The binary “x” is turned to “0” for the Reset and PROD hex value. Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as “x” or “Undefined” actually means that the bit could be either “0” or “1” after Reset, whereas a bit PROD value as “x” or “Don’t-care” means that either “0” or “1” can be written to the bit by software Initialization.

PADCTL_G3_EXTPERIPH2_CLK_0
Offset: 0x0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_EXTPERIPH2_CLK_0
Reset: 0x00000414 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE

Bit	Reset	Description
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	EXTPERIPH2	**PM**: **0** = EXTPERIPH2 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G3_CFG2TMC_EXTPERIPH2_CLK_0

Offset: 0x4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_EXTPERIPH2_CLK_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G3_EXTPERIPH1_CLK_0

Offset: 0x8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_EXTPERIPH1_CLK_0
Reset: 0x00000414 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	EXTPERIPH1	**PM**: **0** = EXTPERIPH1 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G3_CFG2TMC_EXTPERIPH1_CLK_0

Offset: 0xc
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_EXTPERIPH1_CLK_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G3_CAM_I2C_SDA_0

Offset: 0x10
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_CAM_I2C_SDA_0
Reset: 0x00000570 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0111,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	I2C3	**PM**: **0** = I2C3 **1** = VI0 **2** = RSVD2 **3** = VI1

PADCTL_G3_CFG2TMC_CAM_I2C_SDA_0

Offset: 0x14
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_CAM_I2C_SDA_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G3_CAM_I2C_SCL_0

Offset: 0x18
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_CAM_I2C_SCL_0
Reset: 0x00001570 (0bxxxx,xxxx,xxxx,xxxx,xxx1,x1x1,0111,0000)

Bit	Reset	Description
12	ENABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	I2C3	**PM**: **0** = I2C3 **1** = VI0 **2** = VI0_ALT **3** = VI1

PADCTL_G3_CFG2TMC_CAM_I2C_SCL_0

Offset: 0x1c
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_CAM_I2C_SCL_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G3_SOC_GPIO23_0
Offset: 0x20
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_SOC_GPIO23_0
Reset: 0x00000474 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0111,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

Bit	Reset	Description
1:0	VI0	**PM**: **0** = VI0 **1** = VI0_ALT **2** = VI1 **3** = VI1_ALT

PADCTL_G3_CFG2TMC_SOC_GPIO23_0

Offset: 0x24
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_SOC_GPIO23_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G3_SOC_GPIO24_0

Offset: 0x28
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_SOC_GPIO24_0
Reset: 0x00000474 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0111,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	VI0	**PM**: **0** = VI0 **1** = SOC **2** = VI1 **3** = VI1_ALT

PADCTL_G3_CFG2TMC_SOC_GPIO24_0

Offset: 0x2c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_SOC_GPIO24_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G3_SOC_GPIO25_0

Offset: 0x30
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_SOC_GPIO25_0
Reset: 0x00000474 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0111,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	VI0	**PM**: **0** = VI0 **1** = I2S5 **2** = VI1 **3** = DMIC1

PADCTL_G3_CFG2TMC_SOC_GPIO25_0

Offset: 0x34
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_SOC_GPIO25_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G3_PWR_I2C_SCL_0

Offset: 0x38
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_PWR_I2C_SCL_0
Reset: 0x00001570 (0bxxxx,xxxx,xxxx,xxxx,xxx1,x1x1,0111,0000)

Bit	Reset	Description
12	ENABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	I2C5	**PM**: **0** = I2C5 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G3_CFG2TMC_PWR_I2C_SCL_0

Offset: 0x3c
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_PWR_I2C_SCL_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G3_PWR_I2C_SDA_0
Offset: 0x40
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_PWR_I2C_SDA_0
Reset: 0x00000570 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0111,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

Bit	Reset	Description
1:0	I2C5	**PM**: **0** = I2C5 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G3_CFG2TMC_PWR_I2C_SDA_0

Offset: 0x44
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_PWR_I2C_SDA_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G3_SOC_GPIO28_0

Offset: 0x48
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_SOC_GPIO28_0
Reset: 0x00000474 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0111,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	VI0	**PM**: **0** = VI0 **1** = RSVD1 **2** = VI1 **3** = RSVD3

PADCTL_G3_CFG2TMC_SOC_GPIO28_0

Offset: 0x4c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_SOC_GPIO28_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G3_SOC_GPIO29_0

Offset: 0x50
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_SOC_GPIO29_0
Reset: 0x00000474 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0111,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = NV **2** = RSVD2 **3** = RSVD3

PADCTL_G3_CFG2TMC_SOC_GPIO29_0

Offset: 0x54
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_SOC_GPIO29_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G3_SOC_GPIO30_0
Offset: 0x58
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_SOC_GPIO30_0
Reset: 0x00000470 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0111,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = WDT **2** = RSVD2 **3** = RSVD3

PADCTL_G3_CFG2TMC_SOC_GPIO30_0
Offset: 0x5c
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_SOC_GPIO30_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G3_SOC_GPIO31_0
Offset: 0x60
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_SOC_GPIO31_0
Reset: 0x00000474 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0111,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

Bit	Reset	Description
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G3_CFG2TMC_SOC_GPIO31_0

Offset: 0x64
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_SOC_GPIO31_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G3_SOC_GPIO32_0

Offset: 0x68
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_SOC_GPIO32_0
Reset: 0x00000474 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0111,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = EXTPERIPH3 **2** = DCB **3** = RSVD3

PADCTL_G3_CFG2TMC_SOC_GPIO32_0

Offset: 0x6c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_SOC_GPIO32_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G3_SOC_GPIO33_0

Offset: 0x70
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_SOC_GPIO33_0
Reset: 0x00000474 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0111,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = EXTPERIPH4 **2** = DCB **3** = RSVD3

PADCTL_G3_CFG2TMC_SOC_GPIO33_0

Offset: 0x74
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_SOC_GPIO33_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G3_SOC_GPIO35_0

Offset: 0x78
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_SOC_GPIO35_0
Reset: 0x00000574 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0111,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = I2S5 **2** = DMIC1 **3** = RSVD3

PADCTL_G3_CFG2TMC_SOC_GPIO35_0

Offset: 0x7c
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_SOC_GPIO35_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G3_SOC_GPIO37_0
Offset: 0x80
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_SOC_GPIO37_0
Reset: 0x00000574 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0111,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

Bit	Reset	Description
1:0	GP	**PM**: **0** = GP **1** = I2S5 **2** = DMIC4 **3** = DSPK1

PADCTL_G3_CFG2TMC_SOC_GPIO37_0

Offset: 0x84
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_SOC_GPIO37_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G3_SOC_GPIO56_0

Offset: 0x88
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_SOC_GPIO56_0
Reset: 0x00000434 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0011,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = I2S5 **2** = DMIC4 **3** = DSPK1

PADCTL_G3_CFG2TMC_SOC_GPIO56_0

Offset: 0x8c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_SOC_GPIO56_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G3_UART1_CTS_0

Offset: 0x90
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_UART1_CTS_0
Reset: 0x00000414 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	UARTA	**PM**: **0** = UARTA **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G3_CFG2TMC_UART1_CTS_0

Offset: 0x94
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_UART1_CTS_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G3_UART1_RTS_0

Offset: 0x98
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_UART1_RTS_0
Reset: 0x00000454 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,01x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	UARTA	**PM**: **0** = UARTA **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G3_CFG2TMC_UART1_RTS_0

Offset: 0x9c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_UART1_RTS_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G3_UART1_RX_0

Offset: 0xa0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_UART1_RX_0
Reset: 0x00000414 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	UARTA	**PM**: **0** = UARTA **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G3_CFG2TMC_UART1_RX_0

Offset: 0xa4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_UART1_RX_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G3_UART1_TX_0
Offset: 0xa8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_UART1_TX_0
Reset: 0x00000454 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,01x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

Bit	Reset	Description
1:0	UARTA	**PM**: **0** = UARTA **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G3_CFG2TMC_UART1_TX_0

Offset: 0xac
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_SCR_UART1_TX_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G3_EC_FEATURE_0

```
================================
 GLOBAL SLICE
================================
```

Offset: 0x400
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G3_SCR_EC_SCR_0
Reset: 0x00020001 (0b0000,0000,0000,0010,xxxx,xxxx,xx00,0001)

Bit	Reset	Description
31:16	0x2	**NUM_ERR**: Number of errors connected to this collator. This is passed as a build time option to the plugin
5:0	0x1	**NUM_ERR_SLICES**: Number of error slices supported by this error collator, does not include the GlobalSpace and is derived by ceil (NUM_ERR/32). SW shall first read this register to determine the number of slices and read the required number of Error_Status registers .

PADCTL_G3_EC_SWRESET_0

Offset: 0x404
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G3_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SWRST**: 1'b1 : Issue a SW reset to the Error Collator. This will reset all the registers(Except SCR), counters and logic of the Error Collator. SW can use this bit to flush errors logged into the error collator for ex, after Boot, SC7/8 exit. 1'b0 : Do nothing, reset value. This bit is auto-cleared.

PADCTL_G3_EC_MISSIONERR_TYPE_0

Offset: 0x408
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G3_SCR_EC_SCR_0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0101)

Bit	Reset	Description
5:0	0x5	**CODE**: This register indicates the fault code of the error line based on the value of MISSIONERR_INDEX Register. This can be used by a fault handling agent to triage an error without requiring device-specific code.The possible values of this field are: 6'd0 : None 6'd1 : Parity Error on internal data path 6'd2 : ECC SEC Error on internal data path 6'd3 : ECC DED Error on internal data path 6'd4 : Comparator Error 6'd5 : Register Parity Error 6'd6 : Parity Error from on-chip SRAM/Fifo 6'd7 : ECC SEC Error from on-chip SRAM/Fifo 6'd8 : ECC DED Error from on-chip SRAM/Fifo 6'd9 : Clock Monitor Error 6'd10 : Voltage Error 6'd11 : Temperature Error 6'd12 : CRC Error on internal data path 6'd16 : SW Corrected Error 6'd17 : SW Uncorrected Error 6'd18 : SW Generic Error 6'd20 : ECC SED Error on internal data path 6'd21 : ECC SED Error from on-chip SRAM/Fifo 6'd32 : Other HW Corrected Error 6'd33 : Other HW Uncorrected Error All other values : Reserved for future use.

PADCTL_G3_EC_CURRENT_COUNTER_VALUE_0

Offset: 0x40c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G3_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
8:0	0x0	**VALUE**: Provides the current value of the counter corresponding to the error in MissionErr_Index Register. Default provides the value of error 0 counter. Bit[8] is the ovrerflow bit post which the counter saturates and does not counter further.

PADCTL_G3_EC_MISSIONERR_INDEX_0

Offset: 0x414
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_G3_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**IDX**: BINARY Encoded. For error number 32, register should be programmed with value 0x20. Write to this register with Error number will update: - MISSIONERR_TYPE Register with the Error-Code for the Error. - CURRENT_COUNTER_VALUE Register with the error's SEC/DED Counter. - MISSIONERR_USERVALUE with value of the first error_<i>_user signal. SW can use this to trage the error. number shall update the MISSIONERR_TYPE register with the error code and the Current_Counter_Value register with the value of the errors SEC/DED counter. SW can use this register to triage the error.

PADCTL_G3_EC_CORRECTABLE_THRESHOLD_0

Offset: 0x418
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**COUNT**: Threshold value for all SEC Fault Reporting Units connected to this error collator. SEC Errors are logged once the threshold is reached and the overflow bit is set. 7'b0 : Log SEC error after receiving 1 Error. 7'b1 : Log SEC error after receiving 2 Errors. ... 7'bFF : Log SEC error after receiving 256 Errors.

PADCTL_G3_EC_MISSIONERR_INJECT_UNLOCK_0

Offset: 0x41c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G3_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Writes to ERRSLICE_XXX_MISSIONERR_INJECT registers are disabled until this register is written with a value of 0xE1. This is to prevent an inadvertent safety error injection in the safety plugin due to: 1. A fault on ERRSLICE_XXX_MISSIONERR_INJECT register itself. 2. Erroneous SW. The register shall be written with a value of 0x0 to reestablish the lock after user has completed the error injection testing. 0xE1 : Unlock the MISSIONERR_INJECT Register 0x0 : Lock the MISSIONERR_INJECT Register **0** = LOCK **225** = UNLOCK

PADCTL_G3_EC_ERRSLICE0_MISSIONERR_ENABLE_0

```
==================================
 ERROR SLICE - 0
==================================
```

Offset: 0x430
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.G3_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.G3_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.G3_PADCTLREG_reg 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.G3_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL_G3_err_collator 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL_G3_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_G3_EC_ERRSLICE0_MISSIONERR_FORCE_0

Offset: 0x434
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G3_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.G3_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.G3_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL_G3_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_G3_EC_ERRSLICE0_MISSIONERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of MissionError_Enable register, to avoid silent dropping of errors.
Offset: 0x438
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G3_PADCTLREG_reg_scr_wrap was equal to 2'b10. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G3_PADCTLREG_reg_scr_wrap was equal to 2'b01.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G3_PADCTLREG_reg was equal to 2'b10. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G3_PADCTLREG_reg was equal to 2'b01.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G3_err_collator was equal to 2'b10. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G3_err_collator was equal to 2'b01.

PADCTL_G3_EC_ERRSLICE0_MISSIONERR_INJECT_0

Offset: 0x43c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G3_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Assert the inject_error_2 output for Register Parity Error to PADCTL.G3_PADCTLREG_reg_scr_wrap to allow for error injection. 1'b0 -> De-Asseert inject_error_2 output. **0** = DISABLE **1** = ENABLE
1	0x0	**ERR1**: 1'b1 -> Assert the inject_error_1 output for Register Parity Error to PADCTL.G3_PADCTLREG_reg to allow for error injection. 1'b0 -> De-Asseert inject_error_1 output. **0** = DISABLE **1** = ENABLE
0	0x0	**ERR0**: 1'b1 -> Assert the inject_error_0 output for Register Parity Error to PADCTL_G3_err_collator to allow for error injection. 1'b0 -> De-Asseert inject_error_0 output. **0** = DISABLE **1** = ENABLE

PADCTL_G3_EC_ERRSLICE0_LATENTERR_ENABLE_0

Offset: 0x440
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.G3_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.G3_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.G3_PADCTLREG_reg 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.G3_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL_G3_err_collator 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL_G3_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_G3_EC_ERRSLICE0_LATENTERR_FORCE_0

Offset: 0x444
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G3_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.G3_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.G3_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL_G3_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_G3_EC_ERRSLICE0_LATENTERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of LatentError_Enable register, to avoid silent dropping of errors.
Offset: 0x448
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G3_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G3_PADCTLREG_reg_scr_wrap was equal to 2'b00 or 2'b11. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G3_PADCTLREG_reg_scr_wrap was equal to 2'b01 or 2'b10, but no latent error.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G3_PADCTLREG_reg was equal to 2'b00 or 2'b11. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G3_PADCTLREG_reg was equal to 2'b01 or 2'b10, but no latent error.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G3_err_collator was equal to 2'b00 or 2'b11. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G3_err_collator was equal to 2'b01 or 2'b10, but no latent error.

PADCTL_G3_EC_ERRSLICE0_COUNTER_RELOAD_0

Offset: 0x450

Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G3_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.G3_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
1	0x0	**ERR1**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.G3_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
0	0x0	**ERR0**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL_G3_err_collator 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

8.4.5 G5 PAD Control Registers

NOTE:

The binary “x” is turned to “0” for the Reset and PROD hex value.
Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as “x” or “Undefined” actually means that the bit could be either “0” or “1” after Reset, whereas a bit PROD value as “x” or “Don’t-care” means that either “0” or “1” can be written to the bit by software Initialization.

PADCTL_G5_DAP4_SCLK_0
Offset: 0x0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G5_SCR_SCR_DAP4_SCLK_0
Reset: 0x00001414 (0bxxxx,xxxx,xxxx,xxxx,xxx1,x1x0,00x1,0100)

Bit	Reset	Description
12	ENABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	I2S4	**PM**: **0** = I2S4 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G5_CFG2TMC_DAP4_SCLK_0

Offset: 0x4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G5_SCR_SCR_DAP4_SCLK_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G5_DAP4_DOUT_0

Offset: 0x8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G5_SCR_SCR_DAP4_DOUT_0
Reset: 0x00000414 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	I2S4	**PM**: **0** = I2S4 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G5_CFG2TMC_DAP4_DOUT_0

Offset: 0xc
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G5_SCR_SCR_DAP4_DOUT_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G5_DAP4_DIN_0

Offset: 0x10
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G5_SCR_SCR_DAP4_DIN_0
Reset: 0x00000414 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	I2S4	**PM**: **0** = I2S4 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G5_CFG2TMC_DAP4_DIN_0

Offset: 0x14
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G5_SCR_SCR_DAP4_DIN_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G5_DAP4_FS_0
Offset: 0x18
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G5_SCR_SCR_DAP4_FS_0
Reset: 0x00000414 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

Bit	Reset	Description
1:0	I2S4	**PM**: **0** = I2S4 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G5_CFG2TMC_DAP4_FS_0

Offset: 0x1c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G5_SCR_SCR_DAP4_FS_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G5_DAP6_SCLK_0

Offset: 0x20
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G5_SCR_SCR_DAP6_SCLK_0
Reset: 0x00001414 (0bxxxx,xxxx,xxxx,xxxx,xxx1,x1x0,00x1,0100)

Bit	Reset	Description
12	ENABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	I2S6	**PM**: **0** = I2S6 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G5_CFG2TMC_DAP6_SCLK_0

Offset: 0x24
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G5_SCR_SCR_DAP6_SCLK_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G5_DAP6_DOUT_0

Offset: 0x28
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G5_SCR_SCR_DAP6_DOUT_0
Reset: 0x00000414 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	I2S6	**PM**: **0** = I2S6 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G5_CFG2TMC_DAP6_DOUT_0

Offset: 0x2c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G5_SCR_SCR_DAP6_DOUT_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G5_DAP6_DIN_0

Offset: 0x30
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: PADCTL_G5_SCR_SCR_DAP6_DIN_0
Reset: 0x00000414 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	I2S6	**PM**: **0** = I2S6 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G5_CFG2TMC_DAP6_DIN_0

Offset: 0x34
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G5_SCR_SCR_DAP6_DIN_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G5_DAP6_FS_0

Offset: 0x38
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G5_SCR_SCR_DAP6_FS_0
Reset: 0x00000414 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	I2S6	**PM**: **0** = I2S6 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G5_CFG2TMC_DAP6_FS_0

Offset: 0x3c
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_G5_SCR_SCR_DAP6_FS_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G5_EC_FEATURE_0

Offset: 0x400
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G5_SCR_EC_SCR_0
Reset: 0x00020001 (0b0000,0000,0000,0010,xxxx,xxxx,xx00,0001)

Bit	Reset	Description
31:16	0x2	**NUM_ERR**: Number of errors connected to this collator. This is passed as a build time option to the plugin
5:0	0x1	**NUM_ERR_SLICES**: Number of error slices supported by this error collator, does not include the GlobalSpace and is derived by ceil (NUM_ERR/32). SW shall first read this register to determine the number of slices and read the required number of Error_Status registers .

PADCTL_G5_EC_SWRESET_0

Offset: 0x404
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G5_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SWRST**: 1'b1 : Issue a SW reset to the Error Collator. This will reset all the registers(Except SCR), counters and logic of the Error Collator. SW can use this bit to flush errors logged into the error collator for ex, after Boot, SC7/8 exit. 1'b0 : Do nothing, reset value. This bit is auto-cleared.

PADCTL_G5_EC_MISSIONERR_TYPE_0

Offset: 0x408
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G5_SCR_EC_SCR_0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0101)

Bit	Reset	Description
5:0	0x5	**CODE**: This register indicates the fault code of the error line based on the value of MISSIONERR_INDEX Register. This can be used by a fault handling agent to triage an error without requiring device-specific code.The possible values of this field are: 6'd0 : None 6'd1 : Parity Error on internal data path 6'd2 : ECC SEC Error on internal data path 6'd3 : ECC DED Error on internal data path 6'd4 : Comparator Error 6'd5 : Register Parity Error 6'd6 : Parity Error from on-chip SRAM/Fifo 6'd7 : ECC SEC Error from on-chip SRAM/Fifo 6'd8 : ECC DED Error from on-chip SRAM/Fifo 6'd9 : Clock Monitor Error 6'd10 : Voltage Error 6'd11 : Temperature Error 6'd12 : CRC Error on internal data path 6'd16 : SW Corrected Error 6'd17 : SW Uncorrected Error 6'd18 : SW Generic Error 6'd20 : ECC SED Error on internal data path 6'd21 : ECC SED Error from on-chip SRAM/Fifo 6'd32 : Other HW Corrected Error 6'd33 : Other HW Uncorrected Error All other values : Reserved for future use.

PADCTL_G5_EC_CURRENT_COUNTER_VALUE_0

Offset: 0x40c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G5_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
8:0	0x0	**VALUE**: Provides the current value of the counter corresponding to the error in MissionErr_Index Register. Default provides the value of error 0 counter. Bit[8] is the ovrerflow bit post which the counter saturates and does not counter further.

PADCTL_G5_EC_MISSIONERR_INDEX_0

Offset: 0x414
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G5_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**IDX**: BINARY Encoded. For error number 32, register should be programmed with value 0x20. Write to this register with Error number will update: - MISSIONERR_TYPE Register with the Error-Code for the Error. - CURRENT_COUNTER_VALUE Register with the error's SEC/DED Counter. - MISSIONERR_USERVALUE with value of the first error_<i>_user signal. SW can use this to trage the error. number shall update the MISSIONERR_TYPE register with the error code and the Current_Counter_Value register with the value of the errors SEC/DED counter. SW can use this register to triage the error.

PADCTL_G5_EC_CORRECTABLE_THRESHOLD_0

Offset: 0x418
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G5_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**COUNT**: Threshold value for all SEC Fault Reporting Units connected to this error collator. SEC Errors are logged once the threshold is reached and the overflow bit is set. 7'b0 : Log SEC error after receiving 1 Error. 7'b1 : Log SEC error after receiving 2 Errors. ... 7'bFF : Log SEC error after receiving 256 Errors.

PADCTL_G5_EC_MISSIONERR_INJECT_UNLOCK_0

Offset: 0x41c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G5_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Writes to ERRSLICE_XXX_MISSIONERR_INJECT registers are disabled until this register is written with a value of 0xE1. This is to prevent an inadvertent safety error injection in the safety plugin due to: 1. A fault on ERRSLICE_XXX_MISSIONERR_INJECT register itself. 2. Erroneous SW. The register shall be written with a value of 0x0 to reestablish the lock after user has completed the error injection testing. 0xE1 : Unlock the MISSIONERR_INJECT Register 0x0 : Lock the MISSIONERR_INJECT Register **0** = LOCK **225** = UNLOCK

PADCTL_G5_EC_ERRSLICE0_MISSIONERR_ENABLE_0

Offset: 0x430
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G5_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.G5_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.G5_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.G5_PADCTLREG_reg 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.G5_PADCTLREG_reg **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
0	0x1	**ERR0**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL_G5_err_collator 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL_G5_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_G5_EC_ERRSLICE0_MISSIONERR_FORCE_0

Offset: 0x434
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G5_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.G5_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.G5_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL_G5_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_G5_EC_ERRSLICE0_MISSIONERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of MissionError_Enable register, to avoid silent dropping of errors.

Offset: 0x438
Read/Write: R/W

Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G5_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G5_PADCTLREG_reg_scr_wrap was equal to 2'b10. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G5_PADCTLREG_reg_scr_wrap was equal to 2'b01.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G5_PADCTLREG_reg was equal to 2'b10. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G5_PADCTLREG_reg was equal to 2'b01.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G5_err_collator was equal to 2'b10. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G5_err_collator was equal to 2'b01.

PADCTL_G5_EC_ERRSLICE0_MISSIONERR_INJECT_0

Offset: 0x43c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G5_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Assert the inject_error_2 output for Register Parity Error to PADCTL.G5_PADCTLREG_reg_scr_wrap to allow for error injection. 1'b0 -> De-Asseert inject_error_2 output. **0** = DISABLE **1** = ENABLE
1	0x0	**ERR1**: 1'b1 -> Assert the inject_error_1 output for Register Parity Error to PADCTL.G5_PADCTLREG_reg to allow for error injection. 1'b0 -> De-Asseert inject_error_1 output. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
0	0x0	**ERR0**: 1'b1 -> Assert the inject_error_0 output for Register Parity Error to PADCTL_G5_err_collator to allow for error injection. 1'b0 -> De-Asseert inject_error_0 output. **0** = DISABLE **1** = ENABLE

PADCTL_G5_EC_ERRSLICE0_LATENTERR_ENABLE_0

Offset: 0x440
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G5_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.G5_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.G5_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.G5_PADCTLREG_reg 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.G5_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL_G5_err_collator 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL_G5_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_G5_EC_ERRSLICE0_LATENTERR_FORCE_0

Offset: 0x444
Read/Write: WO
Parity Protection: N
Shadow: N

SCR Protection: PADCTL_G5_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.G5_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.G5_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL_G5_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_G5_EC_ERRSLICE0_LATENTERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of LatentError_Enable register, to avoid silent dropping of errors.

Offset: 0x448
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G5_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G5_PADCTLREG_reg_scr_wrap was equal to 2'b00 or 2'b11. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G5_PADCTLREG_reg_scr_wrap was equal to 2'b01 or 2'b10, but no latent error.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G5_PADCTLREG_reg was equal to 2'b00 or 2'b11. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G5_PADCTLREG_reg was equal to 2'b01 or 2'b10, but no latent error.

Bit	Reset	Description
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G5_err_collator was equal to 2'b00 or 2'b11. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G5_err_collator was equal to 2'b01 or 2'b10, but no latent error.

PADCTL_G5_EC_ERRSLICE0_COUNTER_RELOAD_0
Offset: 0x450
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G5_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.G5_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
1	0x0	**ERR1**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.G5_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
0	0x0	**ERR0**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL_G5_err_collator 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

8.4.6 G4 PAD Control Registers

NOTE:

The G4 PAD Control Registers are collectively called PADCTL_A4 in the System Address Map.

The binary “x” is turned to “0” for the Reset and PROD hex value. Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as “x” or “Undefined” actually means that the bit could be either “0” or “1” after Reset, whereas a bit

PROD value as “x” or “Don’t-care” means that either “0” or “1” can be written to the bit by software Initialization.

PADCTL_G4_CPU_PWR_REQ_0

Offset: 0x0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_CPU_PWR_REQ_0
Reset: 0x00000400 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x0,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G4_CFG2TMC_CPU_PWR_REQ_0

Offset: 0x4
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_G4_SCR_SCR_CPU_PWR_REQ_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G4_UART4_CTS_0
Offset: 0x8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_UART4_CTS_0
Reset: 0x00000458 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,01x1,1000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_UP	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	UARTD	**PM**: **0** = UARTD **1** = RSVD1 **2** = I2S7 **3** = RSVD3

PADCTL_G4_CFG2TMC_UART4_CTS_0

Offset: 0xc
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_UART4_CTS_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G4_UART4_RTS_0

Offset: 0x10
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_UART4_RTS_0
Reset: 0x00000454 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,01x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE

Bit	Reset	Description
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	UARTD	**PM**: **0** = UARTD **1** = SPI4 **2** = RSVD2 **3** = RSVD3

PADCTL_G4_CFG2TMC_UART4_RTS_0

Offset: 0x14
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_UART4_RTS_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G4_UART4_RX_0

Offset: 0x18
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_UART4_RX_0
Reset: 0x00000454 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,01x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	UARTD	**PM**: **0** = UARTD **1** = RSVD1 **2** = I2S7 **3** = RSVD3

PADCTL_G4_CFG2TMC_UART4_RX_0

Offset: 0x1c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_UART4_RX_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G4_UART4_TX_0

Offset: 0x20
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_UART4_TX_0
Reset: 0x00000454 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,01x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	UARTD	**PM**: **0** = UARTD **1** = SPI4 **2** = RSVD2 **3** = RSVD3

PADCTL_G4_CFG2TMC_UART4_TX_0

Offset: 0x24
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_UART4_TX_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G4_GEN1_I2C_SCL_0

Offset: 0x28
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_GEN1_I2C_SCL_0
Reset: 0x00001570 (0bxxxx,xxxx,xxxx,xxxx,xxx1,x1x1,0111,0000)

Bit	Reset	Description
12	ENABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	I2C1	**PM**: **0** = I2C1 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G4_CFG2TMC_GEN1_I2C_SCL_0

Offset: 0x2c
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_G4_SCR_SCR_GEN1_I2C_SCL_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G4_GEN1_I2C_SDA_0
Offset: 0x30
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_GEN1_I2C_SDA_0
Reset: 0x00000570 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0111,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

Bit	Reset	Description
1:0	I2C1	**PM**: **0** = I2C1 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G4_CFG2TMC_GEN1_I2C_SDA_0

Offset: 0x34
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_GEN1_I2C_SDA_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G4_SOC_GPIO20_0

Offset: 0x38
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO20_0
Reset: 0x00000454 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,01x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = SDMMC1 **2** = RSVD2 **3** = RSVD3

PADCTL_G4_CFG2TMC_SOC_GPIO20_0

Offset: 0x3c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO20_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G4_SOC_GPIO21_0

Offset: 0x40
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO21_0
Reset: 0x00000410 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = GP **2** = I2S7 **3** = RSVD3

PADCTL_G4_CFG2TMC_SOC_GPIO21_0

Offset: 0x44
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO21_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G4_SOC_GPIO22_0

Offset: 0x48
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO22_0
Reset: 0x00000454 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,01x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = RSVD1 **2** = I2S7 **3** = RSVD3

PADCTL_G4_CFG2TMC_SOC_GPIO22_0

Offset: 0x4c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO22_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G4_SOC_GPIO13_0

Offset: 0x50
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO13_0
Reset: 0x00000458 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,01x1,1000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_UP	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G4_CFG2TMC_SOC_GPIO13_0

Offset: 0x54
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO13_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G4_SOC_GPIO14_0
Offset: 0x58
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO14_0
Reset: 0x00000458 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,01x1,1000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_UP	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = SPI4 **2** = RSVD2 **3** = RSVD3

PADCTL_G4_CFG2TMC_SOC_GPIO14_0

Offset: 0x5c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO14_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G4_SOC_GPIO15_0

Offset: 0x60
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO15_0
Reset: 0x00001454 (0bxxxx,xxxx,xxxx,xxxx,xxx1,x1x0,01x1,0100)

Bit	Reset	Description
12	ENABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE

Bit	Reset	Description
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = SPI4 **2** = RSVD2 **3** = RSVD3

PADCTL_G4_CFG2TMC_SOC_GPIO15_0

Offset: 0x64
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO15_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G4_SOC_GPIO16_0

Offset: 0x68
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO16_0
Reset: 0x00000458 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,01x1,1000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_UP	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = SPI4 **2** = RSVD2 **3** = RSVD3

PADCTL_G4_CFG2TMC_SOC_GPIO16_0

Offset: 0x6c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO16_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G4_SOC_GPIO17_0

Offset: 0x70
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO17_0
Reset: 0x00000454 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,01x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = CCLA **2** = RSVD2 **3** = RSVD3

PADCTL_G4_CFG2TMC_SOC_GPIO17_0

Offset: 0x74
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO17_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G4_SOC_GPIO18_0

Offset: 0x78
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO18_0
Reset: 0x00000454 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,01x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G4_CFG2TMC_SOC_GPIO18_0

Offset: 0x7c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO18_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G4_SOC_GPIO19_0

Offset: 0x80
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO19_0
Reset: 0x00000410 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	GP	**PM**: **0** = GP **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G4_CFG2TMC_SOC_GPIO19_0

Offset: 0x84
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO19_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G4_SOC_GPIO41_0

Offset: 0x88
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO41_0
Reset: 0x00001454 (0bxxxx,xxxx,xxxx,xxxx,xxx1,x1x0,01x1,0100)

Bit	Reset	Description
12	ENABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

Bit	Reset	Description
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = I2S2 **2** = RSVD2 **3** = RSVD3

PADCTL_G4_CFG2TMC_SOC_GPIO41_0

Offset: 0x8c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO41_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G4_SOC_GPIO42_0

Offset: 0x90
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO42_0
Reset: 0x00000414 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = I2S2 **2** = RSVD2 **3** = RSVD3

PADCTL_G4_CFG2TMC_SOC_GPIO42_0

Offset: 0x94
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO42_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G4_SOC_GPIO43_0

Offset: 0x98
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO43_0
Reset: 0x00000414 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = I2S2 **2** = RSVD2 **3** = RSVD3

PADCTL_G4_CFG2TMC_SOC_GPIO43_0

Offset: 0x9c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO43_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G4_SOC_GPIO44_0

Offset: 0xa0
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO44_0
Reset: 0x00000414 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = I2S2 **2** = RSVD2 **3** = RSVD3

PADCTL_G4_CFG2TMC_SOC_GPIO44_0

Offset: 0xa4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO44_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G4_SOC_GPIO06_0

Offset: 0xa8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO06_0
Reset: 0x00000454 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,01x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G4_CFG2TMC_SOC_GPIO06_0

Offset: 0xac
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO06_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G4_SOC_GPIO07_0
Offset: 0xb0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO07_0
Reset: 0x00000410 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	GP	**PM**: **0** = GP **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G4_CFG2TMC_SOC_GPIO07_0

Offset: 0xb4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_SCR_SOC_GPIO07_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G4_EC_FEATURE_0

```
==================================
 GLOBAL SLICE
==================================
```

Offset: 0x400
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G4_SCR_EC_SCR_0
Reset: 0x00040001 (0b0000,0000,0000,0100,xxxx,xxxx,xx00,0001)

Bit	Reset	Description
31:16	0x4	**NUM_ERR**: Number of errors connected to this collator. This is passed as a build time option to the plugin
5:0	0x1	**NUM_ERR_SLICES**: Number of error slices supported by this error collator, does not include the GlobalSpace and is derived by ceil (NUM_ERR/32). SW shall first read this register to determine the number of slices and read the required number of Error_Status registers .

PADCTL_G4_EC_SWRESET_0

Offset: 0x404
Read/Write: WO
Parity Protection: N
Shadow: N

SCR Protection: PADCTL_G4_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SWRST**: 1'b1 : Issue a SW reset to the Error Collator. This will reset all the registers(Except SCR), counters and logic of the Error Collator. SW can use this bit to flush errors logged into the error collator for ex, after Boot, SC7/8 exit. 1'b0 : Do nothing, reset value. This bit is auto-cleared.

PADCTL_G4_EC_MISSIONERR_TYPE_0

Offset: 0x408
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G4_SCR_EC_SCR_0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0101)

Bit	Reset	Description
5:0	0x5	**CODE**: This register indicates the fault code of the error line based on the value of MISSIONERR_INDEX Register. This can be used by a fault handling agent to triage an error without requiring device-specific code.The possible values of this field are: 6'd0 : None 6'd1 : Parity Error on internal data path 6'd2 : ECC SEC Error on internal data path 6'd3 : ECC DED Error on internal data path 6'd4 : Comparator Error 6'd5 : Register Parity Error 6'd6 : Parity Error from on-chip SRAM/Fifo 6'd7 : ECC SEC Error from on-chip SRAM/Fifo 6'd8 : ECC DED Error from on-chip SRAM/Fifo 6'd9 : Clock Monitor Error 6'd10 : Voltage Error 6'd11 : Temperature Error 6'd12 : CRC Error on internal data path 6'd16 : SW Corrected Error 6'd17 : SW Uncorrected Error 6'd18 : SW Generic Error 6'd20 : ECC SED Error on internal data path 6'd21 : ECC SED Error from on-chip SRAM/Fifo 6'd32 : Other HW Corrected Error 6'd33 : Other HW Uncorrected Error All other values : Reserved for future use.

PADCTL_G4_EC_CURRENT_COUNTER_VALUE_0

Offset: 0x40c

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G4_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
8:0	0x0	**VALUE**: Provides the current value of the counter corresponding to the error in MissionErr_Index Register. Default provides the value of error 0 counter. Bit[8] is the ovrerflow bit post which the counter saturates and does not counter further.

PADCTL_G4_EC_MISSIONERR_INDEX_0

Offset: 0x414
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	0x0	**IDX**: BINARY Encoded. For error number 32, register should be programmed with value 0x20. Write to this register with Error number will update: - MISSIONERR_TYPE Register with the Error-Code for the Error. - CURRENT_COUNTER_VALUE Register with the error's SEC/DED Counter. - MISSIONERR_USERVALUE with value of the first error_<i>_user signal. SW can use this to trage the error. number shall update the MISSIONERR_TYPE register with the error code and the Current_Counter_Value register with the value of the errors SEC/DED counter. SW can use this register to triage the error.

PADCTL_G4_EC_CORRECTABLE_THRESHOLD_0

Offset: 0x418
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**COUNT**: Threshold value for all SEC Fault Reporting Units connected to this error collator. SEC Errors are logged once the threshold is reached and the overflow bit is set. 7'b0 : Log SEC error after receiving 1 Error. 7'b1 : Log SEC error after receiving 2 Errors. ... 7'bFF : Log SEC error after receiving 256 Errors.

PADCTL_G4_EC_MISSIONERR_INJECT_UNLOCK_0

Offset: 0x41c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G4_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Writes to ERRSLICE_XXX_MISSIONERR_INJECT registers are disabled until this register is written with a value of 0xE1. This is to prevent an inadvertent safety error injection in the safety plugin due to: 1. A fault on ERRSLICE_XXX_MISSIONERR_INJECT register itself. 2. Erroneous SW. The register shall be written with a value of 0x0 to reestablish the lock after user has completed the error injection testing. 0xE1 : Unlock the MISSIONERR_INJECT Register 0x0 : Lock the MISSIONERR_INJECT Register **0** = LOCK **225** = UNLOCK

```
 0x20 to 0x2C = Reserved for future use.
```

PADCTL_G4_EC_ERRSLICE0_MISSIONERR_ENABLE_0

```
==================================
 ERROR SLICE - 0
==================================
```

Offset: 0x430
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_EC_SCR_0
Reset: 0x0000001f (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1,1111)

Bit	Reset	Description
4	0x1	**ERR4**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.G4_PADCTLREG_reg_l1_scr_wrap 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.G4_PADCTLREG_reg_l1_scr_wrap **0** = DISABLE **1** = ENABLE
3	0x1	**ERR3**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.G4_PADCTLREG_reg_l1 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.G4_PADCTLREG_reg_l1 **0** = DISABLE **1** = ENABLE
2	0x1	**ERR2**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.G4_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.G4_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.G4_PADCTLREG_reg 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.G4_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL_G4_err_collator 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL_G4_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_G4_EC_ERRSLICE0_MISSIONERR_FORCE_0

Offset: 0x434
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G4_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**ERR4**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.G4_PADCTLREG_reg_l1_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
3	0x0	**ERR3**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.G4_PADCTLREG_reg_l1 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.G4_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.G4_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL_G4_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_G4_EC_ERRSLICE0_MISSIONERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of MissionError_Enable register, to avoid silent dropping of errors.
Offset: 0x438
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**ERR4**: 1'b1 -> Error_4_pulse[1:0] for Register Parity Error from PADCTL.G4_PADCTLREG_reg_l1_scr_wrap was equal to 2'b10. 1'b0 -> Error_4_pulse[1:0] for Register Parity Error from PADCTL.G4_PADCTLREG_reg_l1_scr_wrap was equal to 2'b01.
3	0x0	**ERR3**: 1'b1 -> Error_3_pulse[1:0] for Register Parity Error from PADCTL.G4_PADCTLREG_reg_l1 was equal to 2'b10. 1'b0 -> Error_3_pulse[1:0] for Register Parity Error from PADCTL.G4_PADCTLREG_reg_l1 was equal to 2'b01.
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G4_PADCTLREG_reg_scr_wrap was equal to 2'b10. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G4_PADCTLREG_reg_scr_wrap was equal to 2'b01.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G4_PADCTLREG_reg was equal to 2'b10. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G4_PADCTLREG_reg was equal to 2'b01.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G4_err_collator was equal to 2'b10. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G4_err_collator was equal to 2'b01.

PADCTL_G4_EC_ERRSLICE0_MISSIONERR_INJECT_0

Offset: 0x43c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G4_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**ERR4**: 1'b1 -> Assert the inject_error_4 output for Register Parity Error to PADCTL.G4_PADCTLREG_reg_l1_scr_wrap to allow for error injection. 1'b0 -> De-Asseert inject_error_4 output. **0** = DISABLE **1** = ENABLE
3	0x0	**ERR3**: 1'b1 -> Assert the inject_error_3 output for Register Parity Error to PADCTL.G4_PADCTLREG_reg_l1 to allow for error injection. 1'b0 -> De-Asseert inject_error_3 output. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Assert the inject_error_2 output for Register Parity Error to PADCTL.G4_PADCTLREG_reg_scr_wrap to allow for error injection. 1'b0 -> De-Asseert inject_error_2 output. **0** = DISABLE **1** = ENABLE
1	0x0	**ERR1**: 1'b1 -> Assert the inject_error_1 output for Register Parity Error to PADCTL.G4_PADCTLREG_reg to allow for error injection. 1'b0 -> De-Asseert inject_error_1 output. **0** = DISABLE **1** = ENABLE
0	0x0	**ERR0**: 1'b1 -> Assert the inject_error_0 output for Register Parity Error to PADCTL_G4_err_collator to allow for error injection. 1'b0 -> De-Asseert inject_error_0 output. **0** = DISABLE **1** = ENABLE

PADCTL_G4_EC_ERRSLICE0_LATENTERR_ENABLE_0

Offset: 0x440
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_EC_SCR_0
Reset: 0x0000001f (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1,1111)

Bit	Reset	Description
4	0x1	**ERR4**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.G4_PADCTLREG_reg_l1_scr_wrap 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.G4_PADCTLREG_reg_l1_scr_wrap **0** = DISABLE **1** = ENABLE
3	0x1	**ERR3**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.G4_PADCTLREG_reg_l1 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.G4_PADCTLREG_reg_l1 **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.G4_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.G4_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.G4_PADCTLREG_reg 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.G4_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL_G4_err_collator 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL_G4_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_G4_EC_ERRSLICE0_LATENTERR_FORCE_0

Offset: 0x444
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G4_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**ERR4**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.G4_PADCTLREG_reg_l1_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
3	0x0	**ERR3**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.G4_PADCTLREG_reg_l1 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.G4_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.G4_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL_G4_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_G4_EC_ERRSLICE0_LATENTERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of LatentError_Enable register, to avoid silent dropping of errors.
Offset: 0x448
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G4_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**ERR4**: 1'b1 -> Error_4_pulse[1:0] for Register Parity Error from PADCTL.G4_PADCTLREG_reg_l1_scr_wrap was equal to 2'b00 or 2'b11. 1'b0 -> Error_4_pulse[1:0] for Register Parity Error from PADCTL.G4_PADCTLREG_reg_l1_scr_wrap was equal to 2'b01 or 2'b10, but no latent error.
3	0x0	**ERR3**: 1'b1 -> Error_3_pulse[1:0] for Register Parity Error from PADCTL.G4_PADCTLREG_reg_l1 was equal to 2'b00 or 2'b11. 1'b0 -> Error_3_pulse[1:0] for Register Parity Error from PADCTL.G4_PADCTLREG_reg_l1 was equal to 2'b01 or 2'b10, but no latent error.

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G4_PADCTLREG_reg_scr_wrap was equal to 2'b00 or 2'b11. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G4_PADCTLREG_reg_scr_wrap was equal to 2'b01 or 2'b10, but no latent error.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G4_PADCTLREG_reg was equal to 2'b00 or 2'b11. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G4_PADCTLREG_reg was equal to 2'b01 or 2'b10, but no latent error.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G4_err_collator was equal to 2'b00 or 2'b11. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G4_err_collator was equal to 2'b01 or 2'b10, but no latent error.

PADCTL_G4_EC_ERRSLICE0_COUNTER_RELOAD_0

Offset: 0x450
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G4_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**ERR4**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.G4_PADCTLREG_reg_l1_scr_wrap 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
3	0x0	**ERR3**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.G4_PADCTLREG_reg_l1 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
2	0x0	**ERR2**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.G4_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

Bit	Reset	Description
1	0x0	**ERR1**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.G4_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
0	0x0	**ERR0**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL_G4_err_collator 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

8.4.7 Debug PAD Control Registers

NOTE:

The Debug PAD Control Registers are collectively called PADCTL_A5 in the System Address Map.

The binary “x” is turned to “0” for the Reset and PROD hex value. Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as “x” or “Undefined” actually means that the bit could be either “0” or “1” after Reset, whereas a bit PROD value as “x” or “Don’t-care” means that either “0” or “1” can be written to the bit by software Initialization.

PADCTL_DEBUG_SOC_GPIO08_0
Offset: 0x8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_DEBUG_SCR_SCR_SOC_GPIO08_0
Reset: 0x00000414 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_DEBUG_CFG2TMC_SOC_GPIO08_0

Offset: 0xc
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_DEBUG_SCR_SCR_SOC_GPIO08_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_DEBUG_EC_FEATURE_0

```
==================================
 GLOBAL SLICE
==================================
```

Offset: 0x400
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: PADCTL_DEBUG_SCR_EC_SCR_0
Reset: 0x00020001 (0b0000,0000,0000,0010,xxxx,xxxx,xx00,0001)

Bit	Reset	Description
31:16	0x2	**NUM_ERR**: Number of errors connected to this collator. This is passed as a build time option to the plugin
5:0	0x1	**NUM_ERR_SLICES**: Number of error slices supported by this error collator, does not include the GlobalSpace and is derived by ceil (NUM_ERR/32). SW shall first read this register to determine the number of slices and read the required number of Error_Status registers .

PADCTL_DEBUG_EC_SWRESET_0

Offset: 0x404
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_DEBUG_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SWRST**: 1'b1 : Issue a SW reset to the Error Collator. This will reset all the registers(Except SCR), counters and logic of the Error Collator. SW can use this bit to flush errors logged into the error collator for ex, after Boot, SC7/8 exit. 1'b0 : Do nothing, reset value. This bit is auto-cleared.

PADCTL_DEBUG_EC_MISSIONERR_TYPE_0

Offset: 0x408
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_DEBUG_SCR_EC_SCR_0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0101)

Bit	Reset	Description
5:0	0x5	**CODE**: This register indicates the fault code of the error line based on the value of MISSIONERR_INDEX Register. This can be used by a fault handling agent to triage an error without requiring device-specific code.The possible values of this field are: 6'd0 : None 6'd1 : Parity Error on internal data path 6'd2 : ECC SEC Error on internal data path 6'd3 : ECC DED Error on internal data path 6'd4 : Comparator Error 6'd5 : Register Parity Error 6'd6 : Parity Error from on-chip SRAM/Fifo 6'd7 : ECC SEC Error from on-chip SRAM/Fifo 6'd8 : ECC DED Error from on-chip SRAM/Fifo 6'd9 : Clock Monitor Error 6'd10 : Voltage Error 6'd11 : Temperature Error 6'd12 : CRC Error on internal data path 6'd16 : SW Corrected Error 6'd17 : SW Uncorrected Error 6'd18 : SW Generic Error 6'd20 : ECC SED Error on internal data path 6'd21 : ECC SED Error from on-chip SRAM/Fifo 6'd32 : Other HW Corrected Error 6'd33 : Other HW Uncorrected Error All other values : Reserved for future use.

PADCTL_DEBUG_EC_CURRENT_COUNTER_VALUE_0

Offset: 0x40c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_DEBUG_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
8:0	0x0	**VALUE**: Provides the current value of the counter corresponding to the error in MissionErr_Index Register. Default provides the value of error 0 counter. Bit[8] is the ovrerflow bit post which the counter saturates and does not counter further.

PADCTL_DEBUG_EC_MISSIONERR_INDEX_0

Offset: 0x414
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_DEBUG_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**IDX**: BINARY Encoded. For error number 32, register should be programmed with value 0x20. Write to this register with Error number will update: - MISSIONERR_TYPE Register with the Error-Code for the Error. - CURRENT_COUNTER_VALUE Register with the error's SEC/DED Counter. - MISSIONERR_USERVALUE with value of the first error_<i>_user signal. SW can use this to trage the error. number shall update the MISSIONERR_TYPE register with the error code and the Current_Counter_Value register with the value of the errors SEC/DED counter. SW can use this register to triage the error.

PADCTL_DEBUG_EC_CORRECTABLE_THRESHOLD_0

Offset: 0x418
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_DEBUG_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**COUNT**: Threshold value for all SEC Fault Reporting Units connected to this error collator. SEC Errors are logged once the threshold is reached and the overflow bit is set. 7'b0 : Log SEC error after receiving 1 Error. 7'b1 : Log SEC error after receiving 2 Errors. ... 7'bFF : Log SEC error after receiving 256 Errors.

PADCTL_DEBUG_EC_MISSIONERR_INJECT_UNLOCK_0

Offset: 0x41c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_DEBUG_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Writes to ERRSLICE_XXX_MISSIONERR_INJECT registers are disabled until this register is written with a value of 0xE1. This is to prevent an inadvertent safety error injection in the safety plugin due to: 1. A fault on ERRSLICE_XXX_MISSIONERR_INJECT register itself. 2. Erroneous SW. The register shall be written with a value of 0x0 to reestablish the lock after user has completed the error injection testing. 0xE1 : Unlock the MISSIONERR_INJECT Register 0x0 : Lock the MISSIONERR_INJECT Register **0** = LOCK **225** = UNLOCK

PADCTL_DEBUG_EC_ERRSLICE0_MISSIONERR_ENABLE_0

```
==================================
 ERROR SLICE - 0
==================================
```

Offset: 0x430
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_DEBUG_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.DEBUG_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.DEBUG_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.DEBUG_PADCTLREG_reg 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.DEBUG_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL_DEBUG_err_collator 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL_DEBUG_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_DEBUG_EC_ERRSLICE0_MISSIONERR_FORCE_0

Offset: 0x434
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_DEBUG_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.DEBUG_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.DEBUG_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL_DEBUG_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_DEBUG_EC_ERRSLICE0_MISSIONERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of MissionError_Enable register, to avoid silent dropping of errors.
Offset: 0x438
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_DEBUG_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.DEBUG_PADCTLREG_reg_scr_wrap was equal to 2'b10. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.DEBUG_PADCTLREG_reg_scr_wrap was equal to 2'b01.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.DEBUG_PADCTLREG_reg was equal to 2'b10. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.DEBUG_PADCTLREG_reg was equal to 2'b01.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_DEBUG_err_collator was equal to 2'b10. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_DEBUG_err_collator was equal to 2'b01.

PADCTL_DEBUG_EC_ERRSLICE0_MISSIONERR_INJECT_0

Offset: 0x43c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_DEBUG_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Assert the inject_error_2 output for Register Parity Error to PADCTL.DEBUG_PADCTLREG_reg_scr_wrap to allow for error injection. 1'b0 -> De-Asseert inject_error_2 output. **0** = DISABLE **1** = ENABLE
1	0x0	**ERR1**: 1'b1 -> Assert the inject_error_1 output for Register Parity Error to PADCTL.DEBUG_PADCTLREG_reg to allow for error injection. 1'b0 -> De-Asseert inject_error_1 output. **0** = DISABLE **1** = ENABLE
0	0x0	**ERR0**: 1'b1 -> Assert the inject_error_0 output for Register Parity Error to PADCTL_DEBUG_err_collator to allow for error injection. 1'b0 -> De-Asseert inject_error_0 output. **0** = DISABLE **1** = ENABLE

PADCTL_DEBUG_EC_ERRSLICE0_LATENTERR_ENABLE_0

Offset: 0x440
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_DEBUG_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.DEBUG_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.DEBUG_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.DEBUG_PADCTLREG_reg 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.DEBUG_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL_DEBUG_err_collator 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL_DEBUG_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_DEBUG_EC_ERRSLICE0_LATENTERR_FORCE_0

Offset: 0x444
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_DEBUG_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.DEBUG_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.DEBUG_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL_DEBUG_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_DEBUG_EC_ERRSLICE0_LATENTERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of LatentError_Enable register, to avoid silent dropping of errors.
Offset: 0x448
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_DEBUG_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.DEBUG_PADCTLREG_reg_scr_wrap was equal to 2'b00 or 2'b11. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.DEBUG_PADCTLREG_reg_scr_wrap was equal to 2'b01 or 2'b10, but no latent error.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.DEBUG_PADCTLREG_reg was equal to 2'b00 or 2'b11. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.DEBUG_PADCTLREG_reg was equal to 2'b01 or 2'b10, but no latent error.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_DEBUG_err_collator was equal to 2'b00 or 2'b11. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_DEBUG_err_collator was equal to 2'b01 or 2'b10, but no latent error.

PADCTL_DEBUG_EC_ERRSLICE0_COUNTER_RELOAD_0

Offset: 0x450
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_DEBUG_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.DEBUG_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
1	0x0	**ERR1**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.DEBUG_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
0	0x0	**ERR0**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL_DEBUG_err_collator 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

8.4.8 EMMC PAD Control Registers

NOTE:

The EMMC PAD Control Registers are collectively called PADCTL_A6 in the System Address Map.

The binary “x” is turned to “0” for the Reset and PROD hex value. Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as “x” or “Undefined” actually means that the bit could be either “0” or “1” after Reset, whereas a bit PROD value as “x” or “Don’t-care” means that either “0” or “1” can be written to the bit by software Initialization.

PADCTL_EMMC_EMMC_PAD_0
Offset: 0x0
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: PADCTL_EMMC_SCR_SCR_EMMC_PAD_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,xxxx,xxxx,xxxx)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE

PADCTL_EMMC_EMMC_PAD_CLK_0

Offset: 0x8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EMMC_SCR_SCR_EMMC_PAD_CLK_0
Reset: 0x00002024 (0bxxxx,xxxx,xxxx,xxxx,x01x,xxxx,x01x,01xx)

Bit	Reset	Description
14:13	COMP_DRIVE_2X	**DRV_TYPE**: Unused DRV_TYPE registers Not used for EMMC pads DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_DEEP_LPBK**: **0** = DISABLE **1** = ENABLE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

PADCTL_EMMC_EMMC_PAD_CMD_0

Offset: 0x10
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EMMC_SCR_SCR_EMMC_PAD_CMD_0
Reset: 0x00002048 (0bxxxx,xxxx,xxxx,xxxx,x01x,xxxx,x10x,10xx)

Bit	Reset	Description
14:13	COMP_DRIVE_2X	**DRV_TYPE**: Unused DRV_TYPE registers Not used for EMMC pads DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_DEEP_LPBK**: **0** = DISABLE **1** = ENABLE
3:2	PULL_UP	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

PADCTL_EMMC_EMMC_PAD_DQS_0

Offset: 0x18
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EMMC_SCR_SCR_EMMC_PAD_DQS_0
Reset: 0x00000044 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x1xx,01xx)

Bit	Reset	Description
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

PADCTL_EMMC_EMMC_PAD_DAT7_0

Offset: 0x20
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_EMMC_SCR_SCR_EMMC_PAD_DAT7_0
Reset: 0x00002048 (0bxxxx,xxxx,xxxx,xxxx,x01x,xxxx,x10x,10xx)

Bit	Reset	Description
14:13	COMP_DRIVE_2X	**DRV_TYPE**: Unused DRV_TYPE registers Not used for EMMC pads DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_DEEP_LPBK**: **0** = DISABLE **1** = ENABLE
3:2	PULL_UP	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

PADCTL_EMMC_EMMC_PAD_DAT6_0

Offset: 0x28
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EMMC_SCR_SCR_EMMC_PAD_DAT6_0
Reset: 0x00002048 (0bxxxx,xxxx,xxxx,xxxx,x01x,xxxx,x10x,10xx)

Bit	Reset	Description
14:13	COMP_DRIVE_2X	**DRV_TYPE**: Unused DRV_TYPE registers Not used for EMMC pads DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X

Bit	Reset	Description
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_DEEP_LPBK**: **0** = DISABLE **1** = ENABLE
3:2	PULL_UP	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

PADCTL_EMMC_EMMC_PAD_DAT5_0

Offset: 0x30
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EMMC_SCR_SCR_EMMC_PAD_DAT5_0
Reset: 0x00002048 (0bxxxx,xxxx,xxxx,xxxx,x01x,xxxx,x10x,10xx)

Bit	Reset	Description
14:13	COMP_DRIVE_2X	**DRV_TYPE**: Unused DRV_TYPE registers Not used for EMMC pads DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_DEEP_LPBK**: **0** = DISABLE **1** = ENABLE
3:2	PULL_UP	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

PADCTL_EMMC_EMMC_PAD_DAT4_0

Offset: 0x38
Read/Write: R/W

Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EMMC_SCR_SCR_EMMC_PAD_DAT4_0
Reset: 0x00002048 (0bxxxx,xxxx,xxxx,xxxx,x01x,xxxx,x10x,10xx)

Bit	Reset	Description
14:13	COMP_DRIVE_2X	**DRV_TYPE**: Unused DRV_TYPE registers Not used for EMMC pads DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_DEEP_LPBK**: **0** = DISABLE **1** = ENABLE
3:2	PULL_UP	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

PADCTL_EMMC_EMMC_PAD_DAT3_0

Offset: 0x40
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EMMC_SCR_SCR_EMMC_PAD_DAT3_0
Reset: 0x00002048 (0bxxxx,xxxx,0000,xxxx,x01x,xxxx,x10x,10xx)

Bit	Reset	Description
23:20	0x0	**RFU_IN**

Bit	Reset	Description
14:13	COMP_DRIVE_2X	**DRV_TYPE**: Unused DRV_TYPE registers Not used for EMMC pads DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_DEEP_LPBK**: **0** = DISABLE **1** = ENABLE
3:2	PULL_UP	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

PADCTL_EMMC_EMMC_PAD_DAT2_0

Offset: 0x48
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EMMC_SCR_SCR_EMMC_PAD_DAT2_0
Reset: 0x00002048 (0bxxxx,xxxx,0000,xxxx,x01x,xxxx,x10x,10xx)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: Unused DRV_TYPE registers Not used for EMMC pads DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
5	DISABLE	**E_DEEP_LPBK**: **0** = DISABLE **1** = ENABLE
3:2	PULL_UP	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

PADCTL_EMMC_EMMC_PAD_DAT1_0

Offset: 0x50
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EMMC_SCR_SCR_EMMC_PAD_DAT1_0
Reset: 0x00002048 (0bxxxx,xxxx,0000,xxxx,x01x,xxxx,x10x,10xx)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: Unused DRV_TYPE registers Not used for EMMC pads DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_DEEP_LPBK**: **0** = DISABLE **1** = ENABLE
3:2	PULL_UP	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

PADCTL_EMMC_EMMC_PAD_DAT0_0

Offset: 0x58
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: PADCTL_EMMC_SCR_SCR_EMMC_PAD_DAT0_0
Reset: 0x00002048 (0bxxxx,xxxx,0000,xxxx,x01x,xxxx,x10x,10xx)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: Unused DRV_TYPE registers Not used for EMMC pads DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_DEEP_LPBK**: **0** = DISABLE **1** = ENABLE
3:2	PULL_UP	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

PADCTL_EMMC_EC_FEATURE_0

```
================================
 GLOBAL SLICE
================================
```

Offset: 0x400
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_EMMC_SCR_EC_SCR_0
Reset: 0x00020001 (0b0000,0000,0000,0010,xxxx,xxxx,xx00,0001)

Bit	Reset	Description
31:16	0x2	**NUM_ERR**: Number of errors connected to this collator. This is passed as a build time option to the plugin

Bit	Reset	Description
5:0	0x1	**NUM_ERR_SLICES**: Number of error slices supported by this error collator, does not include the GlobalSpace and is derived by ceil (NUM_ERR/32). SW shall first read this register to determine the number of slices and read the required number of Error_Status registers .

PADCTL_EMMC_EC_SWRESET_0

Offset: 0x404
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_EMMC_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SWRST**: 1'b1 : Issue a SW reset to the Error Collator. This will reset all the registers(Except SCR), counters and logic of the Error Collator. SW can use this bit to flush errors logged into the error collator for ex, after Boot, SC7/8 exit. 1'b0 : Do nothing, reset value. This bit is auto-cleared.

PADCTL_EMMC_EC_MISSIONERR_TYPE_0

Offset: 0x408
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_EMMC_SCR_EC_SCR_0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0101)

Bit	Reset	Description
5:0	0x5	**CODE**: This register indicates the fault code of the error line based on the value of MISSIONERR_INDEX Register. This can be used by a fault handling agent to triage an error without requiring device-specific code.The possible values of this field are: 6'd0 : None 6'd1 : Parity Error on internal data path 6'd2 : ECC SEC Error on internal data path 6'd3 : ECC DED Error on internal data path 6'd4 : Comparator Error 6'd5 : Register Parity Error 6'd6 : Parity Error from on-chip SRAM/Fifo 6'd7 : ECC SEC Error from on-chip SRAM/Fifo 6'd8 : ECC DED Error from on-chip SRAM/Fifo 6'd9 : Clock Monitor Error 6'd10 : Voltage Error 6'd11 : Temperature Error 6'd12 : CRC Error on internal data path 6'd16 : SW Corrected Error 6'd17 : SW Uncorrected Error 6'd18 : SW Generic Error 6'd20 : ECC SED Error on internal data path 6'd21 : ECC SED Error from on-chip SRAM/Fifo 6'd32 : Other HW Corrected Error 6'd33 : Other HW Uncorrected Error All other values : Reserved for future use.

PADCTL_EMMC_EC_CURRENT_COUNTER_VALUE_0

Offset: 0x40c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_EMMC_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
8:0	0x0	**VALUE**: Provides the current value of the counter corresponding to the error in MissionErr_Index Register. Default provides the value of error 0 counter. Bit[8] is the ovrerflow bit post which the counter saturates and does not counter further.

PADCTL_EMMC_EC_MISSIONERR_INDEX_0

Offset: 0x414
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_EMMC_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**IDX**: BINARY Encoded. For error number 32, register should be programmed with value 0x20. Write to this register with Error number will update: - MISSIONERR_TYPE Register with the Error-Code for the Error. - CURRENT_COUNTER_VALUE Register with the error's SEC/DED Counter. - MISSIONERR_USERVALUE with value of the first error_<i>_user signal. SW can use this to trage the error. number shall update the MISSIONERR_TYPE register with the error code and the Current_Counter_Value register with the value of the errors SEC/DED counter. SW can use this register to triage the error.

PADCTL_EMMC_EC_CORRECTABLE_THRESHOLD_0

Offset: 0x418
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EMMC_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**COUNT**: Threshold value for all SEC Fault Reporting Units connected to this error collator. SEC Errors are logged once the threshold is reached and the overflow bit is set. 7'b0 : Log SEC error after receiving 1 Error. 7'b1 : Log SEC error after receiving 2 Errors. ... 7'bFF : Log SEC error after receiving 256 Errors.

PADCTL_EMMC_EC_MISSIONERR_INJECT_UNLOCK_0

Offset: 0x41c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_EMMC_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Writes to ERRSLICE_XXX_MISSIONERR_INJECT registers are disabled until this register is written with a value of 0xE1. This is to prevent an inadvertent safety error injection in the safety plugin due to: 1. A fault on ERRSLICE_XXX_MISSIONERR_INJECT register itself. 2. Erroneous SW. The register shall be written with a value of 0x0 to reestablish the lock after user has completed the error injection testing. 0xE1 : Unlock the MISSIONERR_INJECT Register 0x0 : Lock the MISSIONERR_INJECT Register **0** = LOCK **225** = UNLOCK

PADCTL_EMMC_EC_ERRSLICE0_MISSIONERR_ENABLE_0

```
==================================
 ERROR SLICE - 0
==================================
```

Offset: 0x430
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EMMC_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.EMMC_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.EMMC_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.EMMC_PADCTLREG_reg 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.EMMC_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL_EMMC_err_collator 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL_EMMC_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_EMMC_EC_ERRSLICE0_MISSIONERR_FORCE_0

Offset: 0x434
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_EMMC_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.EMMC_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.EMMC_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL_EMMC_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_EMMC_EC_ERRSLICE0_MISSIONERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of MissionError_Enable register, to avoid silent dropping of errors.
Offset: 0x438
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EMMC_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.EMMC_PADCTLREG_reg_scr_wrap was equal to 2'b10. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.EMMC_PADCTLREG_reg_scr_wrap was equal to 2'b01.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.EMMC_PADCTLREG_reg was equal to 2'b10. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.EMMC_PADCTLREG_reg was equal to 2'b01.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_EMMC_err_collator was equal to 2'b10. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_EMMC_err_collator was equal to 2'b01.

PADCTL_EMMC_EC_ERRSLICE0_MISSIONERR_INJECT_0

Offset: 0x43c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_EMMC_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Assert the inject_error_2 output for Register Parity Error to PADCTL.EMMC_PADCTLREG_reg_scr_wrap to allow for error injection. 1'b0 -> De-Asseert inject_error_2 output. **0** = DISABLE **1** = ENABLE
1	0x0	**ERR1**: 1'b1 -> Assert the inject_error_1 output for Register Parity Error to PADCTL.EMMC_PADCTLREG_reg to allow for error injection. 1'b0 -> De-Asseert inject_error_1 output. **0** = DISABLE **1** = ENABLE
0	0x0	**ERR0**: 1'b1 -> Assert the inject_error_0 output for Register Parity Error to PADCTL_EMMC_err_collator to allow for error injection. 1'b0 -> De-Asseert inject_error_0 output. **0** = DISABLE **1** = ENABLE

PADCTL_EMMC_EC_ERRSLICE0_LATENTERR_ENABLE_0

Offset: 0x440
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EMMC_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.EMMC_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.EMMC_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.EMMC_PADCTLREG_reg 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.EMMC_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL_EMMC_err_collator 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL_EMMC_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_EMMC_EC_ERRSLICE0_LATENTERR_FORCE_0

Offset: 0x444
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_EMMC_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.EMMC_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.EMMC_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL_EMMC_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_EMMC_EC_ERRSLICE0_LATENTERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of LatentError_Enable register, to avoid silent dropping of errors.
Offset: 0x448
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EMMC_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.EMMC_PADCTLREG_reg_scr_wrap was equal to 2'b00 or 2'b11. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.EMMC_PADCTLREG_reg_scr_wrap was equal to 2'b01 or 2'b10, but no latent error.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.EMMC_PADCTLREG_reg was equal to 2'b00 or 2'b11. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.EMMC_PADCTLREG_reg was equal to 2'b01 or 2'b10, but no latent error.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_EMMC_err_collator was equal to 2'b00 or 2'b11. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_EMMC_err_collator was equal to 2'b01 or 2'b10, but no latent error.

PADCTL_EMMC_EC_ERRSLICE0_COUNTER_RELOAD_0

Offset: 0x450

Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_EMMC_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.EMMC_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
1	0x0	**ERR1**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.EMMC_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
0	0x0	**ERR0**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL_EMMC_err_collator 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

8.4.9 PEX_CTL PAD Control Registers

NOTE:

The PEX_CTL PAD Control Registers are collectively called PADCTL_A7 in the System Address Map.

The binary “x” is turned to “0” for the Reset and PROD hex value. Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as “x” or “Undefined” actually means that the bit could be either “0” or “1” after Reset, whereas a bit PROD value as “x” or “Don’t-care” means that either “0” or “1” can be written to the bit by software Initialization.

PADCTL_PEX_CTL_PEX_L2_CLKREQ_N_0
Offset: 0x0
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_PEX_CTL_SCR_SCR_PEX_L2_CLKREQ_N_0
Reset: 0x00000560 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0110,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	PE2	**PM**: **0** = PE2 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_PEX_CTL_CFG2TMC_PEX_L2_CLKREQ_N_0

Offset: 0x4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_SCR_PEX_L2_CLKREQ_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**

Bit	Reset	Description
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_PEX_CTL_PEX_WAKE_N_0

Offset: 0x8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_SCR_PEX_WAKE_N_0
Reset: 0x00000474 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0111,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_PEX_CTL_CFG2TMC_PEX_WAKE_N_0

Offset: 0xc
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_SCR_PEX_WAKE_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_PEX_CTL_PEX_L1_CLKREQ_N_0

Offset: 0x10
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_SCR_PEX_L1_CLKREQ_N_0
Reset: 0x00000560 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0110,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE

Bit	Reset	Description
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	PE1	**PM**: **0** = PE1 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_PEX_CTL_CFG2TMC_PEX_L1_CLKREQ_N_0

Offset: 0x14
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_SCR_PEX_L1_CLKREQ_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_PEX_CTL_PEX_L1_RST_N_0

Offset: 0x18
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_SCR_PEX_L1_RST_N_0
Reset: 0x00000520 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0010,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	PE1	**PM**: **0** = PE1 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_PEX_CTL_CFG2TMC_PEX_L1_RST_N_0

Offset: 0x1c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_SCR_PEX_L1_RST_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_PEX_CTL_PEX_L0_CLKREQ_N_0

Offset: 0x20
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_SCR_PEX_L0_CLKREQ_N_0
Reset: 0x00000560 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0110,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	PE0	**PM**: **0** = PE0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_PEX_CTL_CFG2TMC_PEX_L0_CLKREQ_N_0

Offset: 0x24
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_SCR_PEX_L0_CLKREQ_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_PEX_CTL_PEX_L0_RST_N_0

Offset: 0x28
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_SCR_PEX_L0_RST_N_0
Reset: 0x00000520 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0010,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	PE0	**PM**: **0** = PE0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_PEX_CTL_CFG2TMC_PEX_L0_RST_N_0

Offset: 0x2c
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_SCR_PEX_L0_RST_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_PEX_CTL_PEX_L2_RST_N_0
Offset: 0x30
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_SCR_PEX_L2_RST_N_0
Reset: 0x00000520 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0010,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

Bit	Reset	Description
1:0	PE2	**PM**: **0** = PE2 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_PEX_CTL_CFG2TMC_PEX_L2_RST_N_0

Offset: 0x34
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_SCR_PEX_L2_RST_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_PEX_CTL_PEX_L3_CLKREQ_N_0

Offset: 0x38
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_SCR_PEX_L3_CLKREQ_N_0
Reset: 0x00000560 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0110,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	PE3	**PM**: **0** = PE3 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_PEX_CTL_CFG2TMC_PEX_L3_CLKREQ_N_0

Offset: 0x3c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_SCR_PEX_L3_CLKREQ_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_PEX_CTL_PEX_L3_RST_N_0

Offset: 0x40
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_SCR_PEX_L3_RST_N_0
Reset: 0x00000520 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0010,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	PE3	**PM**: **0** = PE3 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_PEX_CTL_CFG2TMC_PEX_L3_RST_N_0

Offset: 0x44
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_SCR_PEX_L3_RST_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_PEX_CTL_PEX_L4_CLKREQ_N_0

Offset: 0x48
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_SCR_PEX_L4_CLKREQ_N_0
Reset: 0x00000560 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0110,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	PE4	**PM**: **0** = PE4 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_PEX_CTL_CFG2TMC_PEX_L4_CLKREQ_N_0

Offset: 0x4c
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_SCR_PEX_L4_CLKREQ_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_PEX_CTL_PEX_L4_RST_N_0
Offset: 0x50
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_SCR_PEX_L4_RST_N_0
Reset: 0x00000520 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0010,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

Bit	Reset	Description
1:0	PE4	**PM**: **0** = PE4 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_PEX_CTL_CFG2TMC_PEX_L4_RST_N_0

Offset: 0x54
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_SCR_PEX_L4_RST_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_PEX_CTL_SOC_GPIO34_0

Offset: 0x58
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_SCR_SOC_GPIO34_0
Reset: 0x00000474 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0111,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_PEX_CTL_CFG2TMC_SOC_GPIO34_0

Offset: 0x5c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_SCR_SOC_GPIO34_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_PEX_CTL_EC_FEATURE_0

```
==================================
 GLOBAL SLICE
==================================
```

Offset: 0x400
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: PADCTL_PEX_CTL_SCR_EC_SCR_0
Reset: 0x00020001 (0b0000,0000,0000,0010,xxxx,xxxx,xx00,0001)

Bit	Reset	Description
31:16	0x2	**NUM_ERR**: Number of errors connected to this collator. This is passed as a build time option to the plugin
5:0	0x1	**NUM_ERR_SLICES**: Number of error slices supported by this error collator, does not include the GlobalSpace and is derived by ceil (NUM_ERR/32). SW shall first read this register to determine the number of slices and read the required number of Error_Status registers .

PADCTL_PEX_CTL_EC_SWRESET_0

Offset: 0x404
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SWRST**: 1'b1 : Issue a SW reset to the Error Collator. This will reset all the registers(Except SCR), counters and logic of the Error Collator. SW can use this bit to flush errors logged into the error collator for ex, after Boot, SC7/8 exit. 1'b0 : Do nothing, reset value. This bit is auto-cleared.

PADCTL_PEX_CTL_EC_MISSIONERR_TYPE_0

Offset: 0x408
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_EC_SCR_0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0101)

Bit	Reset	Description
5:0	0x5	**CODE**: This register indicates the fault code of the error line based on the value of MISSIONERR_INDEX Register. This can be used by a fault handling agent to triage an error without requiring device-specific code.The possible values of this field are: 6'd0 : None 6'd1 : Parity Error on internal data path 6'd2 : ECC SEC Error on internal data path 6'd3 : ECC DED Error on internal data path 6'd4 : Comparator Error 6'd5 : Register Parity Error 6'd6 : Parity Error from on-chip SRAM/Fifo 6'd7 : ECC SEC Error from on-chip SRAM/Fifo 6'd8 : ECC DED Error from on-chip SRAM/Fifo 6'd9 : Clock Monitor Error 6'd10 : Voltage Error 6'd11 : Temperature Error 6'd12 : CRC Error on internal data path 6'd16 : SW Corrected Error 6'd17 : SW Uncorrected Error 6'd18 : SW Generic Error 6'd20 : ECC SED Error on internal data path 6'd21 : ECC SED Error from on-chip SRAM/Fifo 6'd32 : Other HW Corrected Error 6'd33 : Other HW Uncorrected Error All other values : Reserved for future use.

PADCTL_PEX_CTL_EC_CURRENT_COUNTER_VALUE_0

Offset: 0x40c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
8:0	0x0	**VALUE**: Provides the current value of the counter corresponding to the error in MissionErr_Index Register. Default provides the value of error 0 counter. Bit[8] is the ovrerflow bit post which the counter saturates and does not counter further.

PADCTL_PEX_CTL_EC_MISSIONERR_INDEX_0

Offset: 0x414
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_PEX_CTL_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**IDX**: BINARY Encoded. For error number 32, register should be programmed with value 0x20. Write to this register with Error number will update: - MISSIONERR_TYPE Register with the Error-Code for the Error. - CURRENT_COUNTER_VALUE Register with the error's SEC/DED Counter. - MISSIONERR_USERVALUE with value of the first error_<i>_user signal. SW can use this to trage the error. number shall update the MISSIONERR_TYPE register with the error code and the Current_Counter_Value register with the value of the errors SEC/DED counter. SW can use this register to triage the error.

PADCTL_PEX_CTL_EC_CORRECTABLE_THRESHOLD_0

Offset: 0x418
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**COUNT**: Threshold value for all SEC Fault Reporting Units connected to this error collator. SEC Errors are logged once the threshold is reached and the overflow bit is set. 7'b0 : Log SEC error after receiving 1 Error. 7'b1 : Log SEC error after receiving 2 Errors. ... 7'bFF : Log SEC error after receiving 256 Errors.

PADCTL_PEX_CTL_EC_MISSIONERR_INJECT_UNLOCK_0

Offset: 0x41c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Writes to ERRSLICE_XXX_MISSIONERR_INJECT registers are disabled until this register is written with a value of 0xE1. This is to prevent an inadvertent safety error injection in the safety plugin due to: 1. A fault on ERRSLICE_XXX_MISSIONERR_INJECT register itself. 2. Erroneous SW. The register shall be written with a value of 0x0 to reestablish the lock after user has completed the error injection testing. 0xE1 : Unlock the MISSIONERR_INJECT Register 0x0 : Lock the MISSIONERR_INJECT Register **0** = LOCK **225** = UNLOCK

PADCTL_PEX_CTL_EC_ERRSLICE0_MISSIONERR_ENABLE_0

```
==================================
 ERROR SLICE - 0
==================================
```

Offset: 0x430
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.PEX_CTL_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.PEX_CTL_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.PEX_CTL_PADCTLREG_reg 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.PEX_CTL_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL_PEX_CTL_err_collator 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL_PEX_CTL_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_PEX_CTL_EC_ERRSLICE0_MISSIONERR_FORCE_0

Offset: 0x434
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.PEX_CTL_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.PEX_CTL_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL_PEX_CTL_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_PEX_CTL_EC_ERRSLICE0_MISSIONERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of MissionError_Enable register, to avoid silent dropping of errors.
Offset: 0x438
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.PEX_CTL_PADCTLREG_reg_scr_wrap was equal to 2'b10. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.PEX_CTL_PADCTLREG_reg_scr_wrap was equal to 2'b01.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.PEX_CTL_PADCTLREG_reg was equal to 2'b10. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.PEX_CTL_PADCTLREG_reg was equal to 2'b01.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_PEX_CTL_err_collator was equal to 2'b10. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_PEX_CTL_err_collator was equal to 2'b01.

PADCTL_PEX_CTL_EC_ERRSLICE0_MISSIONERR_INJECT_0

Offset: 0x43c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Assert the inject_error_2 output for Register Parity Error to PADCTL.PEX_CTL_PADCTLREG_reg_scr_wrap to allow for error injection. 1'b0 -> De-Asseert inject_error_2 output. **0** = DISABLE **1** = ENABLE
1	0x0	**ERR1**: 1'b1 -> Assert the inject_error_1 output for Register Parity Error to PADCTL.PEX_CTL_PADCTLREG_reg to allow for error injection. 1'b0 -> De-Asseert inject_error_1 output. **0** = DISABLE **1** = ENABLE
0	0x0	**ERR0**: 1'b1 -> Assert the inject_error_0 output for Register Parity Error to PADCTL_PEX_CTL_err_collator to allow for error injection. 1'b0 -> De-Asseert inject_error_0 output. **0** = DISABLE **1** = ENABLE

PADCTL_PEX_CTL_EC_ERRSLICE0_LATENTERR_ENABLE_0

Offset: 0x440
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.PEX_CTL_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.PEX_CTL_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.PEX_CTL_PADCTLREG_reg 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.PEX_CTL_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL_PEX_CTL_err_collator 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL_PEX_CTL_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_PEX_CTL_EC_ERRSLICE0_LATENTERR_FORCE_0

Offset: 0x444
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.PEX_CTL_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.PEX_CTL_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL_PEX_CTL_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_PEX_CTL_EC_ERRSLICE0_LATENTERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of LatentError_Enable register, to avoid silent dropping of errors.
Offset: 0x448
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.PEX_CTL_PADCTLREG_reg_scr_wrap was equal to 2'b00 or 2'b11. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.PEX_CTL_PADCTLREG_reg_scr_wrap was equal to 2'b01 or 2'b10, but no latent error.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.PEX_CTL_PADCTLREG_reg was equal to 2'b00 or 2'b11. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.PEX_CTL_PADCTLREG_reg was equal to 2'b01 or 2'b10, but no latent error.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_PEX_CTL_err_collator was equal to 2'b00 or 2'b11. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_PEX_CTL_err_collator was equal to 2'b01 or 2'b10, but no latent error.

PADCTL_PEX_CTL_EC_ERRSLICE0_COUNTER_RELOAD_0

Offset: 0x450
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_PEX_CTL_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.PEX_CTL_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
1	0x0	**ERR1**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.PEX_CTL_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
0	0x0	**ERR0**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL_PEX_CTL_err_collator 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

8.4.10 SDMMC1 High Voltage PAD Control Registers

NOTE:

The SDMMC1 High Voltage PAD Control Registers are collectively called PADCTL_A8 in the System Address Map.

The binary “x” is turned to “0” for the Reset and PROD hex value. Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as “x” or “Undefined” actually means that the bit could be either “0” or “1” after Reset, whereas a bit PROD value as “x” or “Don’t-care” means that either “0” or “1” can be written to the bit by software Initialization.

PADCTL_SDMMC1_HV_SDMMC1_CLK_0
Offset: 0x0
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_SDMMC1_HV_SCR_SCR_SDMMC1_CLK_0
Reset: 0x0000a474 (0b0000,0000,0000,xxx0,1010,x10x,x111,0100)

Bit	Reset	Description
31:28	0x0	**SPARE_VDD**
27:24	0x0	**SPARE_VAUXC**
23:20	0x0	**RFU_IN**
16:15	0x1	**SCHMITT**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
9	DISABLE	**E_PBIAS_BUF**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SDMMC1	**PM**: **0** = SDMMC1 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_SDMMC1_HV_CFG2TMC_SDMMC1_CLK_0

Offset: 0x4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SDMMC1_HV_SCR_SCR_SDMMC1_CLK_0
Reset: 0x00000000 (0b0000,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:30	0x0	**CFG_CAL_DRVUP_DEF**
29:28	0x0	**CFG_CAL_DRVDN_DEF**

PADCTL_SDMMC1_HV_SDMMC1_CMD_0

Offset: 0x8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SDMMC1_HV_SCR_SCR_SDMMC1_CMD_0
Reset: 0x0000a454 (0b0000,0000,0000,xxx0,1010,x10x,x101,0100)

Bit	Reset	Description
31:28	0x0	**SPARE_VDD**
27:24	0x0	**SPARE_VAUXC**
23:20	0x0	**RFU_IN**
16:15	0x1	**SCHMITT**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO

Bit	Reset	Description
9	DISABLE	**E_PBIAS_BUF**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SDMMC1	**PM**: **0** = SDMMC1 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_SDMMC1_HV_CFG2TMC_SDMMC1_CMD_0

Offset: 0xc
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SDMMC1_HV_SCR_SCR_SDMMC1_CMD_0
Reset: 0x00000000 (0b0000,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:30	0x0	**CFG_CAL_DRVUP_DEF**
29:28	0x0	**CFG_CAL_DRVDN_DEF**

PADCTL_SDMMC1_HV_SDMMC1_COMP_0

Offset: 0x10
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_SDMMC1_HV_SCR_SCR_SDMMC1_COMP_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,xx00)

Bit	Reset	Description
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
1:0	SDMMC1	**PM**: **0** = SDMMC1 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_SDMMC1_HV_SDMMC1_DAT3_0

Offset: 0x18
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SDMMC1_HV_SCR_SCR_SDMMC1_DAT3_0
Reset: 0x0000a454 (0b0000,0000,0000,xxx0,1010,x10x,x101,0100)

Bit	Reset	Description
31:28	0x0	**SPARE_VDD**
27:24	0x0	**SPARE_VAUXC**
23:20	0x0	**RFU_IN**
16:15	0x1	**SCHMITT**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
9	DISABLE	**E_PBIAS_BUF**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SDMMC1	**PM**: **0** = SDMMC1 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_SDMMC1_HV_CFG2TMC_SDMMC1_DAT3_0

Offset: 0x1c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SDMMC1_HV_SCR_SCR_SDMMC1_DAT3_0
Reset: 0x00000000 (0b0000,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:30	0x0	**CFG_CAL_DRVUP_DEF**
29:28	0x0	**CFG_CAL_DRVDN_DEF**

PADCTL_SDMMC1_HV_SDMMC1_DAT2_0

Offset: 0x20
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SDMMC1_HV_SCR_SCR_SDMMC1_DAT2_0
Reset: 0x0000a454 (0b0000,0000,0000,xxx0,1010,x10x,x101,0100)

Bit	Reset	Description
31:28	0x0	**SPARE_VDD**
27:24	0x0	**SPARE_VAUXC**

Bit	Reset	Description
23:20	0x0	**RFU_IN**
16:15	0x1	**SCHMITT**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
9	DISABLE	**E_PBIAS_BUF**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SDMMC1	**PM**: **0** = SDMMC1 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_SDMMC1_HV_CFG2TMC_SDMMC1_DAT2_0

Offset: 0x24
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: PADCTL_SDMMC1_HV_SCR_SCR_SDMMC1_DAT2_0
Reset: 0x00000000 (0b0000,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:30	0x0	**CFG_CAL_DRVUP_DEF**
29:28	0x0	**CFG_CAL_DRVDN_DEF**

PADCTL_SDMMC1_HV_SDMMC1_DAT1_0

Offset: 0x28
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SDMMC1_HV_SCR_SCR_SDMMC1_DAT1_0
Reset: 0x0000a454 (0b0000,0000,0000,xxx0,1010,x10x,x101,0100)

Bit	Reset	Description
31:28	0x0	**SPARE_VDD**
27:24	0x0	**SPARE_VAUXC**
23:20	0x0	**RFU_IN**
16:15	0x1	**SCHMITT**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
9	DISABLE	**E_PBIAS_BUF**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SDMMC1	**PM**: **0** = SDMMC1 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_SDMMC1_HV_CFG2TMC_SDMMC1_DAT1_0

Offset: 0x2c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SDMMC1_HV_SCR_SCR_SDMMC1_DAT1_0
Reset: 0x00000000 (0b0000,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:30	0x0	**CFG_CAL_DRVUP_DEF**
29:28	0x0	**CFG_CAL_DRVDN_DEF**

PADCTL_SDMMC1_HV_SDMMC1_DAT0_0

Offset: 0x30
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SDMMC1_HV_SCR_SCR_SDMMC1_DAT0_0
Reset: 0x0000a454 (0b0000,0000,0000,xxx0,1010,x10x,x101,0100)

Bit	Reset	Description
31:28	0x0	**SPARE_VDD**
27:24	0x0	**SPARE_VAUXC**

Bit	Reset	Description
23:20	0x0	**RFU_IN**
16:15	0x1	**SCHMITT**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
9	DISABLE	**E_PBIAS_BUF**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SDMMC1	**PM**: **0** = SDMMC1 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_SDMMC1_HV_CFG2TMC_SDMMC1_DAT0_0

Offset: 0x34
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: PADCTL_SDMMC1_HV_SCR_SCR_SDMMC1_DAT0_0
Reset: 0x00000000 (0b0000,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:30	0x0	**CFG_CAL_DRVUP_DEF**
29:28	0x0	**CFG_CAL_DRVDN_DEF**

PADCTL_SDMMC1_HV_EC_FEATURE_0

```
==================================
 GLOBAL SLICE
==================================
```

Offset: 0x400
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_SDMMC1_HV_SCR_EC_SCR_0
Reset: 0x00020001 (0b0000,0000,0000,0010,xxxx,xxxx,xx00,0001)

Bit	Reset	Description
31:16	0x2	**NUM_ERR**: Number of errors connected to this collator. This is passed as a build time option to the plugin
5:0	0x1	**NUM_ERR_SLICES**: Number of error slices supported by this error collator, does not include the GlobalSpace and is derived by ceil (NUM_ERR/32). SW shall first read this register to determine the number of slices and read the required number of Error_Status registers .

PADCTL_SDMMC1_HV_EC_SWRESET_0

Offset: 0x404
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_SDMMC1_HV_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SWRST**: 1'b1 : Issue a SW reset to the Error Collator. This will reset all the registers(Except SCR), counters and logic of the Error Collator. SW can use this bit to flush errors logged into the error collator for ex, after Boot, SC7/8 exit. 1'b0 : Do nothing, reset value. This bit is auto-cleared.

PADCTL_SDMMC1_HV_EC_MISSIONERR_TYPE_0

Offset: 0x408
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_SDMMC1_HV_SCR_EC_SCR_0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0101)

Bit	Reset	Description
5:0	0x5	**CODE**: This register indicates the fault code of the error line based on the value of MISSIONERR_INDEX Register. This can be used by a fault handling agent to triage an error without requiring device-specific code.The possible values of this field are: 6'd0 : None 6'd1 : Parity Error on internal data path 6'd2 : ECC SEC Error on internal data path 6'd3 : ECC DED Error on internal data path 6'd4 : Comparator Error 6'd5 : Register Parity Error 6'd6 : Parity Error from on-chip SRAM/Fifo 6'd7 : ECC SEC Error from on-chip SRAM/Fifo 6'd8 : ECC DED Error from on-chip SRAM/Fifo 6'd9 : Clock Monitor Error 6'd10 : Voltage Error 6'd11 : Temperature Error 6'd12 : CRC Error on internal data path 6'd16 : SW Corrected Error 6'd17 : SW Uncorrected Error 6'd18 : SW Generic Error 6'd20 : ECC SED Error on internal data path 6'd21 : ECC SED Error from on-chip SRAM/Fifo 6'd32 : Other HW Corrected Error 6'd33 : Other HW Uncorrected Error All other values : Reserved for future use.

PADCTL_SDMMC1_HV_EC_CURRENT_COUNTER_VALUE_0

Offset: 0x40c
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: PADCTL_SDMMC1_HV_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
8:0	0x0	**VALUE**: Provides the current value of the counter corresponding to the error in MissionErr_Index Register. Default provides the value of error 0 counter. Bit[8] is the ovrerflow bit post which the counter saturates and does not counter further.

PADCTL_SDMMC1_HV_EC_MISSIONERR_INDEX_0

Offset: 0x414
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SDMMC1_HV_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**IDX**: BINARY Encoded. For error number 32, register should be programmed with value 0x20. Write to this register with Error number will update: - MISSIONERR_TYPE Register with the Error-Code for the Error. - CURRENT_COUNTER_VALUE Register with the error's SEC/DED Counter. - MISSIONERR_USERVALUE with value of the first error_<i>_user signal. SW can use this to trage the error. number shall update the MISSIONERR_TYPE register with the error code and the Current_Counter_Value register with the value of the errors SEC/DED counter. SW can use this register to triage the error.

PADCTL_SDMMC1_HV_EC_CORRECTABLE_THRESHOLD_0

Offset: 0x418
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SDMMC1_HV_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**COUNT**: Threshold value for all SEC Fault Reporting Units connected to this error collator. SEC Errors are logged once the threshold is reached and the overflow bit is set. 7'b0 : Log SEC error after receiving 1 Error. 7'b1 : Log SEC error after receiving 2 Errors. ... 7'bFF : Log SEC error after receiving 256 Errors.

PADCTL_SDMMC1_HV_EC_MISSIONERR_INJECT_UNLOCK_0

Offset: 0x41c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_SDMMC1_HV_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Writes to ERRSLICE_XXX_MISSIONERR_INJECT registers are disabled until this register is written with a value of 0xE1. This is to prevent an inadvertent safety error injection in the safety plugin due to: 1. A fault on ERRSLICE_XXX_MISSIONERR_INJECT register itself. 2. Erroneous SW. The register shall be written with a value of 0x0 to reestablish the lock after user has completed the error injection testing. 0xE1 : Unlock the MISSIONERR_INJECT Register 0x0 : Lock the MISSIONERR_INJECT Register **0** = LOCK **225** = UNLOCK

PADCTL_SDMMC1_HV_EC_ERRSLICE0_MISSIONERR_ENABLE_0

```
================================
 ERROR SLICE - 0
================================
```

Offset: 0x430
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SDMMC1_HV_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.SDMMC1_HV_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.SDMMC1_HV_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.SDMMC1_HV_PADCTLREG_reg 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.SDMMC1_HV_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL_SDMMC1_HV_err_collator 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL_SDMMC1_HV_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_SDMMC1_HV_EC_ERRSLICE0_MISSIONERR_FORCE_0

Offset: 0x434
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_SDMMC1_HV_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.SDMMC1_HV_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.SDMMC1_HV_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL_SDMMC1_HV_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_SDMMC1_HV_EC_ERRSLICE0_MISSIONERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of MissionError_Enable register, to avoid silent dropping of errors.
Offset: 0x438
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SDMMC1_HV_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.SDMMC1_HV_PADCTLREG_reg_scr_wrap was equal to 2'b10. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.SDMMC1_HV_PADCTLREG_reg_scr_wrap was equal to 2'b01.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.SDMMC1_HV_PADCTLREG_reg was equal to 2'b10. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.SDMMC1_HV_PADCTLREG_reg was equal to 2'b01.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_SDMMC1_HV_err_collator was equal to 2'b10. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_SDMMC1_HV_err_collator was equal to 2'b01.

PADCTL_SDMMC1_HV_EC_ERRSLICE0_MISSIONERR_INJECT_0

Offset: 0x43c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_SDMMC1_HV_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Assert the inject_error_2 output for Register Parity Error to PADCTL.SDMMC1_HV_PADCTLREG_reg_scr_wrap to allow for error injection. 1'b0 -> De-Asseert inject_error_2 output. **0** = DISABLE **1** = ENABLE
1	0x0	**ERR1**: 1'b1 -> Assert the inject_error_1 output for Register Parity Error to PADCTL.SDMMC1_HV_PADCTLREG_reg to allow for error injection. 1'b0 -> De-Asseert inject_error_1 output. **0** = DISABLE **1** = ENABLE
0	0x0	**ERR0**: 1'b1 -> Assert the inject_error_0 output for Register Parity Error to PADCTL_SDMMC1_HV_err_collator to allow for error injection. 1'b0 -> De-Asseert inject_error_0 output. **0** = DISABLE **1** = ENABLE

PADCTL_SDMMC1_HV_EC_ERRSLICE0_LATENTERR_ENABLE_0

Offset: 0x440
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SDMMC1_HV_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.SDMMC1_HV_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.SDMMC1_HV_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.SDMMC1_HV_PADCTLREG_reg 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.SDMMC1_HV_PADCTLREG_reg **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
0	0x1	**ERR0**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL_SDMMC1_HV_err_collator 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL_SDMMC1_HV_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_SDMMC1_HV_EC_ERRSLICE0_LATENTERR_FORCE_0

Offset: 0x444
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_SDMMC1_HV_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.SDMMC1_HV_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.SDMMC1_HV_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL_SDMMC1_HV_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_SDMMC1_HV_EC_ERRSLICE0_LATENTERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of LatentError_Enable register, to avoid silent dropping of errors.
Offset: 0x448
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: PADCTL_SDMMC1_HV_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.SDMMC1_HV_PADCTLREG_reg_scr_wrap was equal to 2'b00 or 2'b11. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.SDMMC1_HV_PADCTLREG_reg_scr_wrap was equal to 2'b01 or 2'b10, but no latent error.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.SDMMC1_HV_PADCTLREG_reg was equal to 2'b00 or 2'b11. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.SDMMC1_HV_PADCTLREG_reg was equal to 2'b01 or 2'b10, but no latent error.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_SDMMC1_HV_err_collator was equal to 2'b00 or 2'b11. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_SDMMC1_HV_err_collator was equal to 2'b01 or 2'b10, but no latent error.

PADCTL_SDMMC1_HV_EC_ERRSLICE0_COUNTER_RELOAD_0

Offset: 0x450
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_SDMMC1_HV_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.SDMMC1_HV_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
1	0x0	**ERR1**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.SDMMC1_HV_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

Bit	Reset	Description
0	0x0	**ERR0**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL_SDMMC1_HV_err_collator 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

8.4.11 QSPI PAD Control Registers

NOTE:

The QSPI PAD Control Registers are collectively called PADCTL_A11 in the System Address Map.

The binary “x” is turned to “0” for the Reset and PROD hex value. Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as “x” or “Undefined” actually means that the bit could be either “0” or “1” after Reset, whereas a bit PROD value as “x” or “Don’t-care” means that either “0” or “1” can be written to the bit by software Initialization.

PADCTL_QSPI_QSPI0_IO3_0
Offset: 0x0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_QSPI_SCR_SCR_QSPI0_IO3_0
Reset: 0x00002458 (0bxxxx,xxxx,0000,xxxx,x010,x1xx,x101,1000)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_UP	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	QSPI0	**PM**: **0** = QSPI0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_QSPI_QSPI0_IO2_0

Offset: 0x8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_QSPI_SCR_SCR_QSPI0_IO2_0
Reset: 0x00002458 (0bxxxx,xxxx,0000,xxxx,x010,x1xx,x101,1000)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_UP	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	QSPI0	**PM**: **0** = QSPI0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_QSPI_QSPI0_IO1_0

Offset: 0x10
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_QSPI_SCR_SCR_QSPI0_IO1_0
Reset: 0x00002458 (0bxxxx,xxxx,0000,xxxx,x010,x1xx,x101,1000)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_UP	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	QSPI0	**PM**: **0** = QSPI0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_QSPI_QSPI0_IO0_0

Offset: 0x18
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_QSPI_SCR_SCR_QSPI0_IO0_0
Reset: 0x00002458 (0bxxxx,xxxx,0000,xxxx,x010,x1xx,x101,1000)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_UP	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	QSPI0	**PM**: **0** = QSPI0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_QSPI_QSPI0_SCK_0

Offset: 0x20
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_QSPI_SCR_SCR_QSPI0_SCK_0
Reset: 0x00002474 (0bxxxx,xxxx,0000,xxxx,x010,x1xx,x111,0100)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	QSPI0	**PM**: **0** = QSPI0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_QSPI_QSPI0_CS_N_0

Offset: 0x28
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_QSPI_SCR_SCR_QSPI0_CS_N_0
Reset: 0x00002418 (0bxxxx,xxxx,0000,xxxx,x010,x1xx,x001,1000)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_UP	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	QSPI0	**PM**: **0** = QSPI0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_QSPI_QSPI1_IO3_0

Offset: 0x30
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_QSPI_SCR_SCR_QSPI1_IO3_0
Reset: 0x00002458 (0bxxxx,xxxx,0000,xxxx,x010,x1xx,x101,1000)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_UP	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	QSPI1	**PM**: **0** = QSPI1 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_QSPI_QSPI1_IO2_0

Offset: 0x38
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_QSPI_SCR_SCR_QSPI1_IO2_0
Reset: 0x00002458 (0bxxxx,xxxx,0000,xxxx,x010,x1xx,x101,1000)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_UP	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	QSPI1	**PM**: **0** = QSPI1 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_QSPI_QSPI1_IO1_0

Offset: 0x40
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_QSPI_SCR_SCR_QSPI1_IO1_0
Reset: 0x00002458 (0bxxxx,xxxx,0000,xxxx,x010,x1xx,x101,1000)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_UP	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	QSPI1	**PM**: **0** = QSPI1 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_QSPI_QSPI1_IO0_0

Offset: 0x48
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_QSPI_SCR_SCR_QSPI1_IO0_0
Reset: 0x00002458 (0bxxxx,xxxx,0000,xxxx,x010,x1xx,x101,1000)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_UP	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	QSPI1	**PM**: **0** = QSPI1 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_QSPI_QSPI1_SCK_0

Offset: 0x50
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_QSPI_SCR_SCR_QSPI1_SCK_0
Reset: 0x00002474 (0bxxxx,xxxx,0000,xxxx,x010,x1xx,x111,0100)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	QSPI1	**PM**: **0** = QSPI1 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_QSPI_QSPI1_CS_N_0

Offset: 0x58
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_QSPI_SCR_SCR_QSPI1_CS_N_0
Reset: 0x00002418 (0bxxxx,xxxx,0000,xxxx,x010,x1xx,x001,1000)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_UP	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	QSPI1	**PM**: **0** = QSPI1 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_QSPI_QSPI_COMP_0

Offset: 0x60
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_QSPI_SCR_SCR_QSPI_COMP_0
Reset: 0x00002000 (0bxxxx,xxxx,0000,xxxx,x01x,xxxx,xxx0,xx00)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE

Bit	Reset	Description
1:0	QSPI	**PM**: **0** = QSPI **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_QSPI_EC_FEATURE_0

```
=================================
 GLOBAL SLICE
=================================
```

Offset: 0x400
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_QSPI_SCR_EC_SCR_0
Reset: 0x00020001 (0b0000,0000,0000,0010,xxxx,xxxx,xx00,0001)

Bit	Reset	Description
31:16	0x2	**NUM_ERR**: Number of errors connected to this collator. This is passed as a build time option to the plugin
5:0	0x1	**NUM_ERR_SLICES**: Number of error slices supported by this error collator, does not include the GlobalSpace and is derived by ceil (NUM_ERR/32). SW shall first read this register to determine the number of slices and read the required number of Error_Status registers .

PADCTL_QSPI_EC_SWRESET_0

Offset: 0x404
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_QSPI_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SWRST**: 1'b1 : Issue a SW reset to the Error Collator. This will reset all the registers(Except SCR), counters and logic of the Error Collator. SW can use this bit to flush errors logged into the error collator for ex, after Boot, SC7/8 exit. 1'b0 : Do nothing, reset value. This bit is auto-cleared.

PADCTL_QSPI_EC_MISSIONERR_TYPE_0

Offset: 0x408
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_QSPI_SCR_EC_SCR_0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0101)

Bit	Reset	Description
5:0	0x5	**CODE**: This register indicates the fault code of the error line based on the value of MISSIONERR_INDEX Register. This can be used by a fault handling agent to triage an error without requiring device-specific code.The possible values of this field are: 6'd0 : None 6'd1 : Parity Error on internal data path 6'd2 : ECC SEC Error on internal data path 6'd3 : ECC DED Error on internal data path 6'd4 : Comparator Error 6'd5 : Register Parity Error 6'd6 : Parity Error from on-chip SRAM/Fifo 6'd7 : ECC SEC Error from on-chip SRAM/Fifo 6'd8 : ECC DED Error from on-chip SRAM/Fifo 6'd9 : Clock Monitor Error 6'd10 : Voltage Error 6'd11 : Temperature Error 6'd12 : CRC Error on internal data path 6'd16 : SW Corrected Error 6'd17 : SW Uncorrected Error 6'd18 : SW Generic Error 6'd20 : ECC SED Error on internal data path 6'd21 : ECC SED Error from on-chip SRAM/Fifo 6'd32 : Other HW Corrected Error 6'd33 : Other HW Uncorrected Error All other values : Reserved for future use.

PADCTL_QSPI_EC_CURRENT_COUNTER_VALUE_0

Offset: 0x40c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_QSPI_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
8:0	0x0	**VALUE**: Provides the current value of the counter corresponding to the error in MissionErr_Index Register. Default provides the value of error 0 counter. Bit[8] is the ovrerflow bit post which the counter saturates and does not counter further.

PADCTL_QSPI_EC_MISSIONERR_INDEX_0

Offset: 0x414
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_QSPI_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**IDX**: BINARY Encoded. For error number 32, register should be programmed with value 0x20. Write to this register with Error number will update: - MISSIONERR_TYPE Register with the Error-Code for the Error. - CURRENT_COUNTER_VALUE Register with the error's SEC/DED Counter. - MISSIONERR_USERVALUE with value of the first error_<i>_user signal. SW can use this to trage the error. number shall update the MISSIONERR_TYPE register with the error code and the Current_Counter_Value register with the value of the errors SEC/DED counter. SW can use this register to triage the error.

PADCTL_QSPI_EC_CORRECTABLE_THRESHOLD_0

Offset: 0x418
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_QSPI_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**COUNT**: Threshold value for all SEC Fault Reporting Units connected to this error collator. SEC Errors are logged once the threshold is reached and the overflow bit is set. 7'b0 : Log SEC error after receiving 1 Error. 7'b1 : Log SEC error after receiving 2 Errors. ... 7'bFF : Log SEC error after receiving 256 Errors.

PADCTL_QSPI_EC_MISSIONERR_INJECT_UNLOCK_0

Offset: 0x41c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_QSPI_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Writes to ERRSLICE_XXX_MISSIONERR_INJECT registers are disabled until this register is written with a value of 0xE1. This is to prevent an inadvertent safety error injection in the safety plugin due to: 1. A fault on ERRSLICE_XXX_MISSIONERR_INJECT register itself. 2. Erroneous SW. The register shall be written with a value of 0x0 to reestablish the lock after user has completed the error injection testing. 0xE1 : Unlock the MISSIONERR_INJECT Register 0x0 : Lock the MISSIONERR_INJECT Register **0** = LOCK **225** = UNLOCK

PADCTL_QSPI_EC_ERRSLICE0_MISSIONERR_ENABLE_0

```
==================================
 ERROR SLICE - 0
==================================
```

Offset: 0x430
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_QSPI_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.QSPI_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.QSPI_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
1	0x1	**ERR1**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.QSPI_PADCTLREG_reg 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.QSPI_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL_QSPI_err_collator 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL_QSPI_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_QSPI_EC_ERRSLICE0_MISSIONERR_FORCE_0

Offset: 0x434
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_QSPI_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.QSPI_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.QSPI_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL_QSPI_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_QSPI_EC_ERRSLICE0_MISSIONERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of MissionError_Enable register, to avoid silent dropping of errors.
Offset: 0x438
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_QSPI_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.QSPI_PADCTLREG_reg_scr_wrap was equal to 2'b10. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.QSPI_PADCTLREG_reg_scr_wrap was equal to 2'b01.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.QSPI_PADCTLREG_reg was equal to 2'b10. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.QSPI_PADCTLREG_reg was equal to 2'b01.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_QSPI_err_collator was equal to 2'b10. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_QSPI_err_collator was equal to 2'b01.

PADCTL_QSPI_EC_ERRSLICE0_MISSIONERR_INJECT_0

Offset: 0x43c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_QSPI_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Assert the inject_error_2 output for Register Parity Error to PADCTL.QSPI_PADCTLREG_reg_scr_wrap to allow for error injection. 1'b0 -> De-Asseert inject_error_2 output. **0** = DISABLE **1** = ENABLE
1	0x0	**ERR1**: 1'b1 -> Assert the inject_error_1 output for Register Parity Error to PADCTL.QSPI_PADCTLREG_reg to allow for error injection. 1'b0 -> De-Asseert inject_error_1 output. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
0	0x0	**ERR0**: 1'b1 -> Assert the inject_error_0 output for Register Parity Error to PADCTL_QSPI_err_collator to allow for error injection. 1'b0 -> De-Asseert inject_error_0 output. **0** = DISABLE **1** = ENABLE

PADCTL_QSPI_EC_ERRSLICE0_LATENTERR_ENABLE_0

Offset: 0x440
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_QSPI_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.QSPI_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.QSPI_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.QSPI_PADCTLREG_reg 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.QSPI_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL_QSPI_err_collator 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL_QSPI_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_QSPI_EC_ERRSLICE0_LATENTERR_FORCE_0

Offset: 0x444
Read/Write: WO
Parity Protection: N
Shadow: N

SCR Protection: PADCTL_QSPI_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.QSPI_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.QSPI_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL_QSPI_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_QSPI_EC_ERRSLICE0_LATENTERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of LatentError_Enable register, to avoid silent dropping of errors.
Offset: 0x448
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_QSPI_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.QSPI_PADCTLREG_reg_scr_wrap was equal to 2'b00 or 2'b11. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.QSPI_PADCTLREG_reg_scr_wrap was equal to 2'b01 or 2'b10, but no latent error.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.QSPI_PADCTLREG_reg was equal to 2'b00 or 2'b11. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.QSPI_PADCTLREG_reg was equal to 2'b01 or 2'b10, but no latent error.

Bit	Reset	Description
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_QSPI_err_collator was equal to 2'b00 or 2'b11. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_QSPI_err_collator was equal to 2'b01 or 2'b10, but no latent error.

PADCTL_QSPI_EC_ERRSLICE0_COUNTER_RELOAD_0
Offset: 0x450
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_QSPI_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.QSPI_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
1	0x0	**ERR1**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.QSPI_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
0	0x0	**ERR0**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL_QSPI_err_collator 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

8.4.12 SYS PAD Control Registers

NOTE:

The SYS PAD Control Registers are collectively called PADCTL_A12 in the System Address Map.

The binary “x” is turned to “0” for the Reset and PROD hex value. Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as “x” or “Undefined” actually means that the bit could be either “0” or “1” after Reset, whereas a bit

PROD value as “x” or “Don’t-care” means that either “0” or “1” can be written to the bit by software Initialization.

PADCTL_SYS_SHUTDOWN_N_0
Offset: 0x0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SYS_SCR_SCR_SHUTDOWN_N_0
Reset: 0x00000020 (0bxxxx,xxxx,xxxx,xxxx,xxx0,xxx0,001x,00xx)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

PADCTL_SYS_CFG2TMC_SHUTDOWN_N_0
Offset: 0x4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SYS_SCR_SCR_SHUTDOWN_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_SYS_PMU_INT_N_0
Offset: 0x8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SYS_SCR_SCR_PMU_INT_N_0
Reset: 0x00001040 (0bxxxx,xxxx,xxxx,xxxx,xxx1,xxx0,01xx,00xx)

Bit	Reset	Description
12	ENABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

PADCTL_SYS_CFG2TMC_PMU_INT_N_0
Offset: 0xc
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SYS_SCR_SCR_PMU_INT_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_SYS_SCE_ERROR_0
Offset: 0x10

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SYS_SCR_SCR_SCE_ERROR_0
Reset: 0x00000470 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0111,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SCE	**PM**: **0** = SCE **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_SYS_CFG2TMC_SCE_ERROR_0

Offset: 0x14
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SYS_SCR_SCR_SCE_ERROR_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_SYS_SOC_PWR_REQ_0

Offset: 0x18
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SYS_SCR_SCR_SOC_PWR_REQ_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,xxx0,00xx,00xx)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

PADCTL_SYS_CFG2TMC_SOC_PWR_REQ_0

Offset: 0x1c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SYS_SCR_SCR_SOC_PWR_REQ_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**

Bit	Reset	Description
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_SYS_BATT_OC_0
Offset: 0x20
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SYS_SCR_SCR_BATT_OC_0
Reset: 0x00001474 (0bxxxx,xxxx,xxxx,xxxx,xxx1,x1x0,0111,0100)

Bit	Reset	Description
12	ENABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SOC	**PM**: **0** = SOC **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_SYS_CFG2TMC_BATT_OC_0

Offset: 0x24
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SYS_SCR_SCR_BATT_OC_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_SYS_BOOTV_CTL_N_0

Offset: 0x28
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SYS_SCR_SCR_BOOTV_CTL_N_0
Reset: 0x00000410 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE

Bit	Reset	Description
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_SYS_CFG2TMC_BOOTV_CTL_N_0

Offset: 0x2c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SYS_SCR_SCR_BOOTV_CTL_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_SYS_CLK_32K_IN_0

Offset: 0x30
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SYS_SCR_SCR_CLK_32K_IN_0
Reset: 0x00001020 (0bxxxx,xxxx,xxxx,xxxx,xxx1,xxx0,xx1x,00xx)

Bit	Reset	Description
12	ENABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

PADCTL_SYS_CFG2TMC_CLK_32K_IN_0

Offset: 0x34
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SYS_SCR_SCR_CLK_32K_IN_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_SYS_POWER_ON_0

Offset: 0x38
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SYS_SCR_SCR_POWER_ON_0
Reset: 0x00000474 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0111,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_SYS_CFG2TMC_POWER_ON_0

Offset: 0x3c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SYS_SCR_SCR_POWER_ON_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_SYS_SOC_GPIO26_0

Offset: 0x40
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SYS_SCR_SCR_SOC_GPIO26_0
Reset: 0x00001470 (0bxxxx,xxxx,xxxx,xxxx,xxx1,x1x0,0111,0000)

Bit	Reset	Description
12	ENABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_SYS_CFG2TMC_SOC_GPIO26_0

Offset: 0x44
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SYS_SCR_SCR_SOC_GPIO26_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_SYS_SOC_GPIO27_0
Offset: 0x48
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SYS_SCR_SCR_SOC_GPIO27_0
Reset: 0x00000470 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0111,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_SYS_CFG2TMC_SOC_GPIO27_0
Offset: 0x4c
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: PADCTL_SYS_SCR_SCR_SOC_GPIO27_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_SYS_AO_RETENTION_N_0
Offset: 0x50
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SYS_SCR_SCR_AO_RETENTION_N_0
Reset: 0x00000470 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0111,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

Bit	Reset	Description
1:0	GPIO	**PM**: **0** = GPIO **1** = LED **2** = RSVD2 **3** = ISTCTRL

PADCTL_SYS_CFG2TMC_AO_RETENTION_N_0

Offset: 0x54
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SYS_SCR_SCR_AO_RETENTION_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_SYS_VCOMP_ALERT_0

Offset: 0x58
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SYS_SCR_SCR_VCOMP_ALERT_0
Reset: 0x00000474 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0111,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SOC	**PM**: **0** = SOC **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_SYS_CFG2TMC_VCOMP_ALERT_0

Offset: 0x5c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SYS_SCR_SCR_VCOMP_ALERT_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_SYS_HDMI_CEC_0

Offset: 0x60
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SYS_SCR_SCR_HDMI_CEC_0
Reset: 0x00000470 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0111,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	HDMI	**PM**: **0** = HDMI **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_SYS_CFG2TMC_HDMI_CEC_0

Offset: 0x64
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_SYS_SCR_SCR_HDMI_CEC_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

8.4.13 G2 PAD Control Registers

NOTE:

The G2 PAD Control Registers are collectively called PADCTL_A13 in the System Address Map.

The binary “x” is turned to “0” for the Reset and PROD hex value. Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as “x” or “Undefined” actually means that the bit could be either “0” or “1” after Reset, whereas a bit PROD value as “x” or “Don’t-care” means that either “0” or “1” can be written to the bit by software Initialization.

PADCTL_G2_SPI3_MISO_0
Offset: 0x0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_SPI3_MISO_0
Reset: 0x00000474 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0111,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE

Bit	Reset	Description
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SPI3	**PM**: **0** = SPI3 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G2_CFG2TMC_SPI3_MISO_0

Offset: 0x4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_SPI3_MISO_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G2_SPI1_CS0_0

Offset: 0x8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_SPI1_CS0_0
Reset: 0x00000430 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0011,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SPI1	**PM**: **0** = SPI1 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G2_CFG2TMC_SPI1_CS0_0

Offset: 0xc
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_SPI1_CS0_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G2_SPI3_CS0_0

Offset: 0x10
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_SPI3_CS0_0
Reset: 0x00000430 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0011,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SPI3	**PM**: **0** = SPI3 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G2_CFG2TMC_SPI3_CS0_0

Offset: 0x14
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_SPI3_CS0_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G2_SPI1_MISO_0

Offset: 0x18
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_SPI1_MISO_0
Reset: 0x00000434 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0011,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SPI1	**PM**: **0** = SPI1 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G2_CFG2TMC_SPI1_MISO_0

Offset: 0x1c
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_SPI1_MISO_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G2_SPI3_CS1_0
Offset: 0x20
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_SPI3_CS1_0
Reset: 0x00000430 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0011,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

Bit	Reset	Description
1:0	SPI3	**PM**: **0** = SPI3 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G2_CFG2TMC_SPI3_CS1_0

Offset: 0x24
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_SPI3_CS1_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G2_SPI1_SCK_0

Offset: 0x28
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_SPI1_SCK_0
Reset: 0x00001434 (0bxxxx,xxxx,xxxx,xxxx,xxx1,x1x0,0011,0100)

Bit	Reset	Description
12	ENABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SPI1	**PM**: **0** = SPI1 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G2_CFG2TMC_SPI1_SCK_0

Offset: 0x2c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_SPI1_SCK_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G2_SPI3_SCK_0

Offset: 0x30
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_SPI3_SCK_0
Reset: 0x00001434 (0bxxxx,xxxx,xxxx,xxxx,xxx1,x1x0,0011,0100)

Bit	Reset	Description
12	ENABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SPI3	**PM**: **0** = SPI3 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G2_CFG2TMC_SPI3_SCK_0

Offset: 0x34
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_SPI3_SCK_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G2_SPI1_CS1_0

Offset: 0x38
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_SPI1_CS1_0
Reset: 0x00000430 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0011,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SPI1	**PM**: **0** = SPI1 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G2_CFG2TMC_SPI1_CS1_0

Offset: 0x3c
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_SPI1_CS1_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G2_SPI1_MOSI_0
Offset: 0x40
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_SPI1_MOSI_0
Reset: 0x00000434 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0011,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

Bit	Reset	Description
1:0	SPI1	**PM**: **0** = SPI1 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G2_CFG2TMC_SPI1_MOSI_0

Offset: 0x44
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_SPI1_MOSI_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G2_SPI3_MOSI_0

Offset: 0x48
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_SPI3_MOSI_0
Reset: 0x00000434 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0011,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SPI3	**PM**: **0** = SPI3 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G2_CFG2TMC_SPI3_MOSI_0

Offset: 0x4c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_SPI3_MOSI_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G2_UART2_TX_0

Offset: 0x50
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_UART2_TX_0
Reset: 0x00000474 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0111,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	UARTB	**PM**: **0** = UARTB **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G2_CFG2TMC_UART2_TX_0

Offset: 0x54
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_UART2_TX_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G2_UART2_RX_0

Offset: 0x58
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_UART2_RX_0
Reset: 0x00000474 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0111,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	UARTB	**PM**: **0** = UARTB **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G2_CFG2TMC_UART2_RX_0

Offset: 0x5c
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_UART2_RX_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G2_UART2_RTS_0
Offset: 0x60
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_UART2_RTS_0
Reset: 0x00000454 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,01x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	UARTB	**PM**: **0** = UARTB **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G2_CFG2TMC_UART2_RTS_0

Offset: 0x64
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_UART2_RTS_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G2_UART2_CTS_0

Offset: 0x68
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_UART2_CTS_0
Reset: 0x00000458 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,01x1,1000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE

Bit	Reset	Description
3:2	PULL_UP	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	UARTB	**PM**: **0** = UARTB **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G2_CFG2TMC_UART2_CTS_0

Offset: 0x6c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_UART2_CTS_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G2_UART5_TX_0

Offset: 0x70
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_UART5_TX_0
Reset: 0x00000454 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,01x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	UARTE	**PM**: **0** = UARTE **1** = RSVD1 **2** = UARTI **3** = RSVD3

PADCTL_G2_CFG2TMC_UART5_TX_0

Offset: 0x74
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_UART5_TX_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G2_UART5_RX_0

Offset: 0x78
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_UART5_RX_0
Reset: 0x00000454 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,01x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	UARTE	**PM**: **0** = UARTE **1** = RSVD1 **2** = UARTI **3** = RSVD3

PADCTL_G2_CFG2TMC_UART5_RX_0

Offset: 0x7c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_UART5_RX_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G2_UART5_RTS_0

Offset: 0x80
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_UART5_RTS_0
Reset: 0x00000454 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,01x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	UARTE	**PM**: **0** = UARTE **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G2_CFG2TMC_UART5_RTS_0

Offset: 0x84
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_UART5_RTS_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G2_UART5_CTS_0

Offset: 0x88
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_UART5_CTS_0
Reset: 0x00000458 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,01x1,1000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_UP	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	UARTE	**PM**: **0** = UARTE **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G2_CFG2TMC_UART5_CTS_0

Offset: 0x8c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_UART5_CTS_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G2_GPU_PWR_REQ_0
Offset: 0x90
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_GPU_PWR_REQ_0
Reset: 0x00000400 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x0,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

Bit	Reset	Description
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G2_CFG2TMC_GPU_PWR_REQ_0

Offset: 0x94
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_GPU_PWR_REQ_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G2_GP_PWM3_0

Offset: 0x98
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_GP_PWM3_0
Reset: 0x00000430 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0011,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	GP	**PM**: **0** = GP **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G2_CFG2TMC_GP_PWM3_0

Offset: 0x9c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_GP_PWM3_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G2_GP_PWM2_0

Offset: 0xa0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_GP_PWM2_0
Reset: 0x00000430 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0011,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	GP	**PM**: **0** = GP **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G2_CFG2TMC_GP_PWM2_0

Offset: 0xa4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_GP_PWM2_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G2_CV_PWR_REQ_0

Offset: 0xa8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_CV_PWR_REQ_0
Reset: 0x00000400 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x0,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G2_CFG2TMC_CV_PWR_REQ_0

Offset: 0xac
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_G2_SCR_SCR_CV_PWR_REQ_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G2_USB_VBUS_EN0_0
Offset: 0xb0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_USB_VBUS_EN0_0
Reset: 0x00000420 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0010,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

Bit	Reset	Description
1:0	USB	**PM**: **0** = USB **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G2_CFG2TMC_USB_VBUS_EN0_0

Offset: 0xb4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_USB_VBUS_EN0_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G2_USB_VBUS_EN1_0

Offset: 0xb8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_USB_VBUS_EN1_0
Reset: 0x00000420 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0010,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	USB	**PM**: **0** = USB **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G2_CFG2TMC_USB_VBUS_EN1_0

Offset: 0xbc
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_SCR_USB_VBUS_EN1_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G2_EC_FEATURE_0

```
==================================
 GLOBAL SLICE
==================================
```

Offset: 0x400
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: PADCTL_G2_SCR_EC_SCR_0
Reset: 0x00020001 (0b0000,0000,0000,0010,xxxx,xxxx,xx00,0001)

Bit	Reset	Description
31:16	0x2	**NUM_ERR**: Number of errors connected to this collator. This is passed as a build time option to the plugin
5:0	0x1	**NUM_ERR_SLICES**: Number of error slices supported by this error collator, does not include the GlobalSpace and is derived by ceil (NUM_ERR/32). SW shall first read this register to determine the number of slices and read the required number of Error_Status registers .

PADCTL_G2_EC_SWRESET_0

Offset: 0x404
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G2_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SWRST**: 1'b1 : Issue a SW reset to the Error Collator. This will reset all the registers(Except SCR), counters and logic of the Error Collator. SW can use this bit to flush errors logged into the error collator for ex, after Boot, SC7/8 exit. 1'b0 : Do nothing, reset value. This bit is auto-cleared.

PADCTL_G2_EC_MISSIONERR_TYPE_0

Offset: 0x408
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G2_SCR_EC_SCR_0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0101)

Bit	Reset	Description
5:0	0x5	**CODE**: This register indicates the fault code of the error line based on the value of MISSIONERR_INDEX Register. This can be used by a fault handling agent to triage an error without requiring device-specific code.The possible values of this field are: 6'd0 : None 6'd1 : Parity Error on internal data path 6'd2 : ECC SEC Error on internal data path 6'd3 : ECC DED Error on internal data path 6'd4 : Comparator Error 6'd5 : Register Parity Error 6'd6 : Parity Error from on-chip SRAM/Fifo 6'd7 : ECC SEC Error from on-chip SRAM/Fifo 6'd8 : ECC DED Error from on-chip SRAM/Fifo 6'd9 : Clock Monitor Error 6'd10 : Voltage Error 6'd11 : Temperature Error 6'd12 : CRC Error on internal data path 6'd16 : SW Corrected Error 6'd17 : SW Uncorrected Error 6'd18 : SW Generic Error 6'd20 : ECC SED Error on internal data path 6'd21 : ECC SED Error from on-chip SRAM/Fifo 6'd32 : Other HW Corrected Error 6'd33 : Other HW Uncorrected Error All other values : Reserved for future use.

PADCTL_G2_EC_CURRENT_COUNTER_VALUE_0

Offset: 0x40c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G2_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
8:0	0x0	**VALUE**: Provides the current value of the counter corresponding to the error in MissionErr_Index Register. Default provides the value of error 0 counter. Bit[8] is the ovrerflow bit post which the counter saturates and does not counter further.

PADCTL_G2_EC_MISSIONERR_INDEX_0

Offset: 0x414
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_G2_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**IDX**: BINARY Encoded. For error number 32, register should be programmed with value 0x20. Write to this register with Error number will update: - MISSIONERR_TYPE Register with the Error-Code for the Error. - CURRENT_COUNTER_VALUE Register with the error's SEC/DED Counter. - MISSIONERR_USERVALUE with value of the first error_<i>_user signal. SW can use this to trage the error. number shall update the MISSIONERR_TYPE register with the error code and the Current_Counter_Value register with the value of the errors SEC/DED counter. SW can use this register to triage the error.

PADCTL_G2_EC_CORRECTABLE_THRESHOLD_0

Offset: 0x418
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**COUNT**: Threshold value for all SEC Fault Reporting Units connected to this error collator. SEC Errors are logged once the threshold is reached and the overflow bit is set. 7'b0 : Log SEC error after receiving 1 Error. 7'b1 : Log SEC error after receiving 2 Errors. ... 7'bFF : Log SEC error after receiving 256 Errors.

PADCTL_G2_EC_MISSIONERR_INJECT_UNLOCK_0

Offset: 0x41c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G2_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Writes to ERRSLICE_XXX_MISSIONERR_INJECT registers are disabled until this register is written with a value of 0xE1. This is to prevent an inadvertent safety error injection in the safety plugin due to: 1. A fault on ERRSLICE_XXX_MISSIONERR_INJECT register itself. 2. Erroneous SW. The register shall be written with a value of 0x0 to reestablish the lock after user has completed the error injection testing. 0xE1 : Unlock the MISSIONERR_INJECT Register 0x0 : Lock the MISSIONERR_INJECT Register **0** = LOCK **225** = UNLOCK

PADCTL_G2_EC_ERRSLICE0_MISSIONERR_ENABLE_0

```
================================
 ERROR SLICE - 0
================================
```

Offset: 0x430
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.G2_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.G2_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.G2_PADCTLREG_reg 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.G2_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL_G2_err_collator 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL_G2_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_G2_EC_ERRSLICE0_MISSIONERR_FORCE_0

Offset: 0x434
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G2_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.G2_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.G2_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL_G2_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_G2_EC_ERRSLICE0_MISSIONERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of MissionError_Enable register, to avoid silent dropping of errors.
Offset: 0x438
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G2_PADCTLREG_reg_scr_wrap was equal to 2'b10. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G2_PADCTLREG_reg_scr_wrap was equal to 2'b01.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G2_PADCTLREG_reg was equal to 2'b10. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G2_PADCTLREG_reg was equal to 2'b01.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G2_err_collator was equal to 2'b10. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G2_err_collator was equal to 2'b01.

PADCTL_G2_EC_ERRSLICE0_MISSIONERR_INJECT_0

Offset: 0x43c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G2_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Assert the inject_error_2 output for Register Parity Error to PADCTL.G2_PADCTLREG_reg_scr_wrap to allow for error injection. 1'b0 -> De-Asseert inject_error_2 output. **0** = DISABLE **1** = ENABLE
1	0x0	**ERR1**: 1'b1 -> Assert the inject_error_1 output for Register Parity Error to PADCTL.G2_PADCTLREG_reg to allow for error injection. 1'b0 -> De-Asseert inject_error_1 output. **0** = DISABLE **1** = ENABLE
0	0x0	**ERR0**: 1'b1 -> Assert the inject_error_0 output for Register Parity Error to PADCTL_G2_err_collator to allow for error injection. 1'b0 -> De-Asseert inject_error_0 output. **0** = DISABLE **1** = ENABLE

PADCTL_G2_EC_ERRSLICE0_LATENTERR_ENABLE_0

Offset: 0x440
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.G2_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.G2_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.G2_PADCTLREG_reg 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.G2_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL_G2_err_collator 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL_G2_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_G2_EC_ERRSLICE0_LATENTERR_FORCE_0

Offset: 0x444
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G2_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.G2_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.G2_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL_G2_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_G2_EC_ERRSLICE0_LATENTERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of LatentError_Enable register, to avoid silent dropping of errors.
Offset: 0x448
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G2_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G2_PADCTLREG_reg_scr_wrap was equal to 2'b00 or 2'b11. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G2_PADCTLREG_reg_scr_wrap was equal to 2'b01 or 2'b10, but no latent error.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G2_PADCTLREG_reg was equal to 2'b00 or 2'b11. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G2_PADCTLREG_reg was equal to 2'b01 or 2'b10, but no latent error.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G2_err_collator was equal to 2'b00 or 2'b11. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G2_err_collator was equal to 2'b01 or 2'b10, but no latent error.

PADCTL_G2_EC_ERRSLICE0_COUNTER_RELOAD_0

Offset: 0x450

Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G2_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.G2_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
1	0x0	**ERR1**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.G2_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
0	0x0	**ERR0**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL_G2_err_collator 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

8.4.14 Always On PAD Control Registers

NOTE:

The AO PAD Control Registers are collectively called PADCTL_A14 in the System Address Map.

The binary “x” is turned to “0” for the Reset and PROD hex value. Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as “x” or “Undefined” actually means that the bit could be either “0” or “1” after Reset, whereas a bit PROD value as “x” or “Don’t-care” means that either “0” or “1” can be written to the bit by software Initialization.

PADCTL_AO_TOUCH_CLK_0
Offset: 0x0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_SCR_SCR_TOUCH_CLK_0
Reset: 0x00000430 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0011,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	GP	**PM**: **0** = GP **1** = TOUCH **2** = RSVD2 **3** = RSVD3

PADCTL_AO_CFG2TMC_TOUCH_CLK_0

Offset: 0x4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_SCR_SCR_TOUCH_CLK_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_AO_UART3_RX_0

Offset: 0x8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_SCR_SCR_UART3_RX_0
Reset: 0x00000474 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0111,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	UARTC	**PM**: **0** = UARTC **1** = UARTJ **2** = RSVD2 **3** = RSVD3

PADCTL_AO_CFG2TMC_UART3_RX_0

Offset: 0xc
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: PADCTL_AO_SCR_SCR_UART3_RX_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_AO_UART3_TX_0
Offset: 0x10
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_SCR_SCR_UART3_TX_0
Reset: 0x00000434 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0011,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

Bit	Reset	Description
1:0	UARTC	**PM**: **0** = UARTC **1** = UARTJ **2** = RSVD2 **3** = RSVD3

PADCTL_AO_CFG2TMC_UART3_TX_0

Offset: 0x14
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_SCR_SCR_UART3_TX_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_AO_GEN8_I2C_SDA_0

Offset: 0x18
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_SCR_SCR_GEN8_I2C_SDA_0
Reset: 0x00000570 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0111,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	I2C8	**PM**: **0** = I2C8 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_AO_CFG2TMC_GEN8_I2C_SDA_0

Offset: 0x1c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_SCR_SCR_GEN8_I2C_SDA_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_AO_GEN8_I2C_SCL_0

Offset: 0x20
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_SCR_SCR_GEN8_I2C_SCL_0
Reset: 0x00001570 (0bxxxx,xxxx,xxxx,xxxx,xxx1,x1x1,0111,0000)

Bit	Reset	Description
12	ENABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	I2C8	**PM**: **0** = I2C8 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_AO_CFG2TMC_GEN8_I2C_SCL_0

Offset: 0x24
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_SCR_SCR_GEN8_I2C_SCL_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_AO_SPI2_MOSI_0

Offset: 0x28
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_SCR_SCR_SPI2_MOSI_0
Reset: 0x00000474 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0111,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SPI2	**PM**: **0** = SPI2 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_AO_CFG2TMC_SPI2_MOSI_0

Offset: 0x2c
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: PADCTL_AO_SCR_SCR_SPI2_MOSI_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_AO_GEN2_I2C_SCL_0
Offset: 0x30
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_SCR_SCR_GEN2_I2C_SCL_0
Reset: 0x00001570 (0bxxxx,xxxx,xxxx,xxxx,xxx1,x1x1,0111,0000)

Bit	Reset	Description
12	ENABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

Bit	Reset	Description
1:0	I2C2	**PM**: **0** = I2C2 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_AO_CFG2TMC_GEN2_I2C_SCL_0

Offset: 0x34
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_SCR_SCR_GEN2_I2C_SCL_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_AO_SPI2_CS0_0

Offset: 0x38
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_SCR_SCR_SPI2_CS0_0
Reset: 0x00000470 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0111,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SPI2	**PM**: **0** = SPI2 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_AO_CFG2TMC_SPI2_CS0_0

Offset: 0x3c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_SCR_SCR_SPI2_CS0_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_AO_GEN2_I2C_SDA_0

Offset: 0x40
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_SCR_SCR_GEN2_I2C_SDA_0
Reset: 0x00000570 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0111,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	I2C2	**PM**: **0** = I2C2 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_AO_CFG2TMC_GEN2_I2C_SDA_0

Offset: 0x44
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_SCR_SCR_GEN2_I2C_SDA_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_AO_SPI2_SCK_0
Offset: 0x48
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_SCR_SCR_SPI2_SCK_0
Reset: 0x00001474 (0bxxxx,xxxx,xxxx,xxxx,xxx1,x1x0,0111,0100)

Bit	Reset	Description
12	ENABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SPI2	**PM**: **0** = SPI2 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_AO_CFG2TMC_SPI2_SCK_0
Offset: 0x4c
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: PADCTL_AO_SCR_SCR_SPI2_SCK_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_AO_SPI2_MISO_0
Offset: 0x50
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_SCR_SCR_SPI2_MISO_0
Reset: 0x00000474 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0111,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

Bit	Reset	Description
1:0	SPI2	**PM**: **0** = SPI2 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_AO_CFG2TMC_SPI2_MISO_0
Offset: 0x54
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_SCR_SCR_SPI2_MISO_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

8.4.15 Always On High Voltage PAD Control Registers

NOTE:

The Always ON High Voltage PAD Control Registers are collectively called PADCTL_A15 in the System Address Map.

The binary “x” is turned to “0” for the Reset and PROD hex value. Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as “x” or “Undefined” actually means that the bit could be either “0” or “1” after Reset, whereas a bit PROD value as “x” or “Don’t-care” means that either “0” or “1” can be written to the bit by software Initialization.

PADCTL_AO_HV_CAN1_DOUT_0
Offset: 0x0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_HV_SCR_SCR_CAN1_DOUT_0
Reset: 0x0000c410 (0b0000,0000,0000,xxx0,1100,x10x,x001,0000)

Bit	Reset	Description
31:28	0x0	**SPARE_VDD**
27:24	0x0	**SPARE_VAUXC**
23:20	0x0	**RFU_IN**
16:15	0x1	**SCHMITT**
14:13	DEFAULT_DRIVE_1X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
9	DISABLE	**E_PBIAS_BUF**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	CAN1	**PM**: **0** = CAN1 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_AO_HV_CFG2TMC_CAN1_DOUT_0

Offset: 0x4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_HV_SCR_SCR_CAN1_DOUT_0
Reset: 0x00000000 (0b0000,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:30	0x0	**CFG_CAL_DRVUP_DEF**
29:28	0x0	**CFG_CAL_DRVDN_DEF**

PADCTL_AO_HV_CAN1_DIN_0
Offset: 0x8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_HV_SCR_SCR_CAN1_DIN_0
Reset: 0x0000c450 (0b0000,0000,0000,xxx0,1100,x10x,x101,0000)

Bit	Reset	Description
31:28	0x0	**SPARE_VDD**
27:24	0x0	**SPARE_VAUXC**
23:20	0x0	**RFU_IN**
16:15	0x1	**SCHMITT**
14:13	DEFAULT_DRIVE_1X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
9	DISABLE	**E_PBIAS_BUF**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	CAN1	**PM**: **0** = CAN1 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_AO_HV_CFG2TMC_CAN1_DIN_0

Offset: 0xc
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_HV_SCR_SCR_CAN1_DIN_0
Reset: 0x00000000 (0b0000,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:30	0x0	**CFG_CAL_DRVUP_DEF**
29:28	0x0	**CFG_CAL_DRVDN_DEF**

PADCTL_AO_HV_CAN0_DOUT_0

Offset: 0x10
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_HV_SCR_SCR_CAN0_DOUT_0
Reset: 0x0000c410 (0b0000,0000,0000,xxx0,1100,x10x,x001,0000)

Bit	Reset	Description
31:28	0x0	**SPARE_VDD**
27:24	0x0	**SPARE_VAUXC**
23:20	0x0	**RFU_IN**
16:15	0x1	**SCHMITT**
14:13	DEFAULT_DRIVE_1X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
9	DISABLE	**E_PBIAS_BUF**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	CAN0	**PM**: **0** = CAN0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_AO_HV_CFG2TMC_CAN0_DOUT_0

Offset: 0x14
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_HV_SCR_SCR_CAN0_DOUT_0
Reset: 0x00000000 (0b0000,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:30	0x0	**CFG_CAL_DRVUP_DEF**
29:28	0x0	**CFG_CAL_DRVDN_DEF**

PADCTL_AO_HV_CAN0_DIN_0
Offset: 0x18
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_HV_SCR_SCR_CAN0_DIN_0
Reset: 0x0000c450 (0b0000,0000,0000,xxx0,1100,x10x,x101,0000)

Bit	Reset	Description
31:28	0x0	**SPARE_VDD**
27:24	0x0	**SPARE_VAUXC**
23:20	0x0	**RFU_IN**
16:15	0x1	**SCHMITT**
14:13	DEFAULT_DRIVE_1X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
9	DISABLE	**E_PBIAS_BUF**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	CAN0	**PM**: **0** = CAN0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_AO_HV_CFG2TMC_CAN0_DIN_0

Offset: 0x1c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_HV_SCR_SCR_CAN0_DIN_0
Reset: 0x00000000 (0b0000,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:30	0x0	**CFG_CAL_DRVUP_DEF**
29:28	0x0	**CFG_CAL_DRVDN_DEF**

PADCTL_AO_HV_CAN0_STB_0

Offset: 0x20
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_HV_SCR_SCR_CAN0_STB_0
Reset: 0x0000c414 (0b0000,0000,0000,xxx0,1100,x10x,x001,0100)

Bit	Reset	Description
31:28	0x0	**SPARE_VDD**
27:24	0x0	**SPARE_VAUXC**
23:20	0x0	**RFU_IN**
16:15	0x1	**SCHMITT**
14:13	DEFAULT_DRIVE_1X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
9	DISABLE	**E_PBIAS_BUF**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = WDT **2** = TSC **3** = TSC_ALT

PADCTL_AO_HV_CFG2TMC_CAN0_STB_0

Offset: 0x24
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_HV_SCR_SCR_CAN0_STB_0
Reset: 0x00000000 (0b0000,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:30	0x0	**CFG_CAL_DRVUP_DEF**
29:28	0x0	**CFG_CAL_DRVDN_DEF**

PADCTL_AO_HV_CAN0_EN_0
Offset: 0x28
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_HV_SCR_SCR_CAN0_EN_0
Reset: 0x0000c414 (0b0000,0000,0000,xxx0,1100,x10x,x001,0100)

Bit	Reset	Description
31:28	0x0	**SPARE_VDD**
27:24	0x0	**SPARE_VAUXC**
23:20	0x0	**RFU_IN**
16:15	0x1	**SCHMITT**
14:13	DEFAULT_DRIVE_1X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
9	DISABLE	**E_PBIAS_BUF**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_AO_HV_CFG2TMC_CAN0_EN_0

Offset: 0x2c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_HV_SCR_SCR_CAN0_EN_0
Reset: 0x00000000 (0b0000,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:30	0x0	**CFG_CAL_DRVUP_DEF**
29:28	0x0	**CFG_CAL_DRVDN_DEF**

PADCTL_AO_HV_SOC_GPIO49_0

Offset: 0x30
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_HV_SCR_SCR_SOC_GPIO49_0
Reset: 0x0000c414 (0b0000,0000,0000,xxx0,1100,x10x,x001,0100)

Bit	Reset	Description
31:28	0x0	**SPARE_VDD**
27:24	0x0	**SPARE_VAUXC**
23:20	0x0	**RFU_IN**
16:15	0x1	**SCHMITT**
14:13	DEFAULT_DRIVE_1X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
9	DISABLE	**E_PBIAS_BUF**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_AO_HV_CFG2TMC_SOC_GPIO49_0

Offset: 0x34
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_HV_SCR_SCR_SOC_GPIO49_0
Reset: 0x00000000 (0b0000,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:30	0x0	**CFG_CAL_DRVUP_DEF**
29:28	0x0	**CFG_CAL_DRVDN_DEF**

PADCTL_AO_HV_CAN0_ERR_0
Offset: 0x38
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_HV_SCR_SCR_CAN0_ERR_0
Reset: 0x0000c414 (0b0000,0000,0000,xxx0,1100,x10x,x001,0100)

Bit	Reset	Description
31:28	0x0	**SPARE_VDD**
27:24	0x0	**SPARE_VAUXC**
23:20	0x0	**RFU_IN**
16:15	0x1	**SCHMITT**
14:13	DEFAULT_DRIVE_1X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
9	DISABLE	**E_PBIAS_BUF**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = TSC **2** = RSVD2 **3** = TSC_ALT

PADCTL_AO_HV_CFG2TMC_CAN0_ERR_0

Offset: 0x3c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_HV_SCR_SCR_CAN0_ERR_0
Reset: 0x00000000 (0b0000,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:30	0x0	**CFG_CAL_DRVUP_DEF**
29:28	0x0	**CFG_CAL_DRVDN_DEF**

PADCTL_AO_HV_CAN1_STB_0

Offset: 0x40
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_HV_SCR_SCR_CAN1_STB_0
Reset: 0x0000c414 (0b0000,0000,0000,xxx0,1100,x10x,x001,0100)

Bit	Reset	Description
31:28	0x0	**SPARE_VDD**
27:24	0x0	**SPARE_VAUXC**
23:20	0x0	**RFU_IN**
16:15	0x1	**SCHMITT**
14:13	DEFAULT_DRIVE_1X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
9	DISABLE	**E_PBIAS_BUF**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = DMIC3 **2** = DMIC5 **3** = RSVD3

PADCTL_AO_HV_CFG2TMC_CAN1_STB_0

Offset: 0x44
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_HV_SCR_SCR_CAN1_STB_0
Reset: 0x00000000 (0b0000,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:30	0x0	**CFG_CAL_DRVUP_DEF**
29:28	0x0	**CFG_CAL_DRVDN_DEF**

PADCTL_AO_HV_CAN1_EN_0
Offset: 0x48
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_HV_SCR_SCR_CAN1_EN_0
Reset: 0x0000c414 (0b0000,0000,0000,xxx0,1100,x10x,x001,0100)

Bit	Reset	Description
31:28	0x0	**SPARE_VDD**
27:24	0x0	**SPARE_VAUXC**
23:20	0x0	**RFU_IN**
16:15	0x1	**SCHMITT**
14:13	DEFAULT_DRIVE_1X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
9	DISABLE	**E_PBIAS_BUF**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = DMIC3 **2** = DMIC5 **3** = RSVD3

PADCTL_AO_HV_CFG2TMC_CAN1_EN_0

Offset: 0x4c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_HV_SCR_SCR_CAN1_EN_0
Reset: 0x00000000 (0b0000,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:30	0x0	**CFG_CAL_DRVUP_DEF**
29:28	0x0	**CFG_CAL_DRVDN_DEF**

PADCTL_AO_HV_SOC_GPIO50_0

Offset: 0x50
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_HV_SCR_SCR_SOC_GPIO50_0
Reset: 0x0000c414 (0b0000,0000,0000,xxx0,1100,x10x,x001,0100)

Bit	Reset	Description
31:28	0x0	**SPARE_VDD**
27:24	0x0	**SPARE_VAUXC**
23:20	0x0	**RFU_IN**
16:15	0x1	**SCHMITT**
14:13	DEFAULT_DRIVE_1X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
9	DISABLE	**E_PBIAS_BUF**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = TSC **2** = RSVD2 **3** = TSC_ALT

PADCTL_AO_HV_CFG2TMC_SOC_GPIO50_0

Offset: 0x54
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_HV_SCR_SCR_SOC_GPIO50_0
Reset: 0x00000000 (0b0000,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:30	0x0	**CFG_CAL_DRVUP_DEF**
29:28	0x0	**CFG_CAL_DRVDN_DEF**

PADCTL_AO_HV_CAN1_ERR_0

Offset: 0x58
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_HV_SCR_SCR_CAN1_ERR_0
Reset: 0x0000c414 (0b0000,0000,0000,xxx0,1100,x10x,x001,0100)

Bit	Reset	Description
31:28	0x0	**SPARE_VDD**
27:24	0x0	**SPARE_VAUXC**
23:20	0x0	**RFU_IN**
16:15	0x1	**SCHMITT**
14:13	DEFAULT_DRIVE_1X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
9	DISABLE	**E_PBIAS_BUF**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = TSC **2** = RSVD2 **3** = TSC_ALT

PADCTL_AO_HV_CFG2TMC_CAN1_ERR_0
Offset: 0x5c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_AO_HV_SCR_SCR_CAN1_ERR_0
Reset: 0x00000000 (0b0000,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:30	0x0	**CFG_CAL_DRVUP_DEF**
29:28	0x0	**CFG_CAL_DRVDN_DEF**

8.4.16 EDP PAD Control Registers

NOTE:

The EDP PAD Control Registers are collectively called PADCTL_A16 in the System Address Map.

The binary “x” is turned to “0” for the Reset and PROD hex value. Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as “x” or “Undefined” actually means that the bit could be either “0” or “1” after Reset, whereas a bit

PROD value as “x” or “Don’t-care” means that either “0” or “1” can be written to the bit by software Initialization.

PADCTL_EDP_SOC_GPIO36_0

Offset: 0x0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_SOC_GPIO36_0
Reset: 0x00000410 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	ETH0	**PM**: **0** = ETH0 **1** = RSVD1 **2** = DCA **3** = RSVD3

PADCTL_EDP_CFG2TMC_SOC_GPIO36_0

Offset: 0x4
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_EDP_SCR_SCR_SOC_GPIO36_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_EDP_SOC_GPIO53_0
Offset: 0x8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_SOC_GPIO53_0
Reset: 0x00000414 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	ETH0	**PM**: **0** = ETH0 **1** = RSVD1 **2** = DCA **3** = RSVD3

PADCTL_EDP_CFG2TMC_SOC_GPIO53_0

Offset: 0xc
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_SOC_GPIO53_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_EDP_SOC_GPIO55_0

Offset: 0x10
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_SOC_GPIO55_0
Reset: 0x00000414 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE

Bit	Reset	Description
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	ETH2	**PM**: **0** = ETH2 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_EDP_CFG2TMC_SOC_GPIO55_0

Offset: 0x14
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_SOC_GPIO55_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_EDP_SOC_GPIO38_0

Offset: 0x18
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_SOC_GPIO38_0
Reset: 0x00000410 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	ETH1	**PM**: **0** = ETH1 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_EDP_CFG2TMC_SOC_GPIO38_0

Offset: 0x1c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_SOC_GPIO38_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_EDP_SOC_GPIO39_0

Offset: 0x20
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_SOC_GPIO39_0
Reset: 0x00000410 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	GP	**PM**: **0** = GP **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_EDP_CFG2TMC_SOC_GPIO39_0

Offset: 0x24
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_SOC_GPIO39_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_EDP_SOC_GPIO40_0

Offset: 0x28
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_SOC_GPIO40_0
Reset: 0x00000414 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	ETH1	**PM**: **0** = ETH1 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_EDP_CFG2TMC_SOC_GPIO40_0

Offset: 0x2c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_SOC_GPIO40_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_EDP_DP_AUX_CH0_HPD_0

Offset: 0x30
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_DP_AUX_CH0_HPD_0
Reset: 0x00001470 (0bxxxx,xxxx,xxxx,xxxx,xxx1,x1x0,0111,0000)

Bit	Reset	Description
12	ENABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	DP	**PM**: **0** = DP **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_EDP_CFG2TMC_DP_AUX_CH0_HPD_0

Offset: 0x34
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_DP_AUX_CH0_HPD_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_EDP_DP_AUX_CH1_HPD_0

Offset: 0x38
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_DP_AUX_CH1_HPD_0
Reset: 0x00000470 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0111,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE

Bit	Reset	Description
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	ETH3	**PM**: **0** = ETH3 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_EDP_CFG2TMC_DP_AUX_CH1_HPD_0

Offset: 0x3c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_DP_AUX_CH1_HPD_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_EDP_DP_AUX_CH2_HPD_0

Offset: 0x40
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_DP_AUX_CH2_HPD_0
Reset: 0x00000430 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0011,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	ETH3	**PM**: **0** = ETH3 **1** = RSVD1 **2** = DISPLAYB **3** = RSVD3

PADCTL_EDP_CFG2TMC_DP_AUX_CH2_HPD_0

Offset: 0x44
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_DP_AUX_CH2_HPD_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_EDP_DP_AUX_CH3_HPD_0

Offset: 0x48
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_DP_AUX_CH3_HPD_0
Reset: 0x00000430 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,0011,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	ETH2	**PM**: **0** = ETH2 **1** = RSVD1 **2** = DISPLAYA **3** = RSVD3

PADCTL_EDP_CFG2TMC_DP_AUX_CH3_HPD_0

Offset: 0x4c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_DP_AUX_CH3_HPD_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_EDP_DP_AUX_CH1_P_0

Offset: 0x50
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_DP_AUX_CH1_P_0
Reset: 0x00001570 (0bxxxx,xxxx,xxxx,xxxx,xxx1,x1x1,0111,0000)

Bit	Reset	Description
12	ENABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	I2C4	**PM**: **0** = I2C4 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_EDP_CFG2TMC_DP_AUX_CH1_P_0

Offset: 0x54
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_DP_AUX_CH1_P_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_EDP_DP_AUX_CH1_N_0
Offset: 0x58
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_DP_AUX_CH1_N_0
Reset: 0x00000570 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0111,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

Bit	Reset	Description
1:0	I2C4	**PM**: **0** = I2C4 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_EDP_CFG2TMC_DP_AUX_CH1_N_0

Offset: 0x5c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_DP_AUX_CH1_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_EDP_DP_AUX_CH2_P_0

Offset: 0x60
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_DP_AUX_CH2_P_0
Reset: 0x00001570 (0bxxxx,xxxx,xxxx,xxxx,xxx1,x1x1,0111,0000)

Bit	Reset	Description
12	ENABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	I2C7	**PM**: **0** = I2C7 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_EDP_CFG2TMC_DP_AUX_CH2_P_0

Offset: 0x64
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_DP_AUX_CH2_P_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_EDP_DP_AUX_CH2_N_0

Offset: 0x68
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_DP_AUX_CH2_N_0
Reset: 0x00000570 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0111,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	I2C7	**PM**: **0** = I2C7 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_EDP_CFG2TMC_DP_AUX_CH2_N_0

Offset: 0x6c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_DP_AUX_CH2_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_EDP_DP_AUX_CH3_P_0

Offset: 0x70
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_DP_AUX_CH3_P_0
Reset: 0x00001570 (0bxxxx,xxxx,xxxx,xxxx,xxx1,x1x1,0111,0000)

Bit	Reset	Description
12	ENABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	I2C9	**PM**: **0** = I2C9 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_EDP_CFG2TMC_DP_AUX_CH3_P_0

Offset: 0x74
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_DP_AUX_CH3_P_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_EDP_DP_AUX_CH3_N_0
Offset: 0x78
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_DP_AUX_CH3_N_0
Reset: 0x00000570 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0111,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

Bit	Reset	Description
1:0	I2C9	**PM**: **0** = I2C9 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_EDP_CFG2TMC_DP_AUX_CH3_N_0

Offset: 0x7c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_SCR_DP_AUX_CH3_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_EDP_EC_FEATURE_0

```
================================
 GLOBAL SLICE
================================
```

Offset: 0x400
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_EDP_SCR_EC_SCR_0
Reset: 0x00020001 (0b0000,0000,0000,0010,xxxx,xxxx,xx00,0001)

Bit	Reset	Description
31:16	0x2	**NUM_ERR**: Number of errors connected to this collator. This is passed as a build time option to the plugin
5:0	0x1	**NUM_ERR_SLICES**: Number of error slices supported by this error collator, does not include the GlobalSpace and is derived by ceil (NUM_ERR/32). SW shall first read this register to determine the number of slices and read the required number of Error_Status registers .

PADCTL_EDP_EC_SWRESET_0

Offset: 0x404
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_EDP_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SWRST**: 1'b1 : Issue a SW reset to the Error Collator. This will reset all the registers(Except SCR), counters and logic of the Error Collator. SW can use this bit to flush errors logged into the error collator for ex, after Boot, SC7/8 exit. 1'b0 : Do nothing, reset value. This bit is auto-cleared.

PADCTL_EDP_EC_MISSIONERR_TYPE_0

Offset: 0x408
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_EDP_SCR_EC_SCR_0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0101)

Bit	Reset	Description
5:0	0x5	**CODE**: This register indicates the fault code of the error line based on the value of MISSIONERR_INDEX Register. This can be used by a fault handling agent to triage an error without requiring device-specific code.The possible values of this field are: 6'd0 : None 6'd1 : Parity Error on internal data path 6'd2 : ECC SEC Error on internal data path 6'd3 : ECC DED Error on internal data path 6'd4 : Comparator Error 6'd5 : Register Parity Error 6'd6 : Parity Error from on-chip SRAM/Fifo 6'd7 : ECC SEC Error from on-chip SRAM/Fifo 6'd8 : ECC DED Error from on-chip SRAM/Fifo 6'd9 : Clock Monitor Error 6'd10 : Voltage Error 6'd11 : Temperature Error 6'd12 : CRC Error on internal data path 6'd16 : SW Corrected Error 6'd17 : SW Uncorrected Error 6'd18 : SW Generic Error 6'd20 : ECC SED Error on internal data path 6'd21 : ECC SED Error from on-chip SRAM/Fifo 6'd32 : Other HW Corrected Error 6'd33 : Other HW Uncorrected Error All other values : Reserved for future use.

PADCTL_EDP_EC_CURRENT_COUNTER_VALUE_0

Offset: 0x40c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_EDP_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
8:0	0x0	**VALUE**: Provides the current value of the counter corresponding to the error in MissionErr_Index Register. Default provides the value of error 0 counter. Bit[8] is the ovrerflow bit post which the counter saturates and does not counter further.

PADCTL_EDP_EC_MISSIONERR_INDEX_0

Offset: 0x414
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_EDP_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**IDX**: BINARY Encoded. For error number 32, register should be programmed with value 0x20. Write to this register with Error number will update: - MISSIONERR_TYPE Register with the Error-Code for the Error. - CURRENT_COUNTER_VALUE Register with the error's SEC/DED Counter. - MISSIONERR_USERVALUE with value of the first error_<i>_user signal. SW can use this to trage the error. number shall update the MISSIONERR_TYPE register with the error code and the Current_Counter_Value register with the value of the errors SEC/DED counter. SW can use this register to triage the error.

PADCTL_EDP_EC_CORRECTABLE_THRESHOLD_0

Offset: 0x418
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**COUNT**: Threshold value for all SEC Fault Reporting Units connected to this error collator. SEC Errors are logged once the threshold is reached and the overflow bit is set. 7'b0 : Log SEC error after receiving 1 Error. 7'b1 : Log SEC error after receiving 2 Errors. ... 7'bFF : Log SEC error after receiving 256 Errors.

PADCTL_EDP_EC_MISSIONERR_INJECT_UNLOCK_0

Offset: 0x41c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_EDP_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Writes to ERRSLICE_XXX_MISSIONERR_INJECT registers are disabled until this register is written with a value of 0xE1. This is to prevent an inadvertent safety error injection in the safety plugin due to: 1. A fault on ERRSLICE_XXX_MISSIONERR_INJECT register itself. 2. Erroneous SW. The register shall be written with a value of 0x0 to reestablish the lock after user has completed the error injection testing. 0xE1 : Unlock the MISSIONERR_INJECT Register 0x0 : Lock the MISSIONERR_INJECT Register **0** = LOCK **225** = UNLOCK

PADCTL_EDP_EC_ERRSLICE0_MISSIONERR_ENABLE_0

```
==================================
 ERROR SLICE - 0
==================================
```

Offset: 0x430
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.EDP_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.EDP_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.EDP_PADCTLREG_reg 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.EDP_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL_EDP_err_collator 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL_EDP_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_EDP_EC_ERRSLICE0_MISSIONERR_FORCE_0

Offset: 0x434
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_EDP_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.EDP_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.EDP_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL_EDP_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_EDP_EC_ERRSLICE0_MISSIONERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of MissionError_Enable register, to avoid silent dropping of errors.
Offset: 0x438
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.EDP_PADCTLREG_reg_scr_wrap was equal to 2'b10. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.EDP_PADCTLREG_reg_scr_wrap was equal to 2'b01.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.EDP_PADCTLREG_reg was equal to 2'b10. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.EDP_PADCTLREG_reg was equal to 2'b01.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_EDP_err_collator was equal to 2'b10. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_EDP_err_collator was equal to 2'b01.

PADCTL_EDP_EC_ERRSLICE0_MISSIONERR_INJECT_0

Offset: 0x43c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_EDP_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Assert the inject_error_2 output for Register Parity Error to PADCTL.EDP_PADCTLREG_reg_scr_wrap to allow for error injection. 1'b0 -> De-Asseert inject_error_2 output. **0** = DISABLE **1** = ENABLE
1	0x0	**ERR1**: 1'b1 -> Assert the inject_error_1 output for Register Parity Error to PADCTL.EDP_PADCTLREG_reg to allow for error injection. 1'b0 -> De-Asseert inject_error_1 output. **0** = DISABLE **1** = ENABLE
0	0x0	**ERR0**: 1'b1 -> Assert the inject_error_0 output for Register Parity Error to PADCTL_EDP_err_collator to allow for error injection. 1'b0 -> De-Asseert inject_error_0 output. **0** = DISABLE **1** = ENABLE

PADCTL_EDP_EC_ERRSLICE0_LATENTERR_ENABLE_0

Offset: 0x440
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.EDP_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.EDP_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.EDP_PADCTLREG_reg 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.EDP_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL_EDP_err_collator 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL_EDP_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_EDP_EC_ERRSLICE0_LATENTERR_FORCE_0

Offset: 0x444
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_EDP_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.EDP_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.EDP_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL_EDP_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_EDP_EC_ERRSLICE0_LATENTERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of LatentError_Enable register, to avoid silent dropping of errors.
Offset: 0x448
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EDP_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.EDP_PADCTLREG_reg_scr_wrap was equal to 2'b00 or 2'b11. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.EDP_PADCTLREG_reg_scr_wrap was equal to 2'b01 or 2'b10, but no latent error.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.EDP_PADCTLREG_reg was equal to 2'b00 or 2'b11. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.EDP_PADCTLREG_reg was equal to 2'b01 or 2'b10, but no latent error.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_EDP_err_collator was equal to 2'b00 or 2'b11. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_EDP_err_collator was equal to 2'b01 or 2'b10, but no latent error.

PADCTL_EDP_EC_ERRSLICE0_COUNTER_RELOAD_0

Offset: 0x450

Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_EDP_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.EDP_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
1	0x0	**ERR1**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.EDP_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
0	0x0	**ERR0**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL_EDP_err_collator 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

8.4.17 UFS PAD Control Registers

NOTE:

The UFS PAD Control Registers are collectively called PADCTL_A17 in the System Address Map. Use this Block Name (PADCTL_A17) to find the Base Address of these registers in the System Address Map.

NOTE:

The binary “x” is turned to “0” for the Reset and PROD hex value. Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as “x” or “Undefined” actually means that the bit could be either “0” or “1” after Reset, whereas a bit PROD value as “x” or “Don’t-care” means that either “0” or “1” can be written to the bit by software Initialization.

PADCTL_UFS_UFS0_RST_N_0
Offset: 0x0
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: PADCTL_UFS_SCR_SCR_UFS0_RST_N_0
Reset: 0x00002400 (0bxxxx,xxxx,0000,xxxx,x010,x1xx,x000,0000)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	UFS0	**PM**: **0** = UFS0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_UFS_CFG2TMC_UFS0_RST_N_0

Offset: 0x4
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_UFS_SCR_SCR_UFS0_RST_N_0
Reset: 0x0a00a000 (0bxxx0,1010,xxxx,xxx0,1010,xxxx,xxxx,xxxx)

Bit	Reset	Description
28:24	0xa	**CFG_CAL_DRVUP**
16:12	0xa	**CFG_CAL_DRVDN**

PADCTL_UFS_UFS0_REF_CLK_0
Offset: 0x8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_UFS_SCR_SCR_UFS0_REF_CLK_0
Reset: 0x00002400 (0bxxxx,xxxx,0000,xxxx,x010,x1xx,x000,0000)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE

Bit	Reset	Description
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	UFS0	**PM**: **0** = UFS0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_UFS_CFG2TMC_UFS0_REF_CLK_0

Offset: 0xc
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_UFS_SCR_SCR_UFS0_REF_CLK_0
Reset: 0x0a00a000 (0bxxx0,1010,xxxx,xxx0,1010,xxxx,xxxx,xxxx)

Bit	Reset	Description
28:24	0xa	**CFG_CAL_DRVUP**
16:12	0xa	**CFG_CAL_DRVDN**

PADCTL_UFS_EC_FEATURE_0

```
=================================
 GLOBAL SLICE
=================================
```

Offset: 0x400
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_UFS_SCR_EC_SCR_0
Reset: 0x00020001 (0b0000,0000,0000,0010,xxxx,xxxx,xx00,0001)

Bit	Reset	Description
31:16	0x2	**NUM_ERR**: Number of errors connected to this collator. This is passed as a build time option to the plugin

Bit	Reset	Description
5:0	0x1	**NUM_ERR_SLICES**: Number of error slices supported by this error collator, does not include the GlobalSpace and is derived by ceil (NUM_ERR/32). SW shall first read this register to determine the number of slices and read the required number of Error_Status registers .

PADCTL_UFS_EC_SWRESET_0

Offset: 0x404
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_UFS_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SWRST**: 1'b1 : Issue a SW reset to the Error Collator. This will reset all the registers(Except SCR), counters and logic of the Error Collator. SW can use this bit to flush errors logged into the error collator for ex, after Boot, SC7/8 exit. 1'b0 : Do nothing, reset value. This bit is auto-cleared.

PADCTL_UFS_EC_MISSIONERR_TYPE_0

Offset: 0x408
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_UFS_SCR_EC_SCR_0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0101)

Bit	Reset	Description
5:0	0x5	**CODE**: This register indicates the fault code of the error line based on the value of MISSIONERR_INDEX Register. This can be used by a fault handling agent to triage an error without requiring device-specific code.The possible values of this field are: 6'd0 : None 6'd1 : Parity Error on internal data path 6'd2 : ECC SEC Error on internal data path 6'd3 : ECC DED Error on internal data path 6'd4 : Comparator Error 6'd5 : Register Parity Error 6'd6 : Parity Error from on-chip SRAM/Fifo 6'd7 : ECC SEC Error from on-chip SRAM/Fifo 6'd8 : ECC DED Error from on-chip SRAM/Fifo 6'd9 : Clock Monitor Error 6'd10 : Voltage Error 6'd11 : Temperature Error 6'd12 : CRC Error on internal data path 6'd16 : SW Corrected Error 6'd17 : SW Uncorrected Error 6'd18 : SW Generic Error 6'd20 : ECC SED Error on internal data path 6'd21 : ECC SED Error from on-chip SRAM/Fifo 6'd32 : Other HW Corrected Error 6'd33 : Other HW Uncorrected Error All other values : Reserved for future use.

PADCTL_UFS_EC_CURRENT_COUNTER_VALUE_0

Offset: 0x40c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_UFS_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
8:0	0x0	**VALUE**: Provides the current value of the counter corresponding to the error in MissionErr_Index Register. Default provides the value of error 0 counter. Bit[8] is the ovrerflow bit post which the counter saturates and does not counter further.

PADCTL_UFS_EC_MISSIONERR_INDEX_0

Offset: 0x414
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_UFS_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**IDX**: BINARY Encoded. For error number 32, register should be programmed with value 0x20. Write to this register with Error number will update: - MISSIONERR_TYPE Register with the Error-Code for the Error. - CURRENT_COUNTER_VALUE Register with the error's SEC/DED Counter. - MISSIONERR_USERVALUE with value of the first error_<i>_user signal. SW can use this to trage the error. number shall update the MISSIONERR_TYPE register with the error code and the Current_Counter_Value register with the value of the errors SEC/DED counter. SW can use this register to triage the error.

PADCTL_UFS_EC_CORRECTABLE_THRESHOLD_0

Offset: 0x418
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_UFS_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**COUNT**: Threshold value for all SEC Fault Reporting Units connected to this error collator. SEC Errors are logged once the threshold is reached and the overflow bit is set. 7'b0 : Log SEC error after receiving 1 Error. 7'b1 : Log SEC error after receiving 2 Errors. ... 7'bFF : Log SEC error after receiving 256 Errors.

PADCTL_UFS_EC_MISSIONERR_INJECT_UNLOCK_0

Offset: 0x41c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_UFS_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Writes to ERRSLICE_XXX_MISSIONERR_INJECT registers are disabled until this register is written with a value of 0xE1. This is to prevent an inadvertent safety error injection in the safety plugin due to: 1. A fault on ERRSLICE_XXX_MISSIONERR_INJECT register itself. 2. Erroneous SW. The register shall be written with a value of 0x0 to reestablish the lock after user has completed the error injection testing. 0xE1 : Unlock the MISSIONERR_INJECT Register 0x0 : Lock the MISSIONERR_INJECT Register **0** = LOCK **225** = UNLOCK

PADCTL_UFS_EC_ERRSLICE0_MISSIONERR_ENABLE_0

```
==================================
 ERROR SLICE - 0
==================================
```

Offset: 0x430
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_UFS_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.UFS_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.UFS_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.UFS_PADCTLREG_reg 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.UFS_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL_UFS_err_collator 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL_UFS_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_UFS_EC_ERRSLICE0_MISSIONERR_FORCE_0

Offset: 0x434
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_UFS_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.UFS_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.UFS_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL_UFS_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_UFS_EC_ERRSLICE0_MISSIONERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of MissionError_Enable register, to avoid silent dropping of errors.
Offset: 0x438
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_UFS_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.UFS_PADCTLREG_reg_scr_wrap was equal to 2'b10. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.UFS_PADCTLREG_reg_scr_wrap was equal to 2'b01.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.UFS_PADCTLREG_reg was equal to 2'b10. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.UFS_PADCTLREG_reg was equal to 2'b01.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_UFS_err_collator was equal to 2'b10. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_UFS_err_collator was equal to 2'b01.

PADCTL_UFS_EC_ERRSLICE0_MISSIONERR_INJECT_0

Offset: 0x43c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_UFS_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Assert the inject_error_2 output for Register Parity Error to PADCTL.UFS_PADCTLREG_reg_scr_wrap to allow for error injection. 1'b0 -> De-Asseert inject_error_2 output. **0** = DISABLE **1** = ENABLE
1	0x0	**ERR1**: 1'b1 -> Assert the inject_error_1 output for Register Parity Error to PADCTL.UFS_PADCTLREG_reg to allow for error injection. 1'b0 -> De-Asseert inject_error_1 output. **0** = DISABLE **1** = ENABLE
0	0x0	**ERR0**: 1'b1 -> Assert the inject_error_0 output for Register Parity Error to PADCTL_UFS_err_collator to allow for error injection. 1'b0 -> De-Asseert inject_error_0 output. **0** = DISABLE **1** = ENABLE

PADCTL_UFS_EC_ERRSLICE0_LATENTERR_ENABLE_0

Offset: 0x440
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_UFS_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.UFS_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.UFS_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.UFS_PADCTLREG_reg 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.UFS_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL_UFS_err_collator 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL_UFS_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_UFS_EC_ERRSLICE0_LATENTERR_FORCE_0

Offset: 0x444
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_UFS_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.UFS_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.UFS_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL_UFS_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_UFS_EC_ERRSLICE0_LATENTERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of LatentError_Enable register, to avoid silent dropping of errors.
Offset: 0x448
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_UFS_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.UFS_PADCTLREG_reg_scr_wrap was equal to 2'b00 or 2'b11. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.UFS_PADCTLREG_reg_scr_wrap was equal to 2'b01 or 2'b10, but no latent error.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.UFS_PADCTLREG_reg was equal to 2'b00 or 2'b11. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.UFS_PADCTLREG_reg was equal to 2'b01 or 2'b10, but no latent error.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_UFS_err_collator was equal to 2'b00 or 2'b11. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_UFS_err_collator was equal to 2'b01 or 2'b10, but no latent error.

PADCTL_UFS_EC_ERRSLICE0_COUNTER_RELOAD_0

Offset: 0x450

Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_UFS_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.UFS_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
1	0x0	**ERR1**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.UFS_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
0	0x0	**ERR0**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL_UFS_err_collator 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

8.4.18 PEX_CTL_2 PAD Control Registers

NOTE:

The PEX_CTL_2 PAD Control Registers are collectively called PADCTL_A20 in the System Address Map.

The binary “x” is turned to “0” for the Reset and PROD hex value. Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as “x” or “Undefined” actually means that the bit could be either “0” or “1” after Reset, whereas a bit PROD value as “x” or “Don’t-care” means that either “0” or “1” can be written to the bit by software Initialization.

PADCTL_PEX_CTL_2_PEX_L5_CLKREQ_N_0
Offset: 0x0
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_PEX_CTL_2_SCR_SCR_PEX_L5_CLKREQ_N_0
Reset: 0x00000560 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0110,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	PE5	**PM**: **0** = PE5 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_PEX_CTL_2_CFG2TMC_PEX_L5_CLKREQ_N_0

Offset: 0x4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_2_SCR_SCR_PEX_L5_CLKREQ_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**

Bit	Reset	Description
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_PEX_CTL_2_PEX_L5_RST_N_0

Offset: 0x8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_2_SCR_SCR_PEX_L5_RST_N_0
Reset: 0x00000520 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0010,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	PE5	**PM**: **0** = PE5 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_PEX_CTL_2_CFG2TMC_PEX_L5_RST_N_0

Offset: 0xc
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_2_SCR_SCR_PEX_L5_RST_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_PEX_CTL_2_PEX_L6_CLKREQ_N_0

Offset: 0x10
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_2_SCR_SCR_PEX_L6_CLKREQ_N_0
Reset: 0x00000560 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0110,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE

Bit	Reset	Description
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	PE6	**PM**: **0** = PE6 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_PEX_CTL_2_CFG2TMC_PEX_L6_CLKREQ_N_0

Offset: 0x14
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_2_SCR_SCR_PEX_L6_CLKREQ_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_PEX_CTL_2_PEX_L6_RST_N_0

Offset: 0x18
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_2_SCR_SCR_PEX_L6_RST_N_0
Reset: 0x00000520 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0010,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	PE6	**PM**: **0** = PE6 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_PEX_CTL_2_CFG2TMC_PEX_L6_RST_N_0

Offset: 0x1c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_2_SCR_SCR_PEX_L6_RST_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_PEX_CTL_2_EC_FEATURE_0

```
================================
 GLOBAL SLICE
================================
```

Offset: 0x400
Read/Write: RO

Parity Protection: N
Shadow: N
SCR Protection: PADCTL_PEX_CTL_2_SCR_EC_SCR_0
Reset: 0x00020001 (0b0000,0000,0000,0010,xxxx,xxxx,xx00,0001)

Bit	Reset	Description
31:16	0x2	**NUM_ERR**: Number of errors connected to this collator. This is passed as a build time option to the plugin
5:0	0x1	**NUM_ERR_SLICES**: Number of error slices supported by this error collator, does not include the GlobalSpace and is derived by ceil (NUM_ERR/32). SW shall first read this register to determine the number of slices and read the required number of Error_Status registers .

PADCTL_PEX_CTL_2_EC_SWRESET_0

Offset: 0x404
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_PEX_CTL_2_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SWRST**: 1'b1 : Issue a SW reset to the Error Collator. This will reset all the registers(Except SCR), counters and logic of the Error Collator. SW can use this bit to flush errors logged into the error collator for ex, after Boot, SC7/8 exit. 1'b0 : Do nothing, reset value. This bit is auto-cleared.

PADCTL_PEX_CTL_2_EC_MISSIONERR_TYPE_0

Offset: 0x408
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_PEX_CTL_2_SCR_EC_SCR_0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0101)

Bit	Reset	Description
5:0	0x5	**CODE**: This register indicates the fault code of the error line based on the value of MISSIONERR_INDEX Register. This can be used by a fault handling agent to triage an error without requiring device-specific code.The possible values of this field are: 6'd0 : None 6'd1 : Parity Error on internal data path 6'd2 : ECC SEC Error on internal data path 6'd3 : ECC DED Error on internal data path 6'd4 : Comparator Error 6'd5 : Register Parity Error 6'd6 : Parity Error from on-chip SRAM/Fifo 6'd7 : ECC SEC Error from on-chip SRAM/Fifo 6'd8 : ECC DED Error from on-chip SRAM/Fifo 6'd9 : Clock Monitor Error 6'd10 : Voltage Error 6'd11 : Temperature Error 6'd12 : CRC Error on internal data path 6'd16 : SW Corrected Error 6'd17 : SW Uncorrected Error 6'd18 : SW Generic Error 6'd20 : ECC SED Error on internal data path 6'd21 : ECC SED Error from on-chip SRAM/Fifo 6'd32 : Other HW Corrected Error 6'd33 : Other HW Uncorrected Error All other values : Reserved for future use.

PADCTL_PEX_CTL_2_EC_CURRENT_COUNTER_VALUE_0

Offset: 0x40c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_PEX_CTL_2_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
8:0	0x0	**VALUE**: Provides the current value of the counter corresponding to the error in MissionErr_Index Register. Default provides the value of error 0 counter. Bit[8] is the ovrerflow bit post which the counter saturates and does not counter further.

PADCTL_PEX_CTL_2_EC_MISSIONERR_INDEX_0

Offset: 0x414
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_PEX_CTL_2_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**IDX**: BINARY Encoded. For error number 32, register should be programmed with value 0x20. Write to this register with Error number will update: - MISSIONERR_TYPE Register with the Error-Code for the Error. - CURRENT_COUNTER_VALUE Register with the error's SEC/DED Counter. - MISSIONERR_USERVALUE with value of the first error_<i>_user signal. SW can use this to trage the error. number shall update the MISSIONERR_TYPE register with the error code and the Current_Counter_Value register with the value of the errors SEC/DED counter. SW can use this register to triage the error.

PADCTL_PEX_CTL_2_EC_CORRECTABLE_THRESHOLD_0

Offset: 0x418
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_2_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**COUNT**: Threshold value for all SEC Fault Reporting Units connected to this error collator. SEC Errors are logged once the threshold is reached and the overflow bit is set. 7'b0 : Log SEC error after receiving 1 Error. 7'b1 : Log SEC error after receiving 2 Errors. ... 7'bFF : Log SEC error after receiving 256 Errors.

PADCTL_PEX_CTL_2_EC_MISSIONERR_INJECT_UNLOCK_0

Offset: 0x41c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_PEX_CTL_2_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Writes to ERRSLICE_XXX_MISSIONERR_INJECT registers are disabled until this register is written with a value of 0xE1. This is to prevent an inadvertent safety error injection in the safety plugin due to: 1. A fault on ERRSLICE_XXX_MISSIONERR_INJECT register itself. 2. Erroneous SW. The register shall be written with a value of 0x0 to reestablish the lock after user has completed the error injection testing. 0xE1 : Unlock the MISSIONERR_INJECT Register 0x0 : Lock the MISSIONERR_INJECT Register **0** = LOCK **225** = UNLOCK

PADCTL_PEX_CTL_2_EC_ERRSLICE0_MISSIONERR_ENABLE_0

```
==================================
 ERROR SLICE - 0
==================================
```

Offset: 0x430
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_2_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.PEX_CTL_2_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.PEX_CTL_2_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.PEX_CTL_2_PADCTLREG_reg 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.PEX_CTL_2_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL_PEX_CTL_2_err_collator 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL_PEX_CTL_2_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_PEX_CTL_2_EC_ERRSLICE0_MISSIONERR_FORCE_0

Offset: 0x434
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_PEX_CTL_2_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.PEX_CTL_2_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.PEX_CTL_2_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL_PEX_CTL_2_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_PEX_CTL_2_EC_ERRSLICE0_MISSIONERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of MissionError_Enable register, to avoid silent dropping of errors.
Offset: 0x438
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_2_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.PEX_CTL_2_PADCTLREG_reg_scr_wrap was equal to 2'b10. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.PEX_CTL_2_PADCTLREG_reg_scr_wrap was equal to 2'b01.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.PEX_CTL_2_PADCTLREG_reg was equal to 2'b10. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.PEX_CTL_2_PADCTLREG_reg was equal to 2'b01.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_PEX_CTL_2_err_collator was equal to 2'b10. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_PEX_CTL_2_err_collator was equal to 2'b01.

PADCTL_PEX_CTL_2_EC_ERRSLICE0_MISSIONERR_INJECT_0

Offset: 0x43c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_PEX_CTL_2_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Assert the inject_error_2 output for Register Parity Error to PADCTL.PEX_CTL_2_PADCTLREG_reg_scr_wrap to allow for error injection. 1'b0 -> De-Asseert inject_error_2 output. **0** = DISABLE **1** = ENABLE
1	0x0	**ERR1**: 1'b1 -> Assert the inject_error_1 output for Register Parity Error to PADCTL.PEX_CTL_2_PADCTLREG_reg to allow for error injection. 1'b0 -> De-Asseert inject_error_1 output. **0** = DISABLE **1** = ENABLE
0	0x0	**ERR0**: 1'b1 -> Assert the inject_error_0 output for Register Parity Error to PADCTL_PEX_CTL_2_err_collator to allow for error injection. 1'b0 -> De-Asseert inject_error_0 output. **0** = DISABLE **1** = ENABLE

PADCTL_PEX_CTL_2_EC_ERRSLICE0_LATENTERR_ENABLE_0

Offset: 0x440
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_2_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.PEX_CTL_2_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.PEX_CTL_2_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.PEX_CTL_2_PADCTLREG_reg 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.PEX_CTL_2_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL_PEX_CTL_2_err_collator 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL_PEX_CTL_2_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_PEX_CTL_2_EC_ERRSLICE0_LATENTERR_FORCE_0

Offset: 0x444
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_PEX_CTL_2_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.PEX_CTL_2_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.PEX_CTL_2_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL_PEX_CTL_2_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_PEX_CTL_2_EC_ERRSLICE0_LATENTERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of LatentError_Enable register, to avoid silent dropping of errors.
Offset: 0x448
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_2_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.PEX_CTL_2_PADCTLREG_reg_scr_wrap was equal to 2'b00 or 2'b11. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.PEX_CTL_2_PADCTLREG_reg_scr_wrap was equal to 2'b01 or 2'b10, but no latent error.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.PEX_CTL_2_PADCTLREG_reg was equal to 2'b00 or 2'b11. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.PEX_CTL_2_PADCTLREG_reg was equal to 2'b01 or 2'b10, but no latent error.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_PEX_CTL_2_err_collator was equal to 2'b00 or 2'b11. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_PEX_CTL_2_err_collator was equal to 2'b01 or 2'b10, but no latent error.

PADCTL_PEX_CTL_2_EC_ERRSLICE0_COUNTER_RELOAD_0

Offset: 0x450
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_PEX_CTL_2_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.PEX_CTL_2_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
1	0x0	**ERR1**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.PEX_CTL_2_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
0	0x0	**ERR0**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL_PEX_CTL_2_err_collator 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

8.4.19 EQOS PAD Control Registers

NOTE:

The EQOS PAD Control Registers are collectively called PADCTL_A21 in the System Address Map.

The binary “x” is turned to “0” for the Reset and PROD hex value. Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as “x” or “Undefined” actually means that the bit could be either “0” or “1” after Reset, whereas a bit PROD value as “x” or “Don’t-care” means that either “0” or “1” can be written to the bit by software Initialization.

PADCTL_EQOS_EQOS_TD3_0
Offset: 0x0
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_EQOS_SCR_SCR_EQOS_TD3_0
Reset: 0x00002410 (0bxxxx,xxxx,0000,xxxx,x010,x1xx,x001,0000)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	EQOS	**PM**: **0** = EQOS **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_EQOS_EQOS_TD2_0

Offset: 0x8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EQOS_SCR_SCR_EQOS_TD2_0
Reset: 0x00002410 (0bxxxx,xxxx,0000,xxxx,x010,x1xx,x001,0000)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	EQOS	**PM**: **0** = EQOS **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_EQOS_EQOS_TD1_0

Offset: 0x10
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EQOS_SCR_SCR_EQOS_TD1_0
Reset: 0x00002410 (0bxxxx,xxxx,0000,xxxx,x010,x1xx,x001,0000)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	EQOS	**PM**: **0** = EQOS **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_EQOS_EQOS_TD0_0

Offset: 0x18
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EQOS_SCR_SCR_EQOS_TD0_0
Reset: 0x00002410 (0bxxxx,xxxx,0000,xxxx,x010,x1xx,x001,0000)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	EQOS	**PM**: **0** = EQOS **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_EQOS_EQOS_RD3_0

Offset: 0x20
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EQOS_SCR_SCR_EQOS_RD3_0
Reset: 0x00002450 (0bxxxx,xxxx,0000,xxxx,x010,x1xx,x101,0000)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	EQOS	**PM**: **0** = EQOS **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_EQOS_EQOS_RD2_0

Offset: 0x28
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EQOS_SCR_SCR_EQOS_RD2_0
Reset: 0x00002450 (0bxxxx,xxxx,0000,xxxx,x010,x1xx,x101,0000)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	EQOS	**PM**: **0** = EQOS **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_EQOS_EQOS_RD1_0

Offset: 0x30
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EQOS_SCR_SCR_EQOS_RD1_0
Reset: 0x00002450 (0bxxxx,xxxx,0000,xxxx,x010,x1xx,x101,0000)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	EQOS	**PM**: **0** = EQOS **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_EQOS_EQOS_SMA_MDIO_0

Offset: 0x38
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EQOS_SCR_SCR_EQOS_SMA_MDIO_0
Reset: 0x00002450 (0bxxxx,xxxx,0000,xxxx,x010,x1xx,x101,0000)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	EQOS	**PM**: **0** = EQOS **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_EQOS_EQOS_RD0_0

Offset: 0x40
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EQOS_SCR_SCR_EQOS_RD0_0
Reset: 0x00002450 (0bxxxx,xxxx,0000,xxxx,x010,x1xx,x101,0000)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	EQOS	**PM**: **0** = EQOS **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_EQOS_EQOS_SMA_MDC_0

Offset: 0x48
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EQOS_SCR_SCR_EQOS_SMA_MDC_0
Reset: 0x00002410 (0bxxxx,xxxx,0000,xxxx,x010,x1xx,x001,0000)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	EQOS	**PM**: **0** = EQOS **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_EQOS_EQOS_COMP_0

Offset: 0x50
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_EQOS_SCR_SCR_EQOS_COMP_0
Reset: 0x00002000 (0bxxxx,xxxx,0000,xxxx,x01x,xxxx,xxx0,xx00)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
1:0	EQOS	**PM**: **0** = EQOS **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_EQOS_EQOS_TXC_0

Offset: 0x58
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EQOS_SCR_SCR_EQOS_TXC_0
Reset: 0x00002410 (0bxxxx,xxxx,0000,xxxx,x010,x1xx,x001,0000)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO

Bit	Reset	Description
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	EQOS	**PM**: **0** = EQOS **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_EQOS_EQOS_RXC_0

Offset: 0x60
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EQOS_SCR_SCR_EQOS_RXC_0
Reset: 0x00002450 (0bxxxx,xxxx,0000,xxxx,x010,x1xx,x101,0000)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO

Bit	Reset	Description
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	EQOS	**PM**: **0** = EQOS **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_EQOS_EQOS_TX_CTL_0

Offset: 0x68
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EQOS_SCR_SCR_EQOS_TX_CTL_0
Reset: 0x00002410 (0bxxxx,xxxx,0000,xxxx,x010,x1xx,x001,0000)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO

Bit	Reset	Description
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	EQOS	**PM**: **0** = EQOS **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_EQOS_EQOS_RX_CTL_0

Offset: 0x70
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EQOS_SCR_SCR_EQOS_RX_CTL_0
Reset: 0x00002450 (0bxxxx,xxxx,0000,xxxx,x010,x1xx,x101,0000)

Bit	Reset	Description
23:20	0x0	**RFU_IN**
14:13	COMP_DRIVE_2X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO

Bit	Reset	Description
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	EQOS	**PM**: **0** = EQOS **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_EQOS_EC_FEATURE_0

```
================================
 GLOBAL SLICE
================================
```

Offset: 0x400
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_EQOS_SCR_EC_SCR_0
Reset: 0x00020001 (0b0000,0000,0000,0010,xxxx,xxxx,xx00,0001)

Bit	Reset	Description
31:16	0x2	**NUM_ERR**: Number of errors connected to this collator. This is passed as a build time option to the plugin
5:0	0x1	**NUM_ERR_SLICES**: Number of error slices supported by this error collator, does not include the GlobalSpace and is derived by ceil (NUM_ERR/32). SW shall first read this register to determine the number of slices and read the required number of Error_Status registers .

PADCTL_EQOS_EC_SWRESET_0

Offset: 0x404

Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_EQOS_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SWRST**: 1'b1 : Issue a SW reset to the Error Collator. This will reset all the registers(Except SCR), counters and logic of the Error Collator. SW can use this bit to flush errors logged into the error collator for ex, after Boot, SC7/8 exit. 1'b0 : Do nothing, reset value. This bit is auto-cleared.

PADCTL_EQOS_EC_MISSIONERR_TYPE_0

Offset: 0x408
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_EQOS_SCR_EC_SCR_0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0101)

Bit	Reset	Description
5:0	0x5	**CODE**: This register indicates the fault code of the error line based on the value of MISSIONERR_INDEX Register. This can be used by a fault handling agent to triage an error without requiring device-specific code.The possible values of this field are: 6'd0 : None 6'd1 : Parity Error on internal data path 6'd2 : ECC SEC Error on internal data path 6'd3 : ECC DED Error on internal data path 6'd4 : Comparator Error 6'd5 : Register Parity Error 6'd6 : Parity Error from on-chip SRAM/Fifo 6'd7 : ECC SEC Error from on-chip SRAM/Fifo 6'd8 : ECC DED Error from on-chip SRAM/Fifo 6'd9 : Clock Monitor Error 6'd10 : Voltage Error 6'd11 : Temperature Error 6'd12 : CRC Error on internal data path 6'd16 : SW Corrected Error 6'd17 : SW Uncorrected Error 6'd18 : SW Generic Error 6'd20 : ECC SED Error on internal data path 6'd21 : ECC SED Error from on-chip SRAM/Fifo 6'd32 : Other HW Corrected Error 6'd33 : Other HW Uncorrected Error All other values : Reserved for future use.

PADCTL_EQOS_EC_CURRENT_COUNTER_VALUE_0

Offset: 0x40c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_EQOS_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
8:0	0x0	**VALUE**: Provides the current value of the counter corresponding to the error in MissionErr_Index Register. Default provides the value of error 0 counter. Bit[8] is the ovrerflow bit post which the counter saturates and does not counter further.

PADCTL_EQOS_EC_MISSIONERR_INDEX_0

Offset: 0x414
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EQOS_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**IDX**: BINARY Encoded. For error number 32, register should be programmed with value 0x20. Write to this register with Error number will update: - MISSIONERR_TYPE Register with the Error-Code for the Error. - CURRENT_COUNTER_VALUE Register with the error's SEC/DED Counter. - MISSIONERR_USERVALUE with value of the first error_<i>_user signal. SW can use this to trage the error. number shall update the MISSIONERR_TYPE register with the error code and the Current_Counter_Value register with the value of the errors SEC/DED counter. SW can use this register to triage the error.

PADCTL_EQOS_EC_CORRECTABLE_THRESHOLD_0

Offset: 0x418
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_EQOS_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**COUNT**: Threshold value for all SEC Fault Reporting Units connected to this error collator. SEC Errors are logged once the threshold is reached and the overflow bit is set. 7'b0 : Log SEC error after receiving 1 Error. 7'b1 : Log SEC error after receiving 2 Errors. ... 7'bFF : Log SEC error after receiving 256 Errors.

PADCTL_EQOS_EC_MISSIONERR_INJECT_UNLOCK_0

Offset: 0x41c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_EQOS_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Writes to ERRSLICE_XXX_MISSIONERR_INJECT registers are disabled until this register is written with a value of 0xE1. This is to prevent an inadvertent safety error injection in the safety plugin due to: 1. A fault on ERRSLICE_XXX_MISSIONERR_INJECT register itself. 2. Erroneous SW. The register shall be written with a value of 0x0 to reestablish the lock after user has completed the error injection testing. 0xE1 : Unlock the MISSIONERR_INJECT Register 0x0 : Lock the MISSIONERR_INJECT Register **0** = LOCK **225** = UNLOCK

PADCTL_EQOS_EC_ERRSLICE0_MISSIONERR_ENABLE_0

```
==================================
 ERROR SLICE - 0
==================================
```

Offset: 0x430
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EQOS_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.EQOS_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.EQOS_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.EQOS_PADCTLREG_reg 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.EQOS_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL_EQOS_err_collator 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL_EQOS_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_EQOS_EC_ERRSLICE0_MISSIONERR_FORCE_0

Offset: 0x434
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_EQOS_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.EQOS_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.EQOS_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL_EQOS_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_EQOS_EC_ERRSLICE0_MISSIONERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of MissionError_Enable register, to avoid silent dropping of errors.
Offset: 0x438
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EQOS_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.EQOS_PADCTLREG_reg_scr_wrap was equal to 2'b10. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.EQOS_PADCTLREG_reg_scr_wrap was equal to 2'b01.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.EQOS_PADCTLREG_reg was equal to 2'b10. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.EQOS_PADCTLREG_reg was equal to 2'b01.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_EQOS_err_collator was equal to 2'b10. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_EQOS_err_collator was equal to 2'b01.

PADCTL_EQOS_EC_ERRSLICE0_MISSIONERR_INJECT_0

Offset: 0x43c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_EQOS_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Assert the inject_error_2 output for Register Parity Error to PADCTL.EQOS_PADCTLREG_reg_scr_wrap to allow for error injection. 1'b0 -> De-Asseert inject_error_2 output. **0** = DISABLE **1** = ENABLE
1	0x0	**ERR1**: 1'b1 -> Assert the inject_error_1 output for Register Parity Error to PADCTL.EQOS_PADCTLREG_reg to allow for error injection. 1'b0 -> De-Asseert inject_error_1 output. **0** = DISABLE **1** = ENABLE
0	0x0	**ERR0**: 1'b1 -> Assert the inject_error_0 output for Register Parity Error to PADCTL_EQOS_err_collator to allow for error injection. 1'b0 -> De-Asseert inject_error_0 output. **0** = DISABLE **1** = ENABLE

PADCTL_EQOS_EC_ERRSLICE0_LATENTERR_ENABLE_0

Offset: 0x440
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_EQOS_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.EQOS_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.EQOS_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.EQOS_PADCTLREG_reg 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.EQOS_PADCTLREG_reg **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
0	0x1	**ERR0**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL_EQOS_err_collator 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL_EQOS_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_EQOS_EC_ERRSLICE0_LATENTERR_FORCE_0

Offset: 0x444
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_EQOS_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.EQOS_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.EQOS_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL_EQOS_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_EQOS_EC_ERRSLICE0_LATENTERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of LatentError_Enable register, to avoid silent dropping of errors.
Offset: 0x448
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: PADCTL_EQOS_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.EQOS_PADCTLREG_reg_scr_wrap was equal to 2'b00 or 2'b11. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.EQOS_PADCTLREG_reg_scr_wrap was equal to 2'b01 or 2'b10, but no latent error.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.EQOS_PADCTLREG_reg was equal to 2'b00 or 2'b11. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.EQOS_PADCTLREG_reg was equal to 2'b01 or 2'b10, but no latent error.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_EQOS_err_collator was equal to 2'b00 or 2'b11. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_EQOS_err_collator was equal to 2'b01 or 2'b10, but no latent error.

PADCTL_EQOS_EC_ERRSLICE0_COUNTER_RELOAD_0

Offset: 0x450
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_EQOS_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.EQOS_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
1	0x0	**ERR1**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.EQOS_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

Bit	Reset	Description
0	0x0	**ERR0**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL_EQOS_err_collator 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

8.4.20 PEX_CTL_3 PAD Control Registers

NOTE:

The PEX_CTL_3 PAD Control Registers are collectively called PADCTL_A25 in the System Address Map.

The binary “x” is turned to “0” for the Reset and PROD hex value. Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as “x” or “Undefined” actually means that the bit could be either “0” or “1” after Reset, whereas a bit PROD value as “x” or “Don’t-care” means that either “0” or “1” can be written to the bit by software Initialization.

PADCTL_PEX_CTL_3_PEX_L10_CLKREQ_N_0
Offset: 0x0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_3_SCR_SCR_PEX_L10_CLKREQ_N_0
Reset: 0x00000560 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0110,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	PE10	**PM**: **0** = PE10 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_PEX_CTL_3_CFG2TMC_PEX_L10_CLKREQ_N_0

Offset: 0x4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_3_SCR_SCR_PEX_L10_CLKREQ_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_PEX_CTL_3_PEX_L10_RST_N_0

Offset: 0x8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_3_SCR_SCR_PEX_L10_RST_N_0
Reset: 0x00000520 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0010,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	PE10	**PM**: **0** = PE10 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_PEX_CTL_3_CFG2TMC_PEX_L10_RST_N_0

Offset: 0xc
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_3_SCR_SCR_PEX_L10_RST_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_PEX_CTL_3_PEX_L7_CLKREQ_N_0

Offset: 0x10
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_3_SCR_SCR_PEX_L7_CLKREQ_N_0
Reset: 0x00000560 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0110,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	PE7	**PM**: **0** = PE7 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_PEX_CTL_3_CFG2TMC_PEX_L7_CLKREQ_N_0

Offset: 0x14
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_PEX_CTL_3_SCR_SCR_PEX_L7_CLKREQ_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_PEX_CTL_3_PEX_L7_RST_N_0

Offset: 0x18
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_3_SCR_SCR_PEX_L7_RST_N_0
Reset: 0x00000520 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0010,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

Bit	Reset	Description
1:0	PE7	**PM**: **0** = PE7 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_PEX_CTL_3_CFG2TMC_PEX_L7_RST_N_0

Offset: 0x1c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_3_SCR_SCR_PEX_L7_RST_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_PEX_CTL_3_PEX_L8_CLKREQ_N_0

Offset: 0x20
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_3_SCR_SCR_PEX_L8_CLKREQ_N_0
Reset: 0x00000560 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0110,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	PE8	**PM**: **0** = PE8 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_PEX_CTL_3_CFG2TMC_PEX_L8_CLKREQ_N_0

Offset: 0x24
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_3_SCR_SCR_PEX_L8_CLKREQ_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_PEX_CTL_3_PEX_L8_RST_N_0

Offset: 0x28
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_3_SCR_SCR_PEX_L8_RST_N_0
Reset: 0x00000520 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0010,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	PE8	**PM**: **0** = PE8 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_PEX_CTL_3_CFG2TMC_PEX_L8_RST_N_0

Offset: 0x2c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_3_SCR_SCR_PEX_L8_RST_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_PEX_CTL_3_PEX_L9_CLKREQ_N_0

Offset: 0x30
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_3_SCR_SCR_PEX_L9_CLKREQ_N_0
Reset: 0x00000560 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0110,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	PE9	**PM**: **0** = PE9 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_PEX_CTL_3_CFG2TMC_PEX_L9_CLKREQ_N_0

Offset: 0x34
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: PADCTL_PEX_CTL_3_SCR_SCR_PEX_L9_CLKREQ_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_PEX_CTL_3_PEX_L9_RST_N_0
Offset: 0x38
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_3_SCR_SCR_PEX_L9_RST_N_0
Reset: 0x00000520 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x1,0010,0000)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	ENABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	PASSTHROUGH	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

Bit	Reset	Description
1:0	PE9	**PM**: **0** = PE9 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_PEX_CTL_3_CFG2TMC_PEX_L9_RST_N_0

Offset: 0x3c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_3_SCR_SCR_PEX_L9_RST_N_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_PEX_CTL_3_EC_FEATURE_0

```
================================
 GLOBAL SLICE
================================
```

Offset: 0x400
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_PEX_CTL_3_SCR_EC_SCR_0
Reset: 0x00020001 (0b0000,0000,0000,0010,xxxx,xxxx,xx00,0001)

Bit	Reset	Description
31:16	0x2	**NUM_ERR**: Number of errors connected to this collator. This is passed as a build time option to the plugin
5:0	0x1	**NUM_ERR_SLICES**: Number of error slices supported by this error collator, does not include the GlobalSpace and is derived by ceil (NUM_ERR/32). SW shall first read this register to determine the number of slices and read the required number of Error_Status registers .

PADCTL_PEX_CTL_3_EC_SWRESET_0

Offset: 0x404
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_PEX_CTL_3_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SWRST**: 1'b1 : Issue a SW reset to the Error Collator. This will reset all the registers(Except SCR), counters and logic of the Error Collator. SW can use this bit to flush errors logged into the error collator for ex, after Boot, SC7/8 exit. 1'b0 : Do nothing, reset value. This bit is auto-cleared.

PADCTL_PEX_CTL_3_EC_MISSIONERR_TYPE_0

Offset: 0x408
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_PEX_CTL_3_SCR_EC_SCR_0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0101)

Bit	Reset	Description
5:0	0x5	**CODE**: This register indicates the fault code of the error line based on the value of MISSIONERR_INDEX Register. This can be used by a fault handling agent to triage an error without requiring device-specific code.The possible values of this field are: 6'd0 : None 6'd1 : Parity Error on internal data path 6'd2 : ECC SEC Error on internal data path 6'd3 : ECC DED Error on internal data path 6'd4 : Comparator Error 6'd5 : Register Parity Error 6'd6 : Parity Error from on-chip SRAM/Fifo 6'd7 : ECC SEC Error from on-chip SRAM/Fifo 6'd8 : ECC DED Error from on-chip SRAM/Fifo 6'd9 : Clock Monitor Error 6'd10 : Voltage Error 6'd11 : Temperature Error 6'd12 : CRC Error on internal data path 6'd16 : SW Corrected Error 6'd17 : SW Uncorrected Error 6'd18 : SW Generic Error 6'd20 : ECC SED Error on internal data path 6'd21 : ECC SED Error from on-chip SRAM/Fifo 6'd32 : Other HW Corrected Error 6'd33 : Other HW Uncorrected Error All other values : Reserved for future use.

PADCTL_PEX_CTL_3_EC_CURRENT_COUNTER_VALUE_0

Offset: 0x40c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_PEX_CTL_3_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
8:0	0x0	**VALUE**: Provides the current value of the counter corresponding to the error in MissionErr_Index Register. Default provides the value of error 0 counter. Bit[8] is the ovrerflow bit post which the counter saturates and does not counter further.

PADCTL_PEX_CTL_3_EC_MISSIONERR_INDEX_0

Offset: 0x414
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_PEX_CTL_3_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**IDX**: BINARY Encoded. For error number 32, register should be programmed with value 0x20. Write to this register with Error number will update: - MISSIONERR_TYPE Register with the Error-Code for the Error. - CURRENT_COUNTER_VALUE Register with the error's SEC/DED Counter. - MISSIONERR_USERVALUE with value of the first error_<i>_user signal. SW can use this to trage the error. number shall update the MISSIONERR_TYPE register with the error code and the Current_Counter_Value register with the value of the errors SEC/DED counter. SW can use this register to triage the error.

PADCTL_PEX_CTL_3_EC_CORRECTABLE_THRESHOLD_0

Offset: 0x418
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_3_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**COUNT**: Threshold value for all SEC Fault Reporting Units connected to this error collator. SEC Errors are logged once the threshold is reached and the overflow bit is set. 7'b0 : Log SEC error after receiving 1 Error. 7'b1 : Log SEC error after receiving 2 Errors. ... 7'bFF : Log SEC error after receiving 256 Errors.

PADCTL_PEX_CTL_3_EC_MISSIONERR_INJECT_UNLOCK_0

Offset: 0x41c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_PEX_CTL_3_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Writes to ERRSLICE_XXX_MISSIONERR_INJECT registers are disabled until this register is written with a value of 0xE1. This is to prevent an inadvertent safety error injection in the safety plugin due to: 1. A fault on ERRSLICE_XXX_MISSIONERR_INJECT register itself. 2. Erroneous SW. The register shall be written with a value of 0x0 to reestablish the lock after user has completed the error injection testing. 0xE1 : Unlock the MISSIONERR_INJECT Register 0x0 : Lock the MISSIONERR_INJECT Register **0** = LOCK **225** = UNLOCK

PADCTL_PEX_CTL_3_EC_ERRSLICE0_MISSIONERR_ENABLE_0

```
==================================
 ERROR SLICE - 0
==================================
```

Offset: 0x430
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_3_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.PEX_CTL_3_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.PEX_CTL_3_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.PEX_CTL_3_PADCTLREG_reg 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.PEX_CTL_3_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL_PEX_CTL_3_err_collator 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL_PEX_CTL_3_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_PEX_CTL_3_EC_ERRSLICE0_MISSIONERR_FORCE_0

Offset: 0x434
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_PEX_CTL_3_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.PEX_CTL_3_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.PEX_CTL_3_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL_PEX_CTL_3_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_PEX_CTL_3_EC_ERRSLICE0_MISSIONERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of MissionError_Enable register, to avoid silent dropping of errors.
Offset: 0x438
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_3_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.PEX_CTL_3_PADCTLREG_reg_scr_wrap was equal to 2'b10. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.PEX_CTL_3_PADCTLREG_reg_scr_wrap was equal to 2'b01.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.PEX_CTL_3_PADCTLREG_reg was equal to 2'b10. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.PEX_CTL_3_PADCTLREG_reg was equal to 2'b01.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_PEX_CTL_3_err_collator was equal to 2'b10. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_PEX_CTL_3_err_collator was equal to 2'b01.

PADCTL_PEX_CTL_3_EC_ERRSLICE0_MISSIONERR_INJECT_0

Offset: 0x43c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_PEX_CTL_3_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Assert the inject_error_2 output for Register Parity Error to PADCTL.PEX_CTL_3_PADCTLREG_reg_scr_wrap to allow for error injection. 1'b0 -> De-Asseert inject_error_2 output. **0** = DISABLE **1** = ENABLE
1	0x0	**ERR1**: 1'b1 -> Assert the inject_error_1 output for Register Parity Error to PADCTL.PEX_CTL_3_PADCTLREG_reg to allow for error injection. 1'b0 -> De-Asseert inject_error_1 output. **0** = DISABLE **1** = ENABLE
0	0x0	**ERR0**: 1'b1 -> Assert the inject_error_0 output for Register Parity Error to PADCTL_PEX_CTL_3_err_collator to allow for error injection. 1'b0 -> De-Asseert inject_error_0 output. **0** = DISABLE **1** = ENABLE

PADCTL_PEX_CTL_3_EC_ERRSLICE0_LATENTERR_ENABLE_0

Offset: 0x440
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_3_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.PEX_CTL_3_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.PEX_CTL_3_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.PEX_CTL_3_PADCTLREG_reg 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.PEX_CTL_3_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL_PEX_CTL_3_err_collator 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL_PEX_CTL_3_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_PEX_CTL_3_EC_ERRSLICE0_LATENTERR_FORCE_0

Offset: 0x444
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_PEX_CTL_3_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.PEX_CTL_3_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.PEX_CTL_3_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL_PEX_CTL_3_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_PEX_CTL_3_EC_ERRSLICE0_LATENTERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of LatentError_Enable register, to avoid silent dropping of errors.
Offset: 0x448
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_PEX_CTL_3_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.PEX_CTL_3_PADCTLREG_reg_scr_wrap was equal to 2'b00 or 2'b11. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.PEX_CTL_3_PADCTLREG_reg_scr_wrap was equal to 2'b01 or 2'b10, but no latent error.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.PEX_CTL_3_PADCTLREG_reg was equal to 2'b00 or 2'b11. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.PEX_CTL_3_PADCTLREG_reg was equal to 2'b01 or 2'b10, but no latent error.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_PEX_CTL_3_err_collator was equal to 2'b00 or 2'b11. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_PEX_CTL_3_err_collator was equal to 2'b01 or 2'b10, but no latent error.

PADCTL_PEX_CTL_3_EC_ERRSLICE0_COUNTER_RELOAD_0

Offset: 0x450
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_PEX_CTL_3_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.PEX_CTL_3_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
1	0x0	**ERR1**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.PEX_CTL_3_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
0	0x0	**ERR0**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL_PEX_CTL_3_err_collator 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

8.4.21 G7 PAD Control Registers

NOTE:

The G7 PAD Control Registers are collectively called PADCTL_A24 in the System Address Map.

The binary “x” is turned to “0” for the Reset and PROD hex value. Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as “x” or “Undefined” actually means that the bit could be either “0” or “1” after Reset, whereas a bit PROD value as “x” or “Don’t-care” means that either “0” or “1” can be written to the bit by software Initialization.

PADCTL_G7_SOC_GPIO45_0
Offset: 0x0
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_G7_SCR_SCR_SOC_GPIO45_0
Reset: 0x00000414 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = I2S1 **2** = RSVD2 **3** = RSVD3

PADCTL_G7_CFG2TMC_SOC_GPIO45_0

Offset: 0x4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G7_SCR_SCR_SOC_GPIO45_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G7_SOC_GPIO46_0

Offset: 0x8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G7_SCR_SCR_SOC_GPIO46_0
Reset: 0x00000414 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = I2S1 **2** = RSVD2 **3** = RSVD3

PADCTL_G7_CFG2TMC_SOC_GPIO46_0

Offset: 0xc
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_G7_SCR_SCR_SOC_GPIO46_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G7_SOC_GPIO47_0
Offset: 0x10
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G7_SCR_SCR_SOC_GPIO47_0
Reset: 0x00000454 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,01x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = I2S1 **2** = RSVD2 **3** = RSVD3

PADCTL_G7_CFG2TMC_SOC_GPIO47_0

Offset: 0x14
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G7_SCR_SCR_SOC_GPIO47_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G7_SOC_GPIO48_0

Offset: 0x18
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G7_SCR_SCR_SOC_GPIO48_0
Reset: 0x00000414 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE

Bit	Reset	Description
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = I2S1 **2** = RSVD2 **3** = RSVD3

PADCTL_G7_CFG2TMC_SOC_GPIO48_0

Offset: 0x1c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G7_SCR_SCR_SOC_GPIO48_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G7_SOC_GPIO57_0

Offset: 0x20
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G7_SCR_SCR_SOC_GPIO57_0
Reset: 0x00001414 (0bxxxx,xxxx,xxxx,xxxx,xxx1,x1x0,00x1,0100)

Bit	Reset	Description
12	ENABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = I2S8 **2** = RSVD2 **3** = SDMMC1

PADCTL_G7_CFG2TMC_SOC_GPIO57_0

Offset: 0x24
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G7_SCR_SCR_SOC_GPIO57_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G7_SOC_GPIO58_0

Offset: 0x28
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G7_SCR_SCR_SOC_GPIO58_0
Reset: 0x00000414 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = I2S8 **2** = RSVD2 **3** = SDMMC1

PADCTL_G7_CFG2TMC_SOC_GPIO58_0

Offset: 0x2c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G7_SCR_SCR_SOC_GPIO58_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G7_SOC_GPIO59_0

Offset: 0x30
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G7_SCR_SCR_SOC_GPIO59_0
Reset: 0x00000454 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,01x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	AUD	**PM**: **0** = AUD **1** = I2S8 **2** = RSVD2 **3** = RSVD3

PADCTL_G7_CFG2TMC_SOC_GPIO59_0

Offset: 0x34
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G7_SCR_SCR_SOC_GPIO59_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G7_SOC_GPIO60_0

Offset: 0x38
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G7_SCR_SCR_SOC_GPIO60_0
Reset: 0x00000454 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,01x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = I2S8 **2** = NV **3** = IGPU

PADCTL_G7_CFG2TMC_SOC_GPIO60_0

Offset: 0x3c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G7_SCR_SCR_SOC_GPIO60_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G7_SPI5_CS0_0
Offset: 0x40
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G7_SCR_SCR_SPI5_CS0_0
Reset: 0x00000414 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

Bit	Reset	Description
1:0	SPI5	**PM**: **0** = SPI5 **1** = I2S3 **2** = DMIC2 **3** = RSVD3

PADCTL_G7_CFG2TMC_SPI5_CS0_0

Offset: 0x44
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G7_SCR_SCR_SPI5_CS0_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G7_SPI5_MISO_0

Offset: 0x48
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G7_SCR_SCR_SPI5_MISO_0
Reset: 0x00000414 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,00x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SPI5	**PM**: **0** = SPI5 **1** = I2S3 **2** = DSPK0 **3** = RSVD3

PADCTL_G7_CFG2TMC_SPI5_MISO_0

Offset: 0x4c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G7_SCR_SCR_SPI5_MISO_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G7_SPI5_MOSI_0

Offset: 0x50
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G7_SCR_SCR_SPI5_MOSI_0
Reset: 0x00000454 (0bxxxx,xxxx,xxxx,xxxx,xxx0,x1x0,01x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SPI5	**PM**: **0** = SPI5 **1** = I2S3 **2** = DMIC2 **3** = RSVD3

PADCTL_G7_CFG2TMC_SPI5_MOSI_0

Offset: 0x54
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G7_SCR_SCR_SPI5_MOSI_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G7_SPI5_SCK_0

Offset: 0x58
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: PADCTL_G7_SCR_SCR_SPI5_SCK_0
Reset: 0x00001454 (0bxxxx,xxxx,xxxx,xxxx,xxx1,x1x0,01x1,0100)

Bit	Reset	Description
12	ENABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SPI5	**PM**: **0** = SPI5 **1** = I2S3 **2** = DSPK0 **3** = RSVD3

PADCTL_G7_CFG2TMC_SPI5_SCK_0

Offset: 0x5c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G7_SCR_SCR_SPI5_SCK_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G7_EC_FEATURE_0

```
================================
 GLOBAL SLICE
================================
```

Offset: 0x400
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G7_SCR_EC_SCR_0
Reset: 0x00020001 (0b0000,0000,0000,0010,xxxx,xxxx,xx00,0001)

Bit	Reset	Description
31:16	0x2	**NUM_ERR**: Number of errors connected to this collator. This is passed as a build time option to the plugin
5:0	0x1	**NUM_ERR_SLICES**: Number of error slices supported by this error collator, does not include the GlobalSpace and is derived by ceil (NUM_ERR/32). SW shall first read this register to determine the number of slices and read the required number of Error_Status registers .

PADCTL_G7_EC_SWRESET_0

Offset: 0x404
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G7_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SWRST**: 1'b1 : Issue a SW reset to the Error Collator. This will reset all the registers(Except SCR), counters and logic of the Error Collator. SW can use this bit to flush errors logged into the error collator for ex, after Boot, SC7/8 exit. 1'b0 : Do nothing, reset value. This bit is auto-cleared.

PADCTL_G7_EC_MISSIONERR_TYPE_0

Offset: 0x408
Read/Write: RO

Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G7_SCR_EC_SCR_0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0101)

Bit	Reset	Description
5:0	0x5	**CODE**: This register indicates the fault code of the error line based on the value of MISSIONERR_INDEX Register. This can be used by a fault handling agent to triage an error without requiring device-specific code.The possible values of this field are: 6'd0 : None 6'd1 : Parity Error on internal data path 6'd2 : ECC SEC Error on internal data path 6'd3 : ECC DED Error on internal data path 6'd4 : Comparator Error 6'd5 : Register Parity Error 6'd6 : Parity Error from on-chip SRAM/Fifo 6'd7 : ECC SEC Error from on-chip SRAM/Fifo 6'd8 : ECC DED Error from on-chip SRAM/Fifo 6'd9 : Clock Monitor Error 6'd10 : Voltage Error 6'd11 : Temperature Error 6'd12 : CRC Error on internal data path 6'd16 : SW Corrected Error 6'd17 : SW Uncorrected Error 6'd18 : SW Generic Error 6'd20 : ECC SED Error on internal data path 6'd21 : ECC SED Error from on-chip SRAM/Fifo 6'd32 : Other HW Corrected Error 6'd33 : Other HW Uncorrected Error All other values : Reserved for future use.

PADCTL_G7_EC_CURRENT_COUNTER_VALUE_0

Offset: 0x40c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G7_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
8:0	0x0	**VALUE**: Provides the current value of the counter corresponding to the error in MissionErr_Index Register. Default provides the value of error 0 counter. Bit[8] is the ovrerflow bit post which the counter saturates and does not counter further.

PADCTL_G7_EC_MISSIONERR_INDEX_0

Offset: 0x414
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G7_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**IDX**: BINARY Encoded. For error number 32, register should be programmed with value 0x20. Write to this register with Error number will update: - MISSIONERR_TYPE Register with the Error-Code for the Error. - CURRENT_COUNTER_VALUE Register with the error's SEC/DED Counter. - MISSIONERR_USERVALUE with value of the first error_<i>_user signal. SW can use this to trage the error. number shall update the MISSIONERR_TYPE register with the error code and the Current_Counter_Value register with the value of the errors SEC/DED counter. SW can use this register to triage the error.

PADCTL_G7_EC_CORRECTABLE_THRESHOLD_0

Offset: 0x418
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G7_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**COUNT**: Threshold value for all SEC Fault Reporting Units connected to this error collator. SEC Errors are logged once the threshold is reached and the overflow bit is set. 7'b0 : Log SEC error after receiving 1 Error. 7'b1 : Log SEC error after receiving 2 Errors. ... 7'bFF : Log SEC error after receiving 256 Errors.

PADCTL_G7_EC_MISSIONERR_INJECT_UNLOCK_0

Offset: 0x41c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G7_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Writes to ERRSLICE_XXX_MISSIONERR_INJECT registers are disabled until this register is written with a value of 0xE1. This is to prevent an inadvertent safety error injection in the safety plugin due to: 1. A fault on ERRSLICE_XXX_MISSIONERR_INJECT register itself. 2. Erroneous SW. The register shall be written with a value of 0x0 to reestablish the lock after user has completed the error injection testing. 0xE1 : Unlock the MISSIONERR_INJECT Register 0x0 : Lock the MISSIONERR_INJECT Register **0** = LOCK **225** = UNLOCK

PADCTL_G7_EC_ERRSLICE0_MISSIONERR_ENABLE_0

```
==================================
 ERROR SLICE - 0
==================================
```

Offset: 0x430
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G7_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.G7_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.G7_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.G7_PADCTLREG_reg 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.G7_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL_G7_err_collator 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL_G7_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_G7_EC_ERRSLICE0_MISSIONERR_FORCE_0

Offset: 0x434
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G7_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.G7_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.G7_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL_G7_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_G7_EC_ERRSLICE0_MISSIONERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of MissionError_Enable register, to avoid silent dropping of errors.
Offset: 0x438
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G7_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G7_PADCTLREG_reg_scr_wrap was equal to 2'b10. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G7_PADCTLREG_reg_scr_wrap was equal to 2'b01.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G7_PADCTLREG_reg was equal to 2'b10. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G7_PADCTLREG_reg was equal to 2'b01.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G7_err_collator was equal to 2'b10. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G7_err_collator was equal to 2'b01.

PADCTL_G7_EC_ERRSLICE0_MISSIONERR_INJECT_0

Offset: 0x43c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G7_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Assert the inject_error_2 output for Register Parity Error to PADCTL.G7_PADCTLREG_reg_scr_wrap to allow for error injection. 1'b0 -> De-Asseert inject_error_2 output. **0** = DISABLE **1** = ENABLE
1	0x0	**ERR1**: 1'b1 -> Assert the inject_error_1 output for Register Parity Error to PADCTL.G7_PADCTLREG_reg to allow for error injection. 1'b0 -> De-Asseert inject_error_1 output. **0** = DISABLE **1** = ENABLE
0	0x0	**ERR0**: 1'b1 -> Assert the inject_error_0 output for Register Parity Error to PADCTL_G7_err_collator to allow for error injection. 1'b0 -> De-Asseert inject_error_0 output. **0** = DISABLE **1** = ENABLE

PADCTL_G7_EC_ERRSLICE0_LATENTERR_ENABLE_0

Offset: 0x440
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G7_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.G7_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.G7_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.G7_PADCTLREG_reg 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.G7_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL_G7_err_collator 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL_G7_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_G7_EC_ERRSLICE0_LATENTERR_FORCE_0

Offset: 0x444
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G7_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.G7_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.G7_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL_G7_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_G7_EC_ERRSLICE0_LATENTERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of LatentError_Enable register, to avoid silent dropping of errors.
Offset: 0x448
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G7_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G7_PADCTLREG_reg_scr_wrap was equal to 2'b00 or 2'b11. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G7_PADCTLREG_reg_scr_wrap was equal to 2'b01 or 2'b10, but no latent error.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G7_PADCTLREG_reg was equal to 2'b00 or 2'b11. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G7_PADCTLREG_reg was equal to 2'b01 or 2'b10, but no latent error.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G7_err_collator was equal to 2'b00 or 2'b11. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G7_err_collator was equal to 2'b01 or 2'b10, but no latent error.

PADCTL_G7_EC_ERRSLICE0_COUNTER_RELOAD_0

Offset: 0x450

Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G7_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.G7_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
1	0x0	**ERR1**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.G7_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
0	0x0	**ERR0**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL_G7_err_collator 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

8.4.22 G8 PAD Control Registers

NOTE:

The G8 PAD Control Registers are collectively called PSC_PADCTL in the System Address Map.

The binary “x” is turned to “0” for the Reset and PROD hex value. Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as “x” or “Undefined” actually means that the bit could be either “0” or “1” after Reset, whereas a bit PROD value as “x” or “Don’t-care” means that either “0” or “1” can be written to the bit by software Initialization.

PADCTL_G8_SPI6_MISO_0
Offset: 0x0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G8_SCR_SCR_SPI6_MISO_0
Reset: 0x00000014 (0bxxxx,xxxx,xxxx,xxxx,xxx0,xxx0,00x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SPI6	**PM**: **0** = SPI6 **1** = I2C15 **2** = RSVD2 **3** = RSVD3

PADCTL_G8_CFG2TMC_SPI6_MISO_0

Offset: 0x4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G8_SCR_SCR_SPI6_MISO_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G8_SPI6_CS0_0

Offset: 0x8
Read/Write: R/W
Parity Protection: Y

Shadow: N
SCR Protection: PADCTL_G8_SCR_SCR_SPI6_CS0_0
Reset: 0x00000014 (0bxxxx,xxxx,xxxx,xxxx,xxx0,xxx0,00x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SPI6	**PM**: **0** = SPI6 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G8_CFG2TMC_SPI6_CS0_0

Offset: 0xc
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G8_SCR_SCR_SPI6_CS0_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G8_SPI6_SCK_0

Offset: 0x10
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G8_SCR_SCR_SPI6_SCK_0
Reset: 0x00000014 (0bxxxx,xxxx,xxxx,xxxx,xxx0,xxx0,00x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SPI6	**PM**: **0** = SPI6 **1** = I2C15 **2** = RSVD2 **3** = RSVD3

PADCTL_G8_CFG2TMC_SPI6_SCK_0

Offset: 0x14
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G8_SCR_SCR_SPI6_SCK_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G8_SPI6_MOSI_0
Offset: 0x18
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G8_SCR_SCR_SPI6_MOSI_0
Reset: 0x00000014 (0bxxxx,xxxx,xxxx,xxxx,xxx0,xxx0,00x1,0100)

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SPI6	**PM**: **0** = SPI6 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G8_CFG2TMC_SPI6_MOSI_0
Offset: 0x1c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G8_SCR_SCR_SPI6_MOSI_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G8_EC_FEATURE_0

```
=================================
 GLOBAL SLICE
=================================
```

Offset: 0x400
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G8_SCR_EC_SCR_0
Reset: 0x00020001 (0b0000,0000,0000,0010,xxxx,xxxx,xx00,0001)

Bit	Reset	Description
31:16	0x2	**NUM_ERR**: Number of errors connected to this collator. This is passed as a build time option to the plugin
5:0	0x1	**NUM_ERR_SLICES**: Number of error slices supported by this error collator, does not include the GlobalSpace and is derived by ceil (NUM_ERR/32). SW shall first read this register to determine the number of slices and read the required number of Error_Status registers .

PADCTL_G8_EC_SWRESET_0

Offset: 0x404
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G8_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SWRST**: 1'b1 : Issue a SW reset to the Error Collator. This will reset all the registers(Except SCR), counters and logic of the Error Collator. SW can use this bit to flush errors logged into the error collator for ex, after Boot, SC7/8 exit. 1'b0 : Do nothing, reset value. This bit is auto-cleared.

PADCTL_G8_EC_MISSIONERR_TYPE_0

Offset: 0x408
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G8_SCR_EC_SCR_0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0101)

Bit	Reset	Description
5:0	0x5	**CODE**: This register indicates the fault code of the error line based on the value of MISSIONERR_INDEX Register. This can be used by a fault handling agent to triage an error without requiring device-specific code.The possible values of this field are: 6'd0 : None 6'd1 : Parity Error on internal data path 6'd2 : ECC SEC Error on internal data path 6'd3 : ECC DED Error on internal data path 6'd4 : Comparator Error 6'd5 : Register Parity Error 6'd6 : Parity Error from on-chip SRAM/Fifo 6'd7 : ECC SEC Error from on-chip SRAM/Fifo 6'd8 : ECC DED Error from on-chip SRAM/Fifo 6'd9 : Clock Monitor Error 6'd10 : Voltage Error 6'd11 : Temperature Error 6'd12 : CRC Error on internal data path 6'd16 : SW Corrected Error 6'd17 : SW Uncorrected Error 6'd18 : SW Generic Error 6'd20 : ECC SED Error on internal data path 6'd21 : ECC SED Error from on-chip SRAM/Fifo 6'd32 : Other HW Corrected Error 6'd33 : Other HW Uncorrected Error All other values : Reserved for future use.

PADCTL_G8_EC_CURRENT_COUNTER_VALUE_0

Offset: 0x40c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G8_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
8:0	0x0	**VALUE**: Provides the current value of the counter corresponding to the error in MissionErr_Index Register. Default provides the value of error 0 counter. Bit[8] is the ovrerflow bit post which the counter saturates and does not counter further.

PADCTL_G8_EC_MISSIONERR_INDEX_0

Offset: 0x414
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G8_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**IDX**: BINARY Encoded. For error number 32, register should be programmed with value 0x20. Write to this register with Error number will update: - MISSIONERR_TYPE Register with the Error-Code for the Error. - CURRENT_COUNTER_VALUE Register with the error's SEC/DED Counter. - MISSIONERR_USERVALUE with value of the first error_<i>_user signal. SW can use this to trage the error. number shall update the MISSIONERR_TYPE register with the error code and the Current_Counter_Value register with the value of the errors SEC/DED counter. SW can use this register to triage the error.

PADCTL_G8_EC_CORRECTABLE_THRESHOLD_0

Offset: 0x418
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G8_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**COUNT**: Threshold value for all SEC Fault Reporting Units connected to this error collator. SEC Errors are logged once the threshold is reached and the overflow bit is set. 7'b0 : Log SEC error after receiving 1 Error. 7'b1 : Log SEC error after receiving 2 Errors. ... 7'bFF : Log SEC error after receiving 256 Errors.

PADCTL_G8_EC_MISSIONERR_INJECT_UNLOCK_0

Offset: 0x41c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G8_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Writes to ERRSLICE_XXX_MISSIONERR_INJECT registers are disabled until this register is written with a value of 0xE1. This is to prevent an inadvertent safety error injection in the safety plugin due to: 1. A fault on ERRSLICE_XXX_MISSIONERR_INJECT register itself. 2. Erroneous SW. The register shall be written with a value of 0x0 to reestablish the lock after user has completed the error injection testing. 0xE1 : Unlock the MISSIONERR_INJECT Register 0x0 : Lock the MISSIONERR_INJECT Register **0** = LOCK **225** = UNLOCK

PADCTL_G8_EC_ERRSLICE0_MISSIONERR_ENABLE_0

```
==================================
 ERROR SLICE - 0
==================================
```

Offset: 0x430
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G8_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.G8_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.G8_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
1	0x1	**ERR1**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.G8_PADCTLREG_reg 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.G8_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL_G8_err_collator 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL_G8_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_G8_EC_ERRSLICE0_MISSIONERR_FORCE_0

Offset: 0x434
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G8_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.G8_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.G8_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL_G8_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_G8_EC_ERRSLICE0_MISSIONERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of MissionError_Enable register, to avoid silent dropping of errors.
Offset: 0x438
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G8_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G8_PADCTLREG_reg_scr_wrap was equal to 2'b10. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G8_PADCTLREG_reg_scr_wrap was equal to 2'b01.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G8_PADCTLREG_reg was equal to 2'b10. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G8_PADCTLREG_reg was equal to 2'b01.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G8_err_collator was equal to 2'b10. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G8_err_collator was equal to 2'b01.

PADCTL_G8_EC_ERRSLICE0_MISSIONERR_INJECT_0

Offset: 0x43c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G8_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Assert the inject_error_2 output for Register Parity Error to PADCTL.G8_PADCTLREG_reg_scr_wrap to allow for error injection. 1'b0 -> De-Asseert inject_error_2 output. **0** = DISABLE **1** = ENABLE
1	0x0	**ERR1**: 1'b1 -> Assert the inject_error_1 output for Register Parity Error to PADCTL.G8_PADCTLREG_reg to allow for error injection. 1'b0 -> De-Asseert inject_error_1 output. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
0	0x0	**ERR0**: 1'b1 -> Assert the inject_error_0 output for Register Parity Error to PADCTL_G8_err_collator to allow for error injection. 1'b0 -> De-Asseert inject_error_0 output. **0** = DISABLE **1** = ENABLE

PADCTL_G8_EC_ERRSLICE0_LATENTERR_ENABLE_0

Offset: 0x440
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G8_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.G8_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.G8_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.G8_PADCTLREG_reg 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.G8_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL_G8_err_collator 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL_G8_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_G8_EC_ERRSLICE0_LATENTERR_FORCE_0

Offset: 0x444
Read/Write: WO
Parity Protection: N
Shadow: N

SCR Protection: PADCTL_G8_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.G8_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.G8_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL_G8_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_G8_EC_ERRSLICE0_LATENTERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of LatentError_Enable register, to avoid silent dropping of errors.
Offset: 0x448
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G8_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G8_PADCTLREG_reg_scr_wrap was equal to 2'b00 or 2'b11. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G8_PADCTLREG_reg_scr_wrap was equal to 2'b01 or 2'b10, but no latent error.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G8_PADCTLREG_reg was equal to 2'b00 or 2'b11. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G8_PADCTLREG_reg was equal to 2'b01 or 2'b10, but no latent error.

Bit	Reset	Description
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G8_err_collator was equal to 2'b00 or 2'b11. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G8_err_collator was equal to 2'b01 or 2'b10, but no latent error.

PADCTL_G8_EC_ERRSLICE0_COUNTER_RELOAD_0
Offset: 0x450
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G8_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.G8_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
1	0x0	**ERR1**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.G8_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
0	0x0	**ERR0**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL_G8_err_collator 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

8.4.23 G9 PAD Control Registers

NOTE:

The G9 PAD Control Registers are collectively called FSI_PADCTL_A0 in the System Address Map.

The binary “x” is turned to “0” for the Reset and PROD hex value. Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as “x” or “Undefined” actually means that the bit could be either “0” or “1” after Reset, whereas a bit

PROD value as “x” or “Don’t-care” means that either “0” or “1” can be written to the bit by software Initialization.

PADCTL_G9_CAN2_DIN_0
Offset: 0x0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G9_SCR_SCR_CAN2_DIN_0
Reset: 0x0008c450 (0b0000,0000,0000,1000,1100,x10x,x101,0000)

Bit	Reset	Description
31:28	0x0	**SPARE_VDD**
27:24	0x0	**SPARE_VAUXC**
23:20	0x0	**RFU_IN**
19	ENABLE	**E_33V**: **0** = DISABLE **1** = ENABLE
18	DISABLE	**SEL_DPD**: **0** = DISABLE **1** = ENABLE
17	DISABLE	**E_DPD**: **0** = DISABLE **1** = ENABLE
16:15	0x1	**SCHMITT**
14:13	DEFAULT_DRIVE_1X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
9	DISABLE	**E_PBIAS_BUF**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	CAN2	**PM**: **0** = CAN2 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G9_CFG2TMC_CAN2_DIN_0

Offset: 0x4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G9_SCR_SCR_CAN2_DIN_0
Reset: 0x00000000 (0b0000,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:30	0x0	**CFG_CAL_DRVUP_DEF**
29:28	0x0	**CFG_CAL_DRVDN_DEF**

PADCTL_G9_CAN2_DOUT_0

Offset: 0x8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G9_SCR_SCR_CAN2_DOUT_0
Reset: 0x0008c410 (0b0000,0000,0000,1000,1100,x10x,x001,0000)

Bit	Reset	Description
31:28	0x0	**SPARE_VDD**
27:24	0x0	**SPARE_VAUXC**
23:20	0x0	**RFU_IN**
19	ENABLE	**E_33V**: **0** = DISABLE **1** = ENABLE
18	DISABLE	**SEL_DPD**: **0** = DISABLE **1** = ENABLE
17	DISABLE	**E_DPD**: **0** = DISABLE **1** = ENABLE
16:15	0x1	**SCHMITT**
14:13	DEFAULT_DRIVE_1X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
9	DISABLE	**E_PBIAS_BUF**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE

Bit	Reset	Description
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	CAN2	**PM**: **0** = CAN2 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G9_CFG2TMC_CAN2_DOUT_0

Offset: 0xc
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G9_SCR_SCR_CAN2_DOUT_0
Reset: 0x00000000 (0b0000,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:30	0x0	**CFG_CAL_DRVUP_DEF**
29:28	0x0	**CFG_CAL_DRVDN_DEF**

PADCTL_G9_CAN2_STB_0

Offset: 0x10
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G9_SCR_SCR_CAN2_STB_0
Reset: 0x0008c414 (0b0000,0000,0000,1000,1100,x10x,x001,0100)

Bit	Reset	Description
31:28	0x0	**SPARE_VDD**
27:24	0x0	**SPARE_VAUXC**
23:20	0x0	**RFU_IN**
19	ENABLE	**E_33V**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
18	DISABLE	**SEL_DPD**: **0** = DISABLE **1** = ENABLE
17	DISABLE	**E_DPD**: **0** = DISABLE **1** = ENABLE
16:15	0x1	**SCHMITT**
14:13	DEFAULT_DRIVE_1X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
9	DISABLE	**E_PBIAS_BUF**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G9_CFG2TMC_CAN2_STB_0

Offset: 0x14
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G9_SCR_SCR_CAN2_STB_0
Reset: 0x00000000 (0b0000,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:30	0x0	**CFG_CAL_DRVUP_DEF**
29:28	0x0	**CFG_CAL_DRVDN_DEF**

PADCTL_G9_CAN2_EN_0

Offset: 0x18
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G9_SCR_SCR_CAN2_EN_0
Reset: 0x0008c414 (0b0000,0000,0000,1000,1100,x10x,x001,0100)

Bit	Reset	Description
31:28	0x0	**SPARE_VDD**
27:24	0x0	**SPARE_VAUXC**
23:20	0x0	**RFU_IN**
19	ENABLE	**E_33V**: **0** = DISABLE **1** = ENABLE
18	DISABLE	**SEL_DPD**: **0** = DISABLE **1** = ENABLE
17	DISABLE	**E_DPD**: **0** = DISABLE **1** = ENABLE
16:15	0x1	**SCHMITT**

Bit	Reset	Description
14:13	DEFAULT_DRIVE_1X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
9	DISABLE	**E_PBIAS_BUF**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G9_CFG2TMC_CAN2_EN_0

Offset: 0x1c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G9_SCR_SCR_CAN2_EN_0
Reset: 0x00000000 (0b0000,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:30	0x0	**CFG_CAL_DRVUP_DEF**
29:28	0x0	**CFG_CAL_DRVDN_DEF**

PADCTL_G9_CAN2_ERR_0

Offset: 0x20
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G9_SCR_SCR_CAN2_ERR_0
Reset: 0x0008c414 (0b0000,0000,0000,1000,1100,x10x,x001,0100)

Bit	Reset	Description
31:28	0x0	**SPARE_VDD**
27:24	0x0	**SPARE_VAUXC**
23:20	0x0	**RFU_IN**
19	ENABLE	**E_33V**: **0** = DISABLE **1** = ENABLE
18	DISABLE	**SEL_DPD**: **0** = DISABLE **1** = ENABLE
17	DISABLE	**E_DPD**: **0** = DISABLE **1** = ENABLE
16:15	0x1	**SCHMITT**
14:13	DEFAULT_DRIVE_1X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO

Bit	Reset	Description
9	DISABLE	**E_PBIAS_BUF**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G9_CFG2TMC_CAN2_ERR_0

Offset: 0x24
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G9_SCR_SCR_CAN2_ERR_0
Reset: 0x00000000 (0b0000,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:30	0x0	**CFG_CAL_DRVUP_DEF**
29:28	0x0	**CFG_CAL_DRVDN_DEF**

PADCTL_G9_CAN3_DIN_0

Offset: 0x28
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G9_SCR_SCR_CAN3_DIN_0
Reset: 0x0008c450 (0b0000,0000,0000,1000,1100,x10x,x101,0000)

Bit	Reset	Description
31:28	0x0	**SPARE_VDD**
27:24	0x0	**SPARE_VAUXC**
23:20	0x0	**RFU_IN**
19	ENABLE	**E_33V**: **0** = DISABLE **1** = ENABLE
18	DISABLE	**SEL_DPD**: **0** = DISABLE **1** = ENABLE
17	DISABLE	**E_DPD**: **0** = DISABLE **1** = ENABLE
16:15	0x1	**SCHMITT**
14:13	DEFAULT_DRIVE_1X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
9	DISABLE	**E_PBIAS_BUF**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE

Bit	Reset	Description
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	CAN3	**PM**: **0** = CAN3 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G9_CFG2TMC_CAN3_DIN_0

Offset: 0x2c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G9_SCR_SCR_CAN3_DIN_0
Reset: 0x00000000 (0b0000,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:30	0x0	**CFG_CAL_DRVUP_DEF**
29:28	0x0	**CFG_CAL_DRVDN_DEF**

PADCTL_G9_CAN3_DOUT_0

Offset: 0x30
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G9_SCR_SCR_CAN3_DOUT_0
Reset: 0x0008c410 (0b0000,0000,0000,1000,1100,x10x,x001,0000)

Bit	Reset	Description
31:28	0x0	**SPARE_VDD**
27:24	0x0	**SPARE_VAUXC**
23:20	0x0	**RFU_IN**
19	ENABLE	**E_33V**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
18	DISABLE	**SEL_DPD**: **0** = DISABLE **1** = ENABLE
17	DISABLE	**E_DPD**: **0** = DISABLE **1** = ENABLE
16:15	0x1	**SCHMITT**
14:13	DEFAULT_DRIVE_1X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
9	DISABLE	**E_PBIAS_BUF**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	CAN3	**PM**: **0** = CAN3 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G9_CFG2TMC_CAN3_DOUT_0

Offset: 0x34
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G9_SCR_SCR_CAN3_DOUT_0
Reset: 0x00000000 (0b0000,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:30	0x0	**CFG_CAL_DRVUP_DEF**
29:28	0x0	**CFG_CAL_DRVDN_DEF**

PADCTL_G9_CAN3_STB_0

Offset: 0x38
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G9_SCR_SCR_CAN3_STB_0
Reset: 0x0008c414 (0b0000,0000,0000,1000,1100,x10x,x001,0100)

Bit	Reset	Description
31:28	0x0	**SPARE_VDD**
27:24	0x0	**SPARE_VAUXC**
23:20	0x0	**RFU_IN**
19	ENABLE	**E_33V**: **0** = DISABLE **1** = ENABLE
18	DISABLE	**SEL_DPD**: **0** = DISABLE **1** = ENABLE
17	DISABLE	**E_DPD**: **0** = DISABLE **1** = ENABLE
16:15	0x1	**SCHMITT**

Bit	Reset	Description
14:13	DEFAULT_DRIVE_1X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
9	DISABLE	**E_PBIAS_BUF**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G9_CFG2TMC_CAN3_STB_0

Offset: 0x3c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G9_SCR_SCR_CAN3_STB_0
Reset: 0x00000000 (0b0000,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:30	0x0	**CFG_CAL_DRVUP_DEF**
29:28	0x0	**CFG_CAL_DRVDN_DEF**

PADCTL_G9_CAN3_EN_0

Offset: 0x40
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G9_SCR_SCR_CAN3_EN_0
Reset: 0x0008c414 (0b0000,0000,0000,1000,1100,x10x,x001,0100)

Bit	Reset	Description
31:28	0x0	**SPARE_VDD**
27:24	0x0	**SPARE_VAUXC**
23:20	0x0	**RFU_IN**
19	ENABLE	**E_33V**: **0** = DISABLE **1** = ENABLE
18	DISABLE	**SEL_DPD**: **0** = DISABLE **1** = ENABLE
17	DISABLE	**E_DPD**: **0** = DISABLE **1** = ENABLE
16:15	0x1	**SCHMITT**
14:13	DEFAULT_DRIVE_1X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO

Bit	Reset	Description
9	DISABLE	**E_PBIAS_BUF**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G9_CFG2TMC_CAN3_EN_0

Offset: 0x44
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G9_SCR_SCR_CAN3_EN_0
Reset: 0x00000000 (0b0000,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:30	0x0	**CFG_CAL_DRVUP_DEF**
29:28	0x0	**CFG_CAL_DRVDN_DEF**

PADCTL_G9_CAN3_ERR_0

Offset: 0x48
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G9_SCR_SCR_CAN3_ERR_0
Reset: 0x0008c414 (0b0000,0000,0000,1000,1100,x10x,x001,0100)

Bit	Reset	Description
31:28	0x0	**SPARE_VDD**
27:24	0x0	**SPARE_VAUXC**
23:20	0x0	**RFU_IN**
19	ENABLE	**E_33V**: **0** = DISABLE **1** = ENABLE
18	DISABLE	**SEL_DPD**: **0** = DISABLE **1** = ENABLE
17	DISABLE	**E_DPD**: **0** = DISABLE **1** = ENABLE
16:15	0x1	**SCHMITT**
14:13	DEFAULT_DRIVE_1X	**DRV_TYPE**: DRV_TYPE[0]: 0 = 1X driver; 1 = 2X driver; DRV_TYPE[1]: 0 = Driving codes from BDSDMEMCOMP; 1 = Default driving codes; **0** = COMP_DRIVE_1X **1** = COMP_DRIVE_2X **2** = DEFAULT_DRIVE_1X **3** = DEFAULT_DRIVE_2X
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
9	DISABLE	**E_PBIAS_BUF**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE

Bit	Reset	Description
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G9_CFG2TMC_CAN3_ERR_0

Offset: 0x4c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G9_SCR_SCR_CAN3_ERR_0
Reset: 0x00000000 (0b0000,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:30	0x0	**CFG_CAL_DRVUP_DEF**
29:28	0x0	**CFG_CAL_DRVDN_DEF**

PADCTL_G9_EC_FEATURE_0

```
================================
 GLOBAL SLICE
================================
```

Offset: 0x400
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G9_SCR_EC_SCR_0
Reset: 0x00020001 (0b0000,0000,0000,0010,xxxx,xxxx,xx00,0001)

Bit	Reset	Description
31:16	0x2	**NUM_ERR**: Number of errors connected to this collator. This is passed as a build time option to the plugin

Bit	Reset	Description
5:0	0x1	**NUM_ERR_SLICES**: Number of error slices supported by this error collator, does not include the GlobalSpace and is derived by ceil (NUM_ERR/32). SW shall first read this register to determine the number of slices and read the required number of Error_Status registers .

PADCTL_G9_EC_SWRESET_0

Offset: 0x404
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G9_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SWRST**: 1'b1 : Issue a SW reset to the Error Collator. This will reset all the registers(Except SCR), counters and logic of the Error Collator. SW can use this bit to flush errors logged into the error collator for ex, after Boot, SC7/8 exit. 1'b0 : Do nothing, reset value. This bit is auto-cleared.

PADCTL_G9_EC_MISSIONERR_TYPE_0

Offset: 0x408
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G9_SCR_EC_SCR_0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0101)

Bit	Reset	Description
5:0	0x5	**CODE**: This register indicates the fault code of the error line based on the value of MISSIONERR_INDEX Register. This can be used by a fault handling agent to triage an error without requiring device-specific code.The possible values of this field are: 6'd0 : None 6'd1 : Parity Error on internal data path 6'd2 : ECC SEC Error on internal data path 6'd3 : ECC DED Error on internal data path 6'd4 : Comparator Error 6'd5 : Register Parity Error 6'd6 : Parity Error from on-chip SRAM/Fifo 6'd7 : ECC SEC Error from on-chip SRAM/Fifo 6'd8 : ECC DED Error from on-chip SRAM/Fifo 6'd9 : Clock Monitor Error 6'd10 : Voltage Error 6'd11 : Temperature Error 6'd12 : CRC Error on internal data path 6'd16 : SW Corrected Error 6'd17 : SW Uncorrected Error 6'd18 : SW Generic Error 6'd20 : ECC SED Error on internal data path 6'd21 : ECC SED Error from on-chip SRAM/Fifo 6'd32 : Other HW Corrected Error 6'd33 : Other HW Uncorrected Error All other values : Reserved for future use.

PADCTL_G9_EC_CURRENT_COUNTER_VALUE_0

Offset: 0x40c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G9_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
8:0	0x0	**VALUE**: Provides the current value of the counter corresponding to the error in MissionErr_Index Register. Default provides the value of error 0 counter. Bit[8] is the ovrerflow bit post which the counter saturates and does not counter further.

PADCTL_G9_EC_MISSIONERR_INDEX_0

Offset: 0x414
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_G9_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**IDX**: BINARY Encoded. For error number 32, register should be programmed with value 0x20. Write to this register with Error number will update: - MISSIONERR_TYPE Register with the Error-Code for the Error. - CURRENT_COUNTER_VALUE Register with the error's SEC/DED Counter. - MISSIONERR_USERVALUE with value of the first error_<i>_user signal. SW can use this to trage the error. number shall update the MISSIONERR_TYPE register with the error code and the Current_Counter_Value register with the value of the errors SEC/DED counter. SW can use this register to triage the error.

PADCTL_G9_EC_CORRECTABLE_THRESHOLD_0

Offset: 0x418
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G9_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**COUNT**: Threshold value for all SEC Fault Reporting Units connected to this error collator. SEC Errors are logged once the threshold is reached and the overflow bit is set. 7'b0 : Log SEC error after receiving 1 Error. 7'b1 : Log SEC error after receiving 2 Errors. ... 7'bFF : Log SEC error after receiving 256 Errors.

PADCTL_G9_EC_MISSIONERR_INJECT_UNLOCK_0

Offset: 0x41c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G9_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Writes to ERRSLICE_XXX_MISSIONERR_INJECT registers are disabled until this register is written with a value of 0xE1. This is to prevent an inadvertent safety error injection in the safety plugin due to: 1. A fault on ERRSLICE_XXX_MISSIONERR_INJECT register itself. 2. Erroneous SW. The register shall be written with a value of 0x0 to reestablish the lock after user has completed the error injection testing. 0xE1 : Unlock the MISSIONERR_INJECT Register 0x0 : Lock the MISSIONERR_INJECT Register **0** = LOCK **225** = UNLOCK

PADCTL_G9_EC_ERRSLICE0_MISSIONERR_ENABLE_0

```
==================================
 ERROR SLICE - 0
==================================
```

Offset: 0x430
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G9_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.G9_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.G9_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.G9_PADCTLREG_reg 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.G9_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL_G9_err_collator 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL_G9_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_G9_EC_ERRSLICE0_MISSIONERR_FORCE_0

Offset: 0x434
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G9_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.G9_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.G9_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL_G9_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_G9_EC_ERRSLICE0_MISSIONERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of MissionError_Enable register, to avoid silent dropping of errors.
Offset: 0x438
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G9_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G9_PADCTLREG_reg_scr_wrap was equal to 2'b10. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G9_PADCTLREG_reg_scr_wrap was equal to 2'b01.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G9_PADCTLREG_reg was equal to 2'b10. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G9_PADCTLREG_reg was equal to 2'b01.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G9_err_collator was equal to 2'b10. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G9_err_collator was equal to 2'b01.

PADCTL_G9_EC_ERRSLICE0_MISSIONERR_INJECT_0

Offset: 0x43c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G9_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Assert the inject_error_2 output for Register Parity Error to PADCTL.G9_PADCTLREG_reg_scr_wrap to allow for error injection. 1'b0 -> De-Asseert inject_error_2 output. **0** = DISABLE **1** = ENABLE
1	0x0	**ERR1**: 1'b1 -> Assert the inject_error_1 output for Register Parity Error to PADCTL.G9_PADCTLREG_reg to allow for error injection. 1'b0 -> De-Asseert inject_error_1 output. **0** = DISABLE **1** = ENABLE
0	0x0	**ERR0**: 1'b1 -> Assert the inject_error_0 output for Register Parity Error to PADCTL_G9_err_collator to allow for error injection. 1'b0 -> De-Asseert inject_error_0 output. **0** = DISABLE **1** = ENABLE

PADCTL_G9_EC_ERRSLICE0_LATENTERR_ENABLE_0

Offset: 0x440
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G9_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.G9_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.G9_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.G9_PADCTLREG_reg 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.G9_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL_G9_err_collator 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL_G9_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_G9_EC_ERRSLICE0_LATENTERR_FORCE_0

Offset: 0x444
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G9_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.G9_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.G9_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL_G9_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_G9_EC_ERRSLICE0_LATENTERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of LatentError_Enable register, to avoid silent dropping of errors.
Offset: 0x448
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G9_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G9_PADCTLREG_reg_scr_wrap was equal to 2'b00 or 2'b11. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G9_PADCTLREG_reg_scr_wrap was equal to 2'b01 or 2'b10, but no latent error.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G9_PADCTLREG_reg was equal to 2'b00 or 2'b11. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G9_PADCTLREG_reg was equal to 2'b01 or 2'b10, but no latent error.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G9_err_collator was equal to 2'b00 or 2'b11. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G9_err_collator was equal to 2'b01 or 2'b10, but no latent error.

PADCTL_G9_EC_ERRSLICE0_COUNTER_RELOAD_0

Offset: 0x450

Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G9_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.G9_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
1	0x0	**ERR1**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.G9_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
0	0x0	**ERR0**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL_G9_err_collator 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

8.4.24 G10 PAD Control Registers

NOTE:

The G10 PAD Control Registers are collectively called FSI_PADCTL_A1 in the System Address Map.

The binary “x” is turned to “0” for the Reset and PROD hex value. Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as “x” or “Undefined” actually means that the bit could be either “0” or “1” after Reset, whereas a bit PROD value as “x” or “Don’t-care” means that either “0” or “1” can be written to the bit by software Initialization.

PADCTL_G10_SOC_ERROR_0
Offset: 0x0
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_G10_SCR_SCR_SOC_ERROR_0
Reset: 0x00000430 (0bxxxx,xxxx,xxxx,x00x,xxx0,x1x0,0011,0000)

Bit	Reset	Description
18	DISABLE	**SEL_DPD**: **0** = DISABLE **1** = ENABLE
17	DISABLE	**E_DPD**: **0** = DISABLE **1** = ENABLE
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	NONE	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SOC	**PM**: **0** = SOC **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G10_CFG2TMC_SOC_ERROR_0

Offset: 0x4
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_G10_SCR_SCR_SOC_ERROR_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G10_UART7_TX_0
Offset: 0x8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G10_SCR_SCR_UART7_TX_0
Reset: 0x00000434 (0bxxxx,xxxx,xxxx,x00x,xxx0,x1x0,0011,0100)

Bit	Reset	Description
18	DISABLE	**SEL_DPD**: **0** = DISABLE **1** = ENABLE
17	DISABLE	**E_DPD**: **0** = DISABLE **1** = ENABLE
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE

Bit	Reset	Description
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	UARTG	**PM**: **0** = UARTG **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G10_CFG2TMC_UART7_TX_0

Offset: 0xc
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G10_SCR_SCR_UART7_TX_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G10_UART7_RX_0

Offset: 0x10
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G10_SCR_SCR_UART7_RX_0
Reset: 0x00000474 (0bxxxx,xxxx,xxxx,x00x,xxx0,x1x0,0111,0100)

Bit	Reset	Description
18	DISABLE	**SEL_DPD**: **0** = DISABLE **1** = ENABLE
17	DISABLE	**E_DPD**: **0** = DISABLE **1** = ENABLE
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	ENABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	UARTG	**PM**: **0** = UARTG **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G10_CFG2TMC_UART7_RX_0

Offset: 0x14
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G10_SCR_SCR_UART7_RX_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G10_SPI7_SCK_0

Offset: 0x18
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G10_SCR_SCR_SPI7_SCK_0
Reset: 0x00001434 (0bxxxx,xxxx,xxxx,x00x,xxx1,x1x0,0011,0100)

Bit	Reset	Description
18	DISABLE	**SEL_DPD**: **0** = DISABLE **1** = ENABLE
17	DISABLE	**E_DPD**: **0** = DISABLE **1** = ENABLE
12	ENABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SPI7	**PM**: **0** = SPI7 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G10_CFG2TMC_SPI7_SCK_0

Offset: 0x1c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G10_SCR_SCR_SPI7_SCK_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G10_SPI7_MISO_0
Offset: 0x20
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G10_SCR_SCR_SPI7_MISO_0
Reset: 0x00000434 (0bxxxx,xxxx,xxxx,x00x,xxx0,x1x0,0011,0100)

Bit	Reset	Description
18	DISABLE	**SEL_DPD**: **0** = DISABLE **1** = ENABLE
17	DISABLE	**E_DPD**: **0** = DISABLE **1** = ENABLE
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SPI7	**PM**: **0** = SPI7 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G10_CFG2TMC_SPI7_MISO_0

Offset: 0x24
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G10_SCR_SCR_SPI7_MISO_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G10_SPI7_MOSI_0

Offset: 0x28
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G10_SCR_SCR_SPI7_MOSI_0
Reset: 0x00000434 (0bxxxx,xxxx,xxxx,x00x,xxx0,x1x0,0011,0100)

Bit	Reset	Description
18	DISABLE	**SEL_DPD**: **0** = DISABLE **1** = ENABLE
17	DISABLE	**E_DPD**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SPI7	**PM**: **0** = SPI7 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G10_CFG2TMC_SPI7_MOSI_0

Offset: 0x2c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G10_SCR_SCR_SPI7_MOSI_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G10_SPI7_CS0_0
Offset: 0x30
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G10_SCR_SCR_SPI7_CS0_0
Reset: 0x00000434 (0bxxxx,xxxx,xxxx,x00x,xxx0,x1x0,0011,0100)

Bit	Reset	Description
18	DISABLE	**SEL_DPD**: **0** = DISABLE **1** = ENABLE
17	DISABLE	**E_DPD**: **0** = DISABLE **1** = ENABLE
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	SPI7	**PM**: **0** = SPI7 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G10_CFG2TMC_SPI7_CS0_0

Offset: 0x34
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G10_SCR_SCR_SPI7_CS0_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G10_SOC_GPIO51_0

Offset: 0x38
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G10_SCR_SCR_SOC_GPIO51_0
Reset: 0x00000434 (0bxxxx,xxxx,xxxx,x00x,xxx0,x1x0,0011,0100)

Bit	Reset	Description
18	DISABLE	**SEL_DPD**: **0** = DISABLE **1** = ENABLE
17	DISABLE	**E_DPD**: **0** = DISABLE **1** = ENABLE
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G10_CFG2TMC_SOC_GPIO51_0

Offset: 0x3c
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G10_SCR_SCR_SOC_GPIO51_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G10_SOC_GPIO52_0

Offset: 0x40
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G10_SCR_SCR_SOC_GPIO52_0
Reset: 0x00000434 (0bxxxx,xxxx,xxxx,x00x,xxx0,x1x0,0011,0100)

Bit	Reset	Description
18	DISABLE	**SEL_DPD**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
17	DISABLE	**E_DPD**: **0** = DISABLE **1** = ENABLE
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G10_CFG2TMC_SOC_GPIO52_0

Offset: 0x44
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G10_SCR_SCR_SOC_GPIO52_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**

Bit	Reset	Description
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G10_EC_FEATURE_0

```
=================================
 GLOBAL SLICE
=================================
```

Offset: 0x400
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G10_SCR_EC_SCR_0
Reset: 0x00020001 (0b0000,0000,0000,0010,xxxx,xxxx,xx00,0001)

Bit	Reset	Description
31:16	0x2	**NUM_ERR**: Number of errors connected to this collator. This is passed as a build time option to the plugin
5:0	0x1	**NUM_ERR_SLICES**: Number of error slices supported by this error collator, does not include the GlobalSpace and is derived by ceil (NUM_ERR/32). SW shall first read this register to determine the number of slices and read the required number of Error_Status registers .

PADCTL_G10_EC_SWRESET_0

Offset: 0x404
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G10_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SWRST**: 1'b1 : Issue a SW reset to the Error Collator. This will reset all the registers(Except SCR), counters and logic of the Error Collator. SW can use this bit to flush errors logged into the error collator for ex, after Boot, SC7/8 exit. 1'b0 : Do nothing, reset value. This bit is auto-cleared.

PADCTL_G10_EC_MISSIONERR_TYPE_0

Offset: 0x408
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G10_SCR_EC_SCR_0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0101)

Bit	Reset	Description
5:0	0x5	**CODE**: This register indicates the fault code of the error line based on the value of MISSIONERR_INDEX Register. This can be used by a fault handling agent to triage an error without requiring device-specific code.The possible values of this field are: 6'd0 : None 6'd1 : Parity Error on internal data path 6'd2 : ECC SEC Error on internal data path 6'd3 : ECC DED Error on internal data path 6'd4 : Comparator Error 6'd5 : Register Parity Error 6'd6 : Parity Error from on-chip SRAM/Fifo 6'd7 : ECC SEC Error from on-chip SRAM/Fifo 6'd8 : ECC DED Error from on-chip SRAM/Fifo 6'd9 : Clock Monitor Error 6'd10 : Voltage Error 6'd11 : Temperature Error 6'd12 : CRC Error on internal data path 6'd16 : SW Corrected Error 6'd17 : SW Uncorrected Error 6'd18 : SW Generic Error 6'd20 : ECC SED Error on internal data path 6'd21 : ECC SED Error from on-chip SRAM/Fifo 6'd32 : Other HW Corrected Error 6'd33 : Other HW Uncorrected Error All other values : Reserved for future use.

PADCTL_G10_EC_CURRENT_COUNTER_VALUE_0

Offset: 0x40c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G10_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
8:0	0x0	**VALUE**: Provides the current value of the counter corresponding to the error in MissionErr_Index Register. Default provides the value of error 0 counter. Bit[8] is the ovrerflow bit post which the counter saturates and does not counter further.

PADCTL_G10_EC_MISSIONERR_INDEX_0

Offset: 0x414
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G10_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**IDX**: BINARY Encoded. For error number 32, register should be programmed with value 0x20. Write to this register with Error number will update: - MISSIONERR_TYPE Register with the Error-Code for the Error. - CURRENT_COUNTER_VALUE Register with the error's SEC/DED Counter. - MISSIONERR_USERVALUE with value of the first error_<i>_user signal. SW can use this to trage the error. number shall update the MISSIONERR_TYPE register with the error code and the Current_Counter_Value register with the value of the errors SEC/DED counter. SW can use this register to triage the error.

PADCTL_G10_EC_CORRECTABLE_THRESHOLD_0

Offset: 0x418
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G10_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**COUNT**: Threshold value for all SEC Fault Reporting Units connected to this error collator. SEC Errors are logged once the threshold is reached and the overflow bit is set. 7'b0 : Log SEC error after receiving 1 Error. 7'b1 : Log SEC error after receiving 2 Errors. ... 7'bFF : Log SEC error after receiving 256 Errors.

PADCTL_G10_EC_MISSIONERR_INJECT_UNLOCK_0

Offset: 0x41c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G10_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Writes to ERRSLICE_XXX_MISSIONERR_INJECT registers are disabled until this register is written with a value of 0xE1. This is to prevent an inadvertent safety error injection in the safety plugin due to: 1. A fault on ERRSLICE_XXX_MISSIONERR_INJECT register itself. 2. Erroneous SW. The register shall be written with a value of 0x0 to reestablish the lock after user has completed the error injection testing. 0xE1 : Unlock the MISSIONERR_INJECT Register 0x0 : Lock the MISSIONERR_INJECT Register **0** = LOCK **225** = UNLOCK

PADCTL_G10_EC_ERRSLICE0_MISSIONERR_ENABLE_0

```
==================================
 ERROR SLICE - 0
==================================
```

Offset: 0x430
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G10_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.G10_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.G10_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
1	0x1	**ERR1**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.G10_PADCTLREG_reg 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.G10_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL_G10_err_collator 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL_G10_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_G10_EC_ERRSLICE0_MISSIONERR_FORCE_0

Offset: 0x434
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G10_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.G10_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.G10_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL_G10_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_G10_EC_ERRSLICE0_MISSIONERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of MissionError_Enable register, to avoid silent dropping of errors.
Offset: 0x438
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G10_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G10_PADCTLREG_reg_scr_wrap was equal to 2'b10. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G10_PADCTLREG_reg_scr_wrap was equal to 2'b01.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G10_PADCTLREG_reg was equal to 2'b10. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G10_PADCTLREG_reg was equal to 2'b01.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G10_err_collator was equal to 2'b10. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G10_err_collator was equal to 2'b01.

PADCTL_G10_EC_ERRSLICE0_MISSIONERR_INJECT_0

Offset: 0x43c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G10_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Assert the inject_error_2 output for Register Parity Error to PADCTL.G10_PADCTLREG_reg_scr_wrap to allow for error injection. 1'b0 -> De-Asseert inject_error_2 output. **0** = DISABLE **1** = ENABLE
1	0x0	**ERR1**: 1'b1 -> Assert the inject_error_1 output for Register Parity Error to PADCTL.G10_PADCTLREG_reg to allow for error injection. 1'b0 -> De-Asseert inject_error_1 output. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
0	0x0	**ERR0**: 1'b1 -> Assert the inject_error_0 output for Register Parity Error to PADCTL_G10_err_collator to allow for error injection. 1'b0 -> De-Asseert inject_error_0 output. **0** = DISABLE **1** = ENABLE

PADCTL_G10_EC_ERRSLICE0_LATENTERR_ENABLE_0

Offset: 0x440
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G10_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.G10_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.G10_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.G10_PADCTLREG_reg 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.G10_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL_G10_err_collator 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL_G10_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_G10_EC_ERRSLICE0_LATENTERR_FORCE_0

Offset: 0x444
Read/Write: WO
Parity Protection: N
Shadow: N

SCR Protection: PADCTL_G10_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.G10_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.G10_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL_G10_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_G10_EC_ERRSLICE0_LATENTERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of LatentError_Enable register, to avoid silent dropping of errors.
Offset: 0x448
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G10_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G10_PADCTLREG_reg_scr_wrap was equal to 2'b00 or 2'b11. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G10_PADCTLREG_reg_scr_wrap was equal to 2'b01 or 2'b10, but no latent error.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G10_PADCTLREG_reg was equal to 2'b00 or 2'b11. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G10_PADCTLREG_reg was equal to 2'b01 or 2'b10, but no latent error.

Bit	Reset	Description
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G10_err_collator was equal to 2'b00 or 2'b11. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G10_err_collator was equal to 2'b01 or 2'b10, but no latent error.

PADCTL_G10_EC_ERRSLICE0_COUNTER_RELOAD_0
Offset: 0x450
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G10_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.G10_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
1	0x0	**ERR1**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.G10_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
0	0x0	**ERR0**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL_G10_err_collator 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

8.4.25 G11 PAD Control Registers

NOTE:

The G11 PAD Control Registers are collectively called FSI_PADCTL_A2 in the System Address Map.

The binary “x” is turned to “0” for the Reset and PROD hex value. Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as “x” or “Undefined” actually means that the bit could be either “0” or “1” after Reset, whereas a bit

PROD value as “x” or “Don’t-care” means that either “0” or “1” can be written to the bit by software Initialization.

PADCTL_G11_SOC_GPIO61_0
Offset: 0x0
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G11_SCR_SCR_SOC_GPIO61_0
Reset: 0x00000434 (0bxxxx,xxxx,xxxx,x00x,xxx0,x1x0,0011,0100)

Bit	Reset	Description
18	DISABLE	**SEL_DPD**: **0** = DISABLE **1** = ENABLE
17	DISABLE	**E_DPD**: **0** = DISABLE **1** = ENABLE
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD

Bit	Reset	Description
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G11_CFG2TMC_SOC_GPIO61_0

Offset: 0x4
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G11_SCR_SCR_SOC_GPIO61_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G11_SOC_GPIO62_0

Offset: 0x8
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G11_SCR_SCR_SOC_GPIO62_0
Reset: 0x00000434 (0bxxxx,xxxx,xxxx,x00x,xxx0,x1x0,0011,0100)

Bit	Reset	Description
18	DISABLE	**SEL_DPD**: **0** = DISABLE **1** = ENABLE
17	DISABLE	**E_DPD**: **0** = DISABLE **1** = ENABLE
12	DISABLE	**E_SCHMT**: **0** = DISABLE **1** = ENABLE
10	SFIO	**GPIO_SF_SEL**: **0** = GPIO **1** = SFIO

Bit	Reset	Description
8	DISABLE	**E_LPDR**: **0** = DISABLE **1** = ENABLE
7	DISABLE	**E_LPBK**: **0** = DISABLE **1** = ENABLE
6	DISABLE	**E_INPUT**: **0** = DISABLE **1** = ENABLE
5	ENABLE	**E_IO_HV**: **0** = DISABLE **1** = ENABLE
4	TRISTATE	**TRISTATE**: **0** = PASSTHROUGH **1** = TRISTATE
3:2	PULL_DOWN	**PUPD**: **0** = NONE **1** = PULL_DOWN **2** = PULL_UP **3** = RSVD
1:0	RSVD0	**PM**: **0** = RSVD0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3

PADCTL_G11_CFG2TMC_SOC_GPIO62_0

Offset: 0xc
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G11_SCR_SCR_SOC_GPIO62_0
Reset: 0x00000000 (0bxxxx,xxx0,0000,xxx0,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:20	0x0	**CFG_CAL_DRVUP**
16:12	0x0	**CFG_CAL_DRVDN**

PADCTL_G11_EC_FEATURE_0

```
================================
 GLOBAL SLICE
================================
```

Offset: 0x400
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G11_SCR_EC_SCR_0
Reset: 0x00020001 (0b0000,0000,0000,0010,xxxx,xxxx,xx00,0001)

Bit	Reset	Description
31:16	0x2	**NUM_ERR**: Number of errors connected to this collator. This is passed as a build time option to the plugin
5:0	0x1	**NUM_ERR_SLICES**: Number of error slices supported by this error collator, does not include the GlobalSpace and is derived by ceil (NUM_ERR/32). SW shall first read this register to determine the number of slices and read the required number of Error_Status registers .

PADCTL_G11_EC_SWRESET_0

Offset: 0x404
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G11_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SWRST**: 1'b1 : Issue a SW reset to the Error Collator. This will reset all the registers(Except SCR), counters and logic of the Error Collator. SW can use this bit to flush errors logged into the error collator for ex, after Boot, SC7/8 exit. 1'b0 : Do nothing, reset value. This bit is auto-cleared.

PADCTL_G11_EC_MISSIONERR_TYPE_0

Offset: 0x408
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G11_SCR_EC_SCR_0
Reset: 0x00000005 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0101)

Bit	Reset	Description
5:0	0x5	**CODE**: This register indicates the fault code of the error line based on the value of MISSIONERR_INDEX Register. This can be used by a fault handling agent to triage an error without requiring device-specific code.The possible values of this field are: 6'd0 : None 6'd1 : Parity Error on internal data path 6'd2 : ECC SEC Error on internal data path 6'd3 : ECC DED Error on internal data path 6'd4 : Comparator Error 6'd5 : Register Parity Error 6'd6 : Parity Error from on-chip SRAM/Fifo 6'd7 : ECC SEC Error from on-chip SRAM/Fifo 6'd8 : ECC DED Error from on-chip SRAM/Fifo 6'd9 : Clock Monitor Error 6'd10 : Voltage Error 6'd11 : Temperature Error 6'd12 : CRC Error on internal data path 6'd16 : SW Corrected Error 6'd17 : SW Uncorrected Error 6'd18 : SW Generic Error 6'd20 : ECC SED Error on internal data path 6'd21 : ECC SED Error from on-chip SRAM/Fifo 6'd32 : Other HW Corrected Error 6'd33 : Other HW Uncorrected Error All other values : Reserved for future use.

PADCTL_G11_EC_CURRENT_COUNTER_VALUE_0

Offset: 0x40c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G11_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
8:0	0x0	**VALUE**: Provides the current value of the counter corresponding to the error in MissionErr_Index Register. Default provides the value of error 0 counter. Bit[8] is the ovrerflow bit post which the counter saturates and does not counter further.

PADCTL_G11_EC_MISSIONERR_INDEX_0

Offset: 0x414
Read/Write: R/W
Parity Protection: Y
Shadow: N

SCR Protection: PADCTL_G11_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**IDX**: BINARY Encoded. For error number 32, register should be programmed with value 0x20. Write to this register with Error number will update: - MISSIONERR_TYPE Register with the Error-Code for the Error. - CURRENT_COUNTER_VALUE Register with the error's SEC/DED Counter. - MISSIONERR_USERVALUE with value of the first error_<i>_user signal. SW can use this to trage the error. number shall update the MISSIONERR_TYPE register with the error code and the Current_Counter_Value register with the value of the errors SEC/DED counter. SW can use this register to triage the error.

PADCTL_G11_EC_CORRECTABLE_THRESHOLD_0

Offset: 0x418
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G11_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**COUNT**: Threshold value for all SEC Fault Reporting Units connected to this error collator. SEC Errors are logged once the threshold is reached and the overflow bit is set. 7'b0 : Log SEC error after receiving 1 Error. 7'b1 : Log SEC error after receiving 2 Errors. ... 7'bFF : Log SEC error after receiving 256 Errors.

PADCTL_G11_EC_MISSIONERR_INJECT_UNLOCK_0

Offset: 0x41c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G11_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VALUE**: Writes to ERRSLICE_XXX_MISSIONERR_INJECT registers are disabled until this register is written with a value of 0xE1. This is to prevent an inadvertent safety error injection in the safety plugin due to: 1. A fault on ERRSLICE_XXX_MISSIONERR_INJECT register itself. 2. Erroneous SW. The register shall be written with a value of 0x0 to reestablish the lock after user has completed the error injection testing. 0xE1 : Unlock the MISSIONERR_INJECT Register 0x0 : Lock the MISSIONERR_INJECT Register **0** = LOCK **225** = UNLOCK

PADCTL_G11_EC_ERRSLICE0_MISSIONERR_ENABLE_0

```
==================================
 ERROR SLICE - 0
==================================
```

Offset: 0x430
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G11_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.G11_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.G11_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL.G11_PADCTLREG_reg 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL.G11_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Mission Error Reporting for Register Parity Error from PADCTL_G11_err_collator 1'b0 -> Disable Mission Error Reporting for Register Parity Error from PADCTL_G11_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_G11_EC_ERRSLICE0_MISSIONERR_FORCE_0

Offset: 0x434
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G11_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.G11_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL.G11_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Mission Error Reporting for Register Parity Error from PADCTL_G11_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_G11_EC_ERRSLICE0_MISSIONERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of MissionError_Enable register, to avoid silent dropping of errors.
Offset: 0x438
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G11_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G11_PADCTLREG_reg_scr_wrap was equal to 2'b10. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G11_PADCTLREG_reg_scr_wrap was equal to 2'b01.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G11_PADCTLREG_reg was equal to 2'b10. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G11_PADCTLREG_reg was equal to 2'b01.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G11_err_collator was equal to 2'b10. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G11_err_collator was equal to 2'b01.

PADCTL_G11_EC_ERRSLICE0_MISSIONERR_INJECT_0

Offset: 0x43c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G11_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Assert the inject_error_2 output for Register Parity Error to PADCTL.G11_PADCTLREG_reg_scr_wrap to allow for error injection. 1'b0 -> De-Asseert inject_error_2 output. **0** = DISABLE **1** = ENABLE
1	0x0	**ERR1**: 1'b1 -> Assert the inject_error_1 output for Register Parity Error to PADCTL.G11_PADCTLREG_reg to allow for error injection. 1'b0 -> De-Asseert inject_error_1 output. **0** = DISABLE **1** = ENABLE
0	0x0	**ERR0**: 1'b1 -> Assert the inject_error_0 output for Register Parity Error to PADCTL_G11_err_collator to allow for error injection. 1'b0 -> De-Asseert inject_error_0 output. **0** = DISABLE **1** = ENABLE

PADCTL_G11_EC_ERRSLICE0_LATENTERR_ENABLE_0

Offset: 0x440
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G11_SCR_EC_SCR_0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x111)

Bit	Reset	Description
2	0x1	**ERR2**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.G11_PADCTLREG_reg_scr_wrap 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.G11_PADCTLREG_reg_scr_wrap **0** = DISABLE **1** = ENABLE
1	0x1	**ERR1**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL.G11_PADCTLREG_reg 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL.G11_PADCTLREG_reg **0** = DISABLE **1** = ENABLE
0	0x1	**ERR0**: 1'b1 -> Enable Latent Error Reporting for Register Parity Error from PADCTL_G11_err_collator 1'b0 -> Disable Latent Error Reporting for Register Parity Error from PADCTL_G11_err_collator **0** = DISABLE **1** = ENABLE

PADCTL_G11_EC_ERRSLICE0_LATENTERR_FORCE_0

Offset: 0x444
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G11_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.G11_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

Bit	Reset	Description
1	0x0	**ERR1**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL.G11_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE
0	0x0	**ERR0**: 1'b1 -> Force Assertion of Latent Error Reporting for Register Parity Error from PADCTL_G11_err_collator 1'b0 -> Do Nothing **0** = NOFORCE **1** = FORCE

PADCTL_G11_EC_ERRSLICE0_LATENTERR_STATUS_0

SW must write 1 to clear the fields of this register.
Bits in this register continue to be logged independent of the value of LatentError_Enable register, to avoid silent dropping of errors.
Offset: 0x448
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: PADCTL_G11_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G11_PADCTLREG_reg_scr_wrap was equal to 2'b00 or 2'b11. 1'b0 -> Error_2_pulse[1:0] for Register Parity Error from PADCTL.G11_PADCTLREG_reg_scr_wrap was equal to 2'b01 or 2'b10, but no latent error.
1	0x0	**ERR1**: 1'b1 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G11_PADCTLREG_reg was equal to 2'b00 or 2'b11. 1'b0 -> Error_1_pulse[1:0] for Register Parity Error from PADCTL.G11_PADCTLREG_reg was equal to 2'b01 or 2'b10, but no latent error.
0	0x0	**ERR0**: 1'b1 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G11_err_collator was equal to 2'b00 or 2'b11. 1'b0 -> Error_0_pulse[1:0] for Register Parity Error from PADCTL_G11_err_collator was equal to 2'b01 or 2'b10, but no latent error.

PADCTL_G11_EC_ERRSLICE0_COUNTER_RELOAD_0

Offset: 0x450

Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: PADCTL_G11_SCR_EC_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ERR2**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.G11_PADCTLREG_reg_scr_wrap 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
1	0x0	**ERR1**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL.G11_PADCTLREG_reg 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD
0	0x0	**ERR0**: 1'b1 -> Reload Error Counter for Register Parity Error from PADCTL_G11_err_collator 1'b0 -> Do Nothing **0** = NORELOAD **1** = RELOAD

8.5 GPIO Controllers

8.5.1 Overview

The GPIO Controllers allow control of General Purpose IO pins for a variety of system functions. In addition to the basic GPIO functionalities, the GPIO Controllers also support virtualization and a secure access control mechanism with firewalls. The following key words are used in the description of the GPIO Controllers:

- **GPIO Pin**

  A single General-Purpose I/O pin which may share a physical I/O pin with other functional I/O pins through Pin Multiplexing (PinMux).
  For details of how the PinMux mechanism works, refer to the **Programming Model** section of the PinMux chapter in this TRM.
  PinMux details are provided by NVIDIA in the form of a spreadsheet, referred to just as *Pinmux*. Please ask your NVIDIA representative for details.

- **GPIO Port**

A collection of up to eight GPIO Pins that are physically close to one another.

- **GPIO Controller**

  A GPIO Controller instance controlling up to eight GPIO Ports, or 64 GPIO Pins, based on the address map available for the GPIO Controller instance's APB client.

- **Hypervisor (HV)**

  A low-level software layer managing hardware resources and memory, and supporting one or more Guest Operating Systems by providing communication and isolation to them.

- **Guest OS**

  The OS running on a Virtual Machine (VM).

8.5.1.1 Relevant Chapters in the TRM

- Clock and Reset Controller (CAR)
- Interrupt Controllers
- vGIC Interrupt Controller
- Multi-Purpose I/O Pins and Pin Multiplexing (PinMux)
- Timers

8.5.1.2 Features

There are altogether nine GPIO Controllers in the Orin SoC.

- Six in VDD_SOC core power (TOP cluster).
- One in VDD_RTC core power (AON cluster).
- Two in VDD_FSI core power (FSI cluster).

Each of the GPIO Controller maps to a set of GPIO Ports in association with their Pad Control Groups as shown in the table below.

Table 8.56 GPIO Controllers and Port Mapping

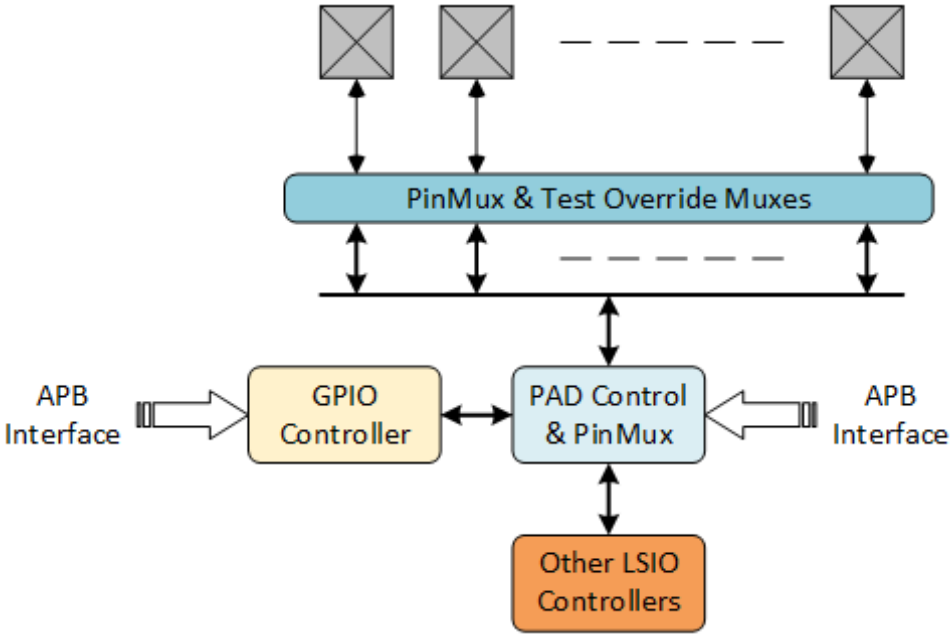
GPIO Controller	GPIO Ports	Pad Control Groups	Padlet Group / Voltage Domain
GPIO_CTL0	A (8)*	G5	VDDIO_G5
	B (1)*	DEBUG	VDDIO_DEBUG
	AC (8)* AD (4)*	G7	VDDIO_G7

GPIO Controller	GPIO Ports	Pad Control Groups	Padlet Group / Voltage Domain
GPIO_CTL1	X (8)* Y (8)* Z (8)*	G2	VDDIO_G2
GPIO_CTL2	M (8)* N (8)*	EDP	VDDIO_EDP
	P (8)* Q (8)* R (6)*	G3	VDDIO_G3
GPIO_CTL3	K (8)* L (4)*	PEX_CTL	VDDIO_PEX_CTL
	AE (2)*	UFS	VDDIO_UFS
	AF (4)*	PEX_CTL_2	VDDIO_PEX_CTL_2
	AG (8)*	PEX_CTL_3	VDDIO_PEX_CTL_3
GPIO_CTL4	G (8)* H (8)* I (7)*	G4	VDDIO_G4
GPIO_CTL5	C (8)* D (4)*	QSPI	VDDIO_QSPI
	E (8)* F (6)*	EQOS	VDDIO_EQOS
	J (6)*	SDMMC1_HV	VDDIO_SDMMC1_HV
GPIO_AON	AA (8)* BB (4)*	AO_HV	VDDIO_AO_HV
	CC (8)* DD (3)*	AO	VDDIO_AO
	EE (8)* GG (1)*	SYS	VDDIO_SYS
GPIO_FSI_CTL0	S (8)* T (2)*	G9	VDDIO_G9
	W (2)*	G11	VDDIO_G11
GPIO_FSI_CTL1	U (8)* V (1)*	G10	VDDIO_G10

*: Number of GPIO Pins in the given GPIO Port.

The Orin GPIO Controllers support over 200 GPIO Pins (217 to be exact), most of which share physical I/O pins with other functional (including testing) I/O's under the control of PinMux and Pad Control. The following diagram illustrates the functional structure encompassing the GPIO Controller, PAD Control, PinMux, and physical I/O pins.

Figure 8.38 Functional Structure Encompassing GPIO Controller, LSIO Controllers, PAD Control, PinMux, and Physical I/O Pins



For details of the PAD Control and PinMux, refer to the PinMux chapter in this TRM.

8.5.2 Functional Description

8.5.2.1 Basic GPIO Controller Structure

The GPIO Controller manages groups of GPIO Pins for

- Input with the physical pin input available at the corresponding GPIO register (GPIO_<*i*>/<*iii*>_INPUT_0<*j*>_0) to be read by software.
- Output with software-written values in the corresponding GPIO register (GPIO_<*i*>/<*iii*>_OUTPUT_VALUE_0<*j*>_0).

where <*i*> = A, B, C, D, E, F, G, H, I, J, K, L, M, N, P, Q, R, X, Y, Z, AA, AC, AD, AE, AF, AG, BB, CC, DD, EE, GG,

<*iii*> = S, T, W, U, V, and

<*j*> ∈ {0, 1, 2, 3, 4, 5, 6, 7}.

NOTE:

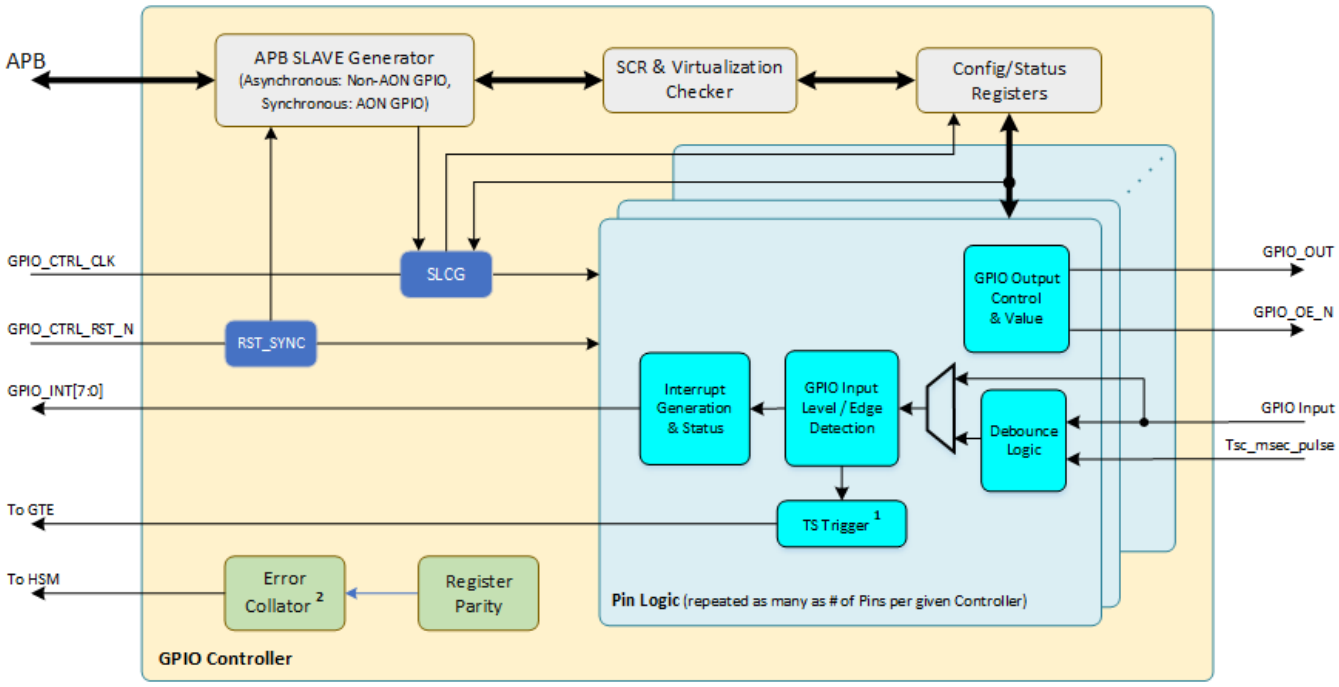
For details of these GPIO Input and Output registers, refer to the **Registers** section below.

The GPIO Controller at the GPIO Pin level has provisions to perform the following functions:

- Configuring the GPIO Pin for Input/Output direction.
- Sampling a GPIO Input Data to the corresponding register and Driving GPIO Output Data from the corresponding register to a GPIO Pin.
- Generating Interrupts based on the programmed polarity levels and transitions.
- De-bouncing and Filtering of input signals.

The following diagram shows the basic blocks in the GPIO Controller.

Figure 8.39 GPIO Controller Block Diagram



NOTE:

1. The TimeStamp (TS) Trigger is in the AON GPIO Controller only.
2. The Error Collator (EC) of the AON GPIO Controller is outside the AON GPIO Controller, but still in the Always ON power region.

8.5.2.1.1 APB SLAVE Generator and Config/Status Registers

The APB Slave Generator interfaces with the Control BackBone (CBB) through the APB interface. Apart from the proper protocol interfacing and termination, this block also implements the necessary Clock crossing structures.

The Config/Status Register block which interfaces with the SoC via the APB bus through the APB Interface Generator, consists of all GPIO Controllers Configuration Registers and Status Registers listed as follows:

- GPIO_<*i*>_ENABLE_CONFIG_0<*j*>_0,
- GPIO_<*i*>_DEBOUNCE_THRESHOLD_0<*j*>_0,
- GPIO_<*i*>_OUTPUT_CONTROL_0<*j*>_0,
- GPIO_<*i*>_INTERRUPT_CLEAR_0<*j*>_0,
- GPIO_<*i*>_INTERRUPT_STATUS_G<*jj*>_0,
- GPIO_<*i*>_INT<*jj*>_ROUTE_MAPPING_0,

where <*i*> = A, B, C, D, E, F, G, H, I, J, K, L, M, N, P, Q, R, X, Y, Z, AA, AC, AD, AE, AF, AG, BB, CC, DD, EE, GG,

<*j*> ∈ {0, 1, 2, 3, 4, 5, 6, 7}, and

<*jj*> = 0, 1, 2, 3, 4, 5, 6, 7.

- GPIO_<*iii*>_ENABLE_CONFIG_0<*j*>_0

- GPIO_<*iii*>_DEBOUNCE_THRESHOLD_0<*j*>_0,
- GPIO_<*iii*>_OUTPUT_CONTROL_0<*j*>_0,
- GPIO_<*iii*>_INTERRUPT_CLEAR_0<*j*>_0,
- GPIO_<*iii*>_INTERRUPT_STATUS_G<*jj*>_0,
- GPIO_<*iii*>_INT<*jj*>_ROUTE_MAPPING_0,

where <*iii*> = S, T, W, U, V,

<*j*> ∈ {0, 1, 2, 3, 4, 5, 6, 7}, and

<*jj*> = 0, 1, 2, 3, 4, 5, 6, 7.

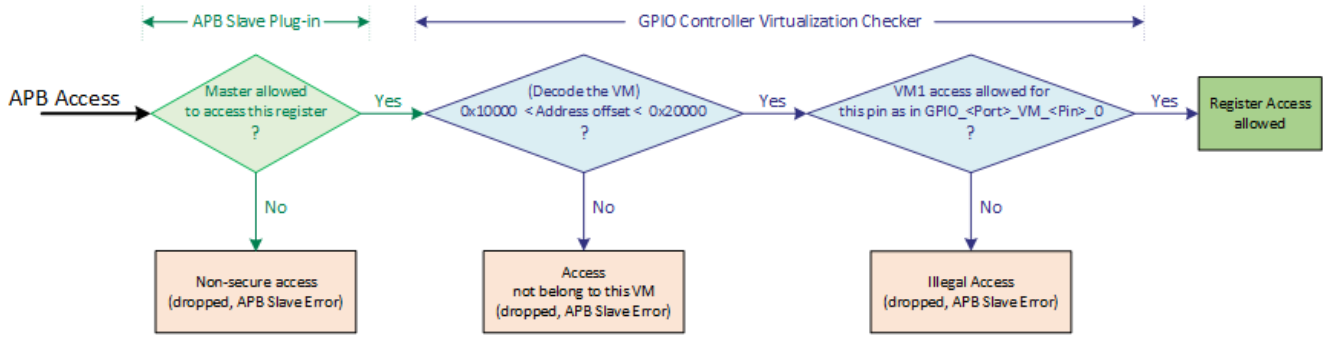
NOTE:

For details of these GPIO Input and Output registers, refer to the **Registers** section below.

8.5.2.1.2 SCR and Virtualization Checker

The Virtualization checker validates the incoming APB transaction with respect to VM config attributes set up by Hypervisor, then accordingly permits or denies access to the specific register. Non-privileged Guest OS access violations are dropped with APB slave error.

Figure 8.40 GPIO Virtualization Checker for GPIO_CTL<0/1/2/3/4/5>



8.5.2.1.3 GPIO Debounce Logic

The Debounce logic is a simple Glitch Filter with a filtering granularity of 1 msec. The Debounce logic receives a constant 1-msec pulse from the Timestamp System Counter (TSC), a toggle-based event to indicate 1-msec intervals. The pulse from the TSC is double synchronized and edge detected for reference filtering.

8.5.2.1.4 GPIO Input and Level/Edge Detection

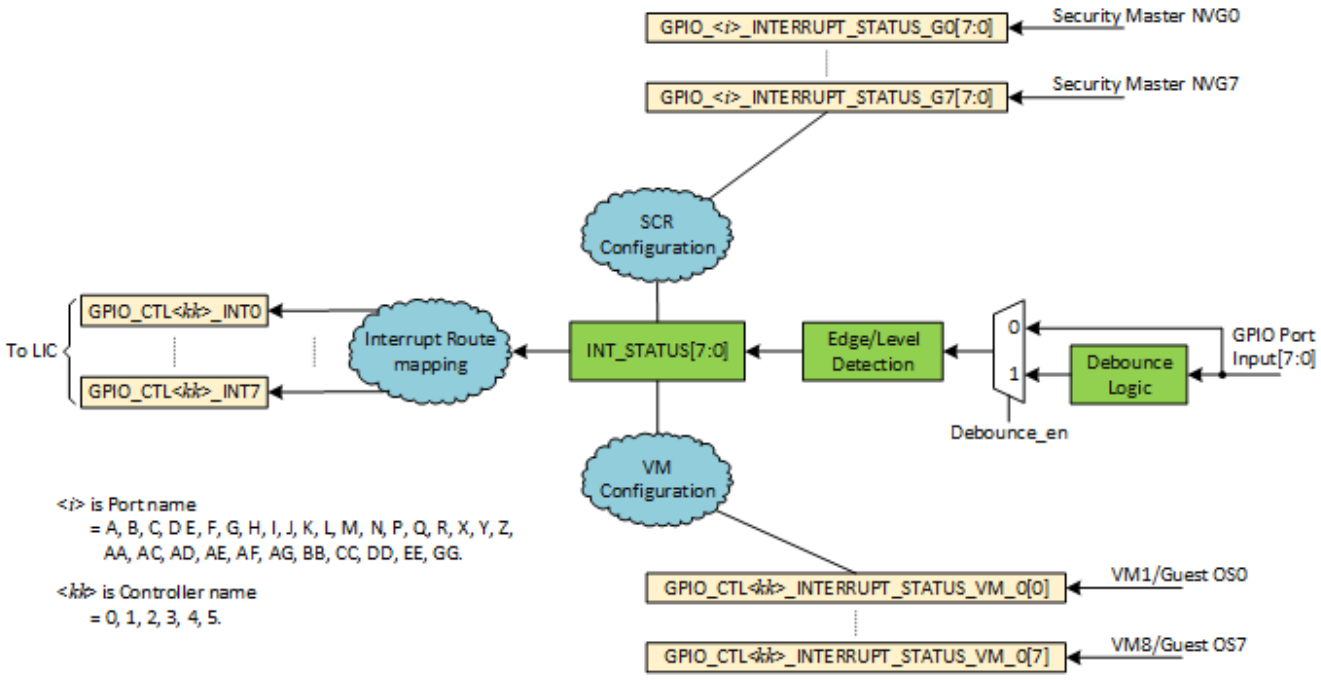
This block processes the GPIO inputs (from pads through PinMux logic) for edge/level detection, updates the Status Registers, and forwards the triggers to the Interrupt generation block & TS trigger block. The GPIO Pin input level is updated in the GPIO_<*i*>/<*iii*>_INPUT_0<*j*>_0 register for software access.

8.5.2.1.5 Interrupt Generation and Status

This block generates Interrupt triggers based on the output of the GPIO Input Level/Edge Detection block. The generated Interrupt triggers are then routed to the LIC followed by generated Interrupt Status.

The following diagram shows the consolidated view of GPIO Interrupt generation and Status updates per Pin for each Security Master (NVG0 through NVG7) as well as Virtual Machine (VM1/ Guest OS0 through VM8/Guest OS7).

Figure 8.41 GPIO Controller Interrupt and Interrupt Status Scheme



For details of the GPIO Interrupts, refer to the **GPIO Interrupts** section below.

8.5.2.1.6 TS Trigger

The TS (TimeStamp) Trigger block generates triggers for the timestamping block, i.e., the Generic Timestamping Engine (GTE) block outside of the GPIO Controller based on the output of the GPIO Input & Level/Edge Detection block.

8.5.2.1.7 GPIO Output Control and Value

This part of logic drives the GPIO output as programmed in the GPIO OUT register (GPIO_<*i*>/<*iii*>_OUTPUT_VALUE_<*j*>_0) by software. The GPIO Controller output propagates through the PinMux logic then to the physical pin/pad.

8.5.2.1.8 GPIO Clock Gating

The GPIO Controller implements the Second Level Clock Gating (SLCG) scheme for power optimization. In line with the per-pin configuration capabilities, the SLCG structures are realized such that

- the Config Registers logic Clock is gated except when Register access is active,

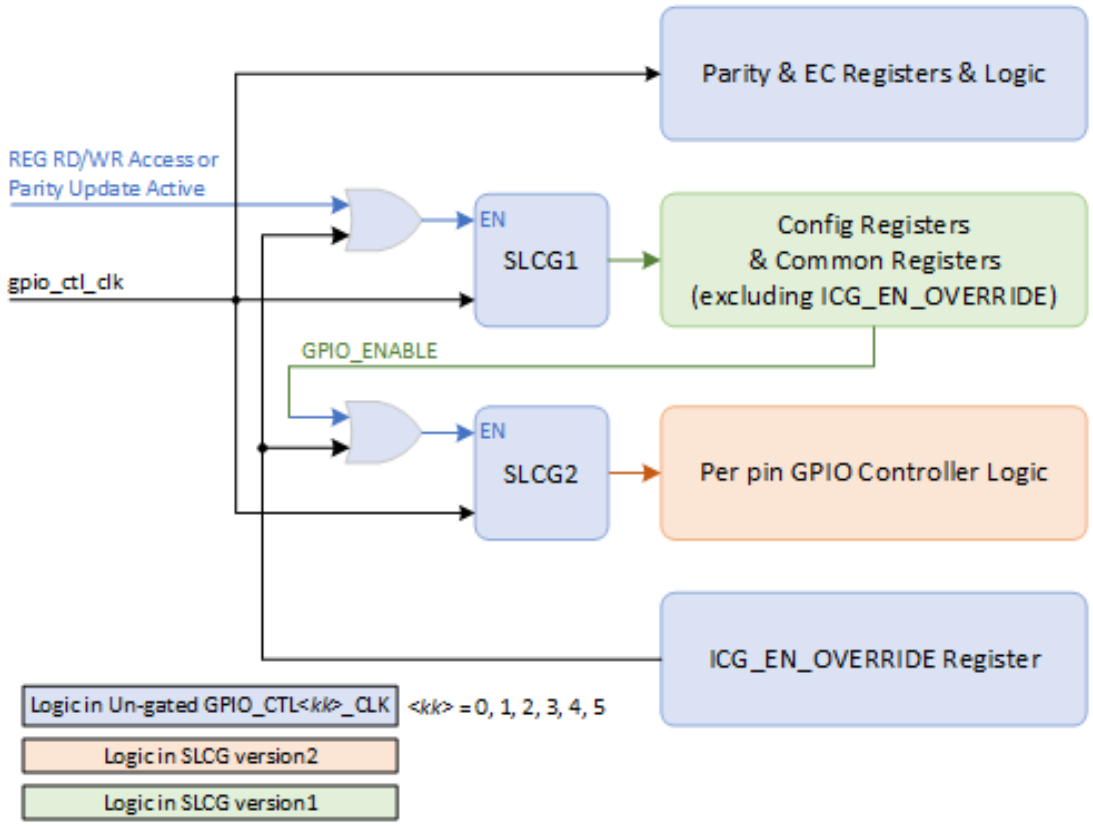
- the per-pin GPIO logic Clock is gated except when the GPIO Pin is active in either Input or Output mode, *i.e.* (GPIO_<*i*>_ENABLE_CONFIG_<*j*>_0.GPIO_ENABLE == 1).

Both Second Level Clock Gating (SLCG) structures, SLCG1 and SLCG2, can be bypassed with the GPIO_<*k*>/FSI<*m*>_ICG_EN_OVERRIDE_0 register which is on the un-gated version of the GPIO_CTL<*kk*>_CLK. In addition, the Parity and Error Collator (EC) registers-related logic is also on the un-gated version of the GPIO_CTL<*kk*>_CLK for the purpose of capturing possible Parity errors.

NOTE:

<*k*> = CTL0, CTL1, CTL2, CTL3, CTL4, CTL5, AON,
<m> = 0, 1, and
<*kk*> = 0, 1, 2, 3, 4, 5.

Figure 8.42 GPIO Controller Clock Gating Overview



Clock Gating by CAR during L2 Reset

Under L2 Reset, the CAR logic gates the Clock for Non-AON and FSI GPIO Controllers which have multiple Reset versions. This is done to avoid meta-stable issues in the GPIO logic that is not reset by the L2 Reset.

Such CAR logic is un-gated after L2 Reset by enabling the special GPIOs to hold the state prior to the L2 Reset.

For details of CAR, refer to the Clock and Reset Controller (CAR) chapter in this TRM.

8.5.2.2 GPIO TimeStamping Interface

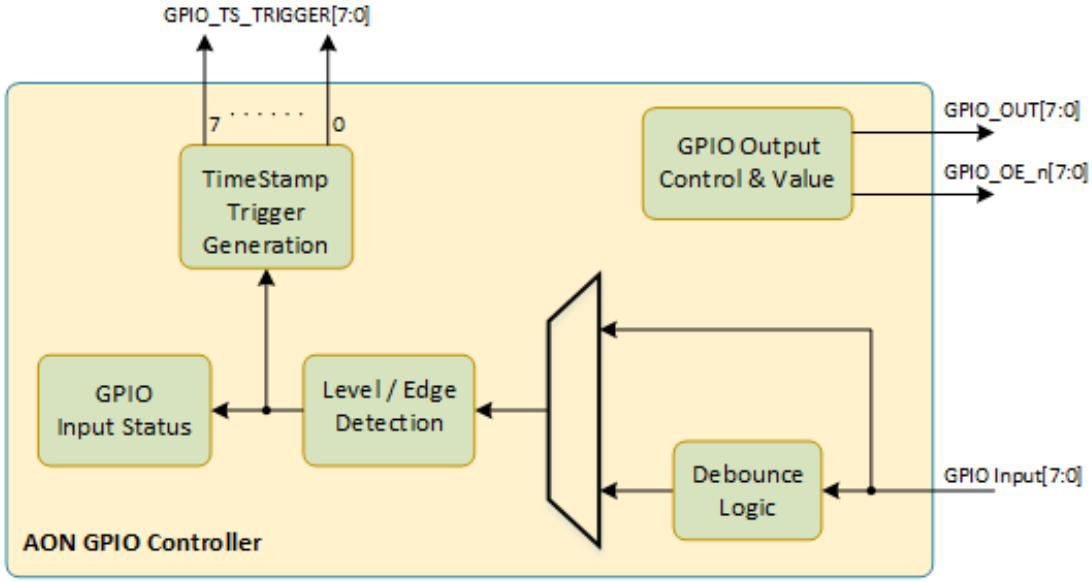
The Generic Timestamping Engine (GTE) logic in Orin performs event timestamping based on the change in input trigger events and logs the Event ID along with the Global Timestamp in a FIFO. The module that stores the timestamping value is called Capture Slice. Multiple Capture slices are muxed/arbitrated into one Timestamping Slice via the APB. Triggering events based on GPIO input pin changes are forwarded to the GTE by the GPIO Controller to enable the timestamping mechanism.

For details of the Generic Timestamping Engine (GTE), refer to the Timers chapter in this TRM.

The timestamping provision is limited only to GPIOs handled by the AON GPIO Controller. Hence, use cases or system requirements that call for accurate time synchronization between Orin and a connected device, the connected device must be connected to Orin through an AON GPIO Pin to take advantage of the GTE-backed timestamping to achieve reliable hardware-based time synchronization.

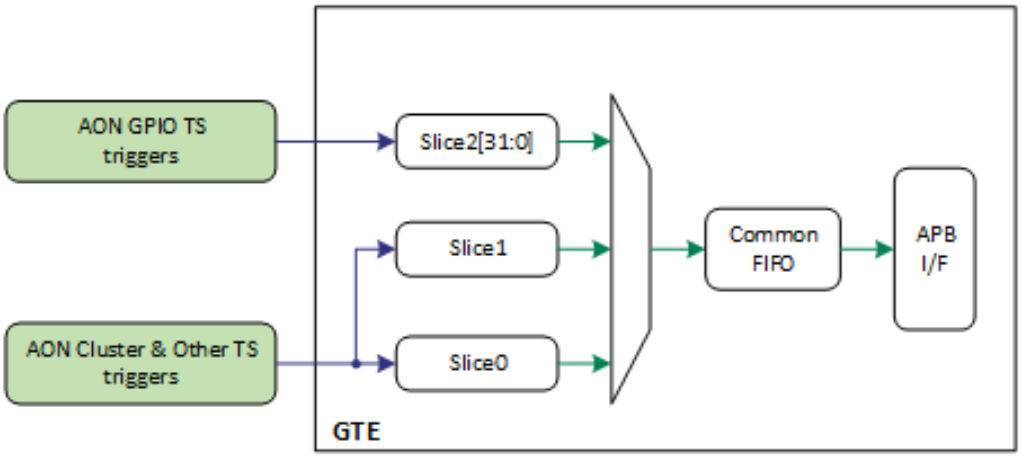
Timestamping of a GPIO Pin handled by Non-AON GPIO Controllers can be achieved by timestamping the Interrupts generated form the GPIO Pin to the LIC. The following diagram shows the timestamping triggering structure in the AON GPIO Controller.

Figure 8.43 TimeStamping Triggering Structure from AON GPIO



In Orin, there are 32 AON GPIO Pins. Their timestamp (TS) triggers together with other AON clusters TS triggers are forwarded to the GTE and collected in the Common FIFO inside the GTE as shown in the following diagram.

Figure 8.44 AON to GTE Connection Structure



8.5.2.2.1 GPIO AON Timestamping Usage

The AON cluster's main purpose is the handling of various sensor related triggers along with the Sensor Processing Engine (SPE). One such timestamping of sensor events is the measurement of pulse widths and identification of certain specific sequence event.

The timestamp tick granularity is at 32 ns in AON time, with 20-bit count to support event detection up to 33 msec.

8.5.2.3 GPIO Clock and Reset Control

Each GPIO Controller is independent and can operate out of the best possible Clock source meeting its Clock policy constraints and close to its physical location. Following is the summary of high-level Clock requirements.

- APB Clock for register interface and GPIO Core Clock for GPIO Controller logic separated to facilitate transaction-based Clock gating.
- APB Clock and GPIO Core Clock completely asynchronous.
- No relationship between Clocks of different GPIO Controllers.
- AON GPIO Controller core Clock connected to APB Clock to minimize the number of Clocks in AON cluster.
- One software Reset (bit) and software/driver entity for each GPIO Controller instance.

- Non-AON GPIO Controllers with L1 warm Reset to handle mixed-Reset cases. Controller Clocks during L2 Reset.
- FSI GPIO Controller Clock limited to XTAL Clock source of 40 MHz.

8.5.2.3.1 System Reset Mapping to GPIO Controllers

The GPIO Controllers can be subject to various Resets outlined as follows:

L2 Cold Reset

- Covering AON GPIO Controller logic, Config registers, and Common registers. Out of these, Common registers are on the secure Reset version which does not have software Reset provision. Register Parity and logic are on the same Reset as the register being protected.

L1 Warm Reset

- Covering GPIO Controller/logic associated with CPU_PWR_REQ. L1 Warm Reset does not have software Reset provision. Also, there is flush/soft Reset synchronous clear option for EC registers.

L2 Warm Reset

- Covering GPIO Controller logic, Config registers, Common registers, and EC registers/logic. Out of these, Common registers and EC registers/logic are on the secure Reset version which does not have software Reset provision. Register Parity and logic are on the same Reset as the register being protected.
- Covering FSI GPIO Controller logic, Config registers, and Common registers. All the logic in on the secure Reset version which does not have software Reset provision. Register Parity and logic are on same Reset as the register being protected.

Further, the following table shows the mapping of different System Resets to GPIO Controllers and logic in both AON and Non-AON clusters.

Table 8.57 System Reset Mapping to GPIO Controller

Reset Source Level	Associated Sources	GPIO Controller in Non-AON Cluster	GPIO Regs and Logic on CPU_PWR_REQ	GPIO Controller in AON Cluster	Remark
Cold	Chip Reset (SYS_RST_N)	Yes	Yes	Yes	
Warm	SC7 Wake Reset	Yes	Yes	No	AON GPIO Controller must survive SC7 state. Hence, SC7 Wake does not reset it.
Level 2	All WDT Debug Reset	Yes	No	Yes	The AON GPIO Controller together with all AON I/Os get reset in L2.

Reset Source Level	Associated Sources	GPIO Controller in Non-AON Cluster	GPIO Regs and Logic on CPU_PWR_REQ	GPIO Controller in AON Cluster	Remark
Level 1c	Software Cold Reset	Yes	Yes	Yes	GPIO pin CPU_PWR_REQ, resets the I/O state only on Cold and Warm Resets in L1. CPU power pins in GPIO_CTL4 are out of Reset at L1b.
Level 1b	All WDT POR Reset, HSM Internal System Reset	Yes	Yes	Yes	
Level 1a	Sensor Reset	Yes	Yes	Yes	
Level 0	External POR Reset	Yes	Yes	Yes	GPIO pin CPU_PWR_REQ, resets the I/O state only on Cold and Warm Resets in L0.

NOTE:

Yes - Controller/Logic is reset with indicated system Reset variant.
No - Controller/Logic is not reset with indicated system Reset variant.

GPIO Controller Logic on L1 Reset

To ensure that the CPU power rail is not switched OFF during L2 Reset cycle and thereby preserving the MCA state, the GPIO Pin CPU_PWR_REQ does not get reset during L2 Reset. Instead, only the I/O state gets reset on Cold and Warm Resets below L2 (i.e. L0/L1) Reset. In addition, the GPIO_CTL4 GPIO Controller is associated with CPU_PWR_REQ, thus requiring L1 Reset when Reset is needed.

Similarly, to retain registers states through L2 Reset, the Non-AON GPIO Controller's Parity logic, Error Collator logic, and registers are also on L1 Reset.

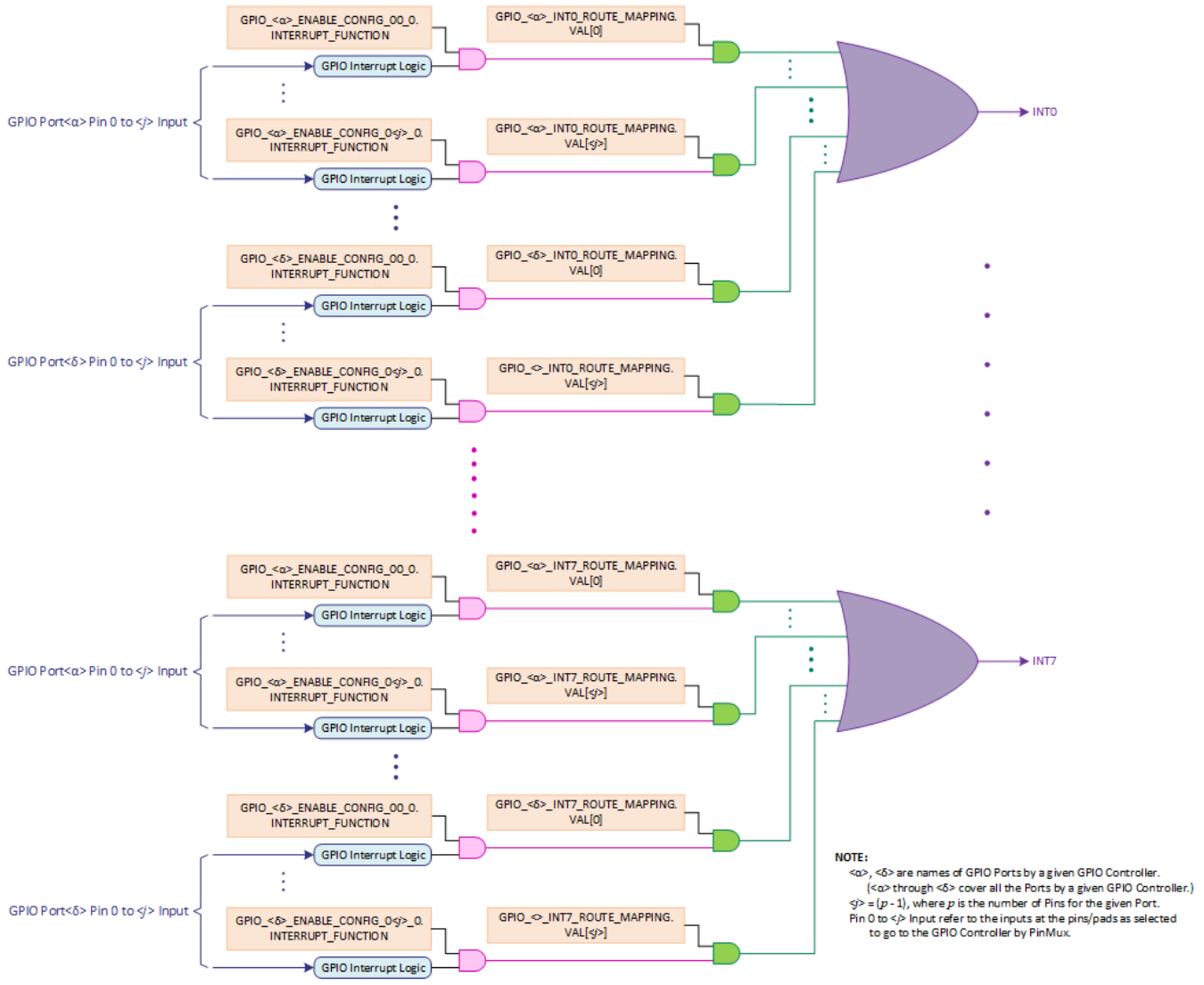
8.5.2.4 GPIO Interrupts

This section provides details of Interrupts and the Interrupt Status Registers.

8.5.2.4.1 Interrupt Generation

Each GPIO input change can trigger an Interrupt, reflected in the Interrupt Status. Each Interrupt line has a simple route mapping indicating which GPIO Pin event is mapped to trigger the Interrupt line to the LIC as shown in the following diagram.

Figure 8.45 Interrupt Generation per GPIO Controller



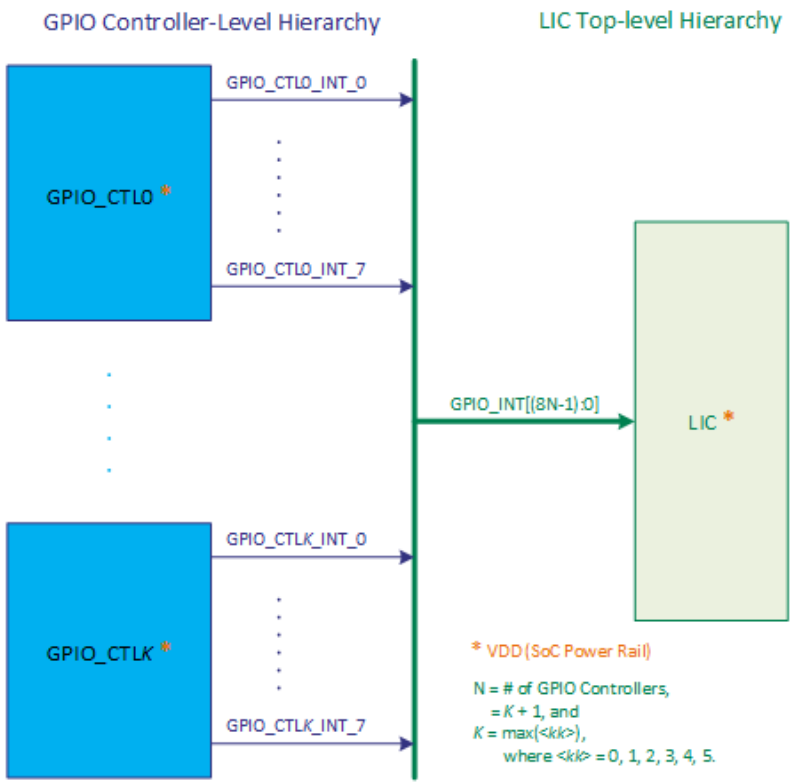
8.5.2.4.2 Non-AON GPIO Interrupts Connectivity

Each Non-AON GPIO Controller generates 8 Interrupts and they are directly connected to the LIC at the top level. This ensures that software entities serving the Interrupt know which GPIO Controller has generated the Interrupt by reading the vector status from the LIC.

The Interrupt triggers for multiple Pins are combined and channeled to the LIC from each of the GPIO Controllers. Additional latency in Interrupt handling is expected while servicing GPIO Interrupts as this involves multiple MMIO Reads. The Interrupt handler needs to read the GPIO Interrupt Status Registers to figure out the exact GPIO Pin source that triggered the Interrupt.

The following diagram shows how the Interrupts are routed to the LIC.

Figure 8.46 Non-AON GPIO Controllers Interrupts Generation and Connection to LIC



8.5.2.4.3 AON GPIO Interrupts Connectivity

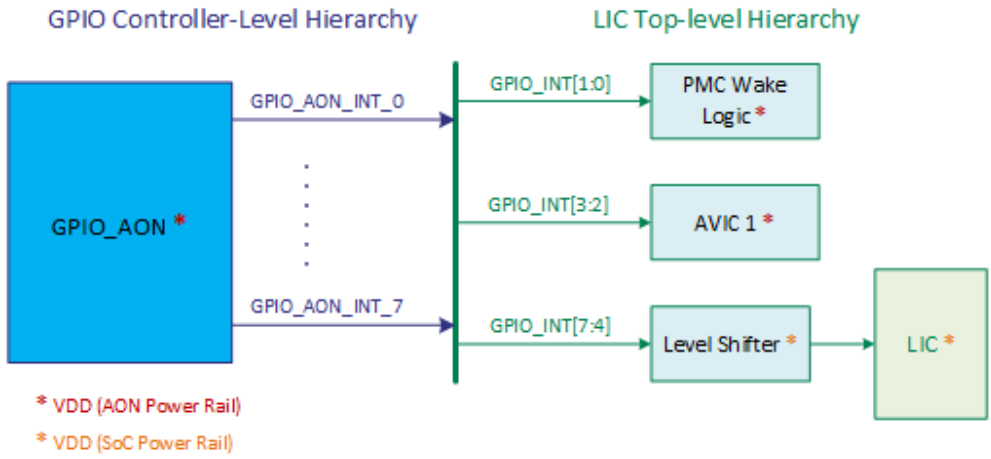
The AON GPIO Controller also generates eight Interrupt outputs, not all of which are routed to the LIC though. Specifically,

- two Interrupts (aon_gpio_int[1:0]) are connected as wake event to facilitate GPIO wake from SC7,
- two other Interrupts (aon_gpio_int[3:2]) are connected to AON VIC to trigger Interrupts to SPE, and
- remaining four Interrupts (aon_gpio_int[7:4]) are routed to the LIC through a level shifter in the VDD_SOC power domain.

The AON GPIO Controller does not support Virtualization. Hence, all features related VM configuration and outputs for the Guest OS are unused.

The following diagram shows the breakdown of AON GPIO Controller Interrupt routing.

Figure 8.47 AON GPIO Controller Interrupt Generation and Connection to LIC



8.5.2.4.4 Non-AON GPIO Controller Interrupts in the Multi-CPU and Virtualized Environment

In Orin, there are multiple CPUs across sub-systems referred to as Security Masters. In the Context of this TRM, Security Master, Security Domain, and Security Group all refer to one such CPU with a set of security privileges in the system. Depending on the CPU's capability and platform usage models, a Security Master can:

- run a rich OS like Linux/Android,
- be an instance of Virtual Machine (also referred to as Guest OS), where each Virtual Machine forms a Security Domain in software, and
- run firmware components on Cortex-R5 sub-systems, also known as clusters.

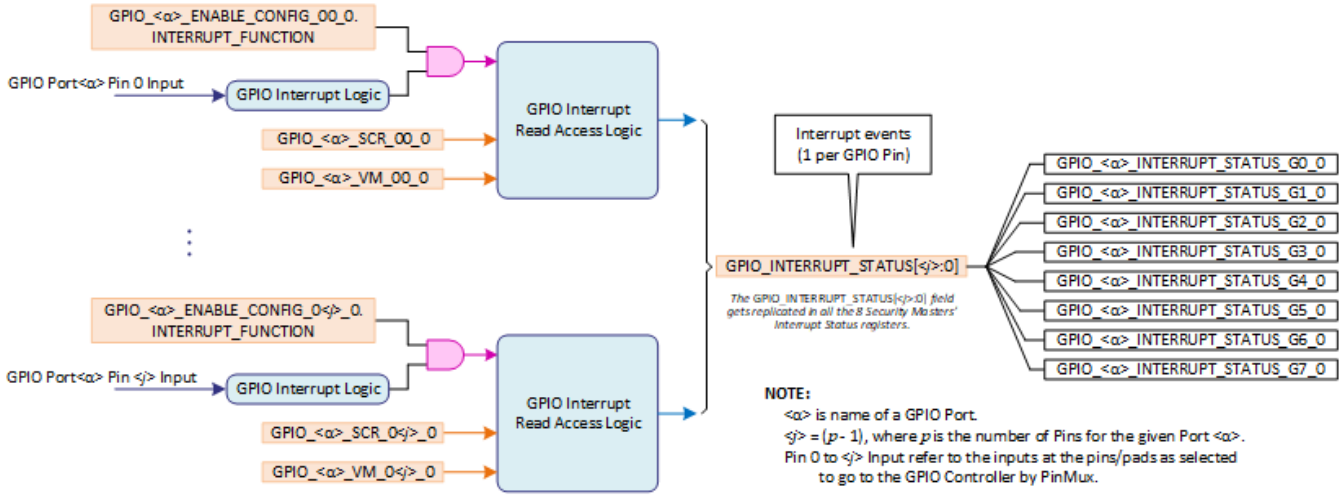
A GPIO Pin can be owned by any of the aforementioned software entities of a Security Group. Hence, each GPIO Controller is designed to generate Interrupts for each independent software entity, i.e., one Interrupt per VM. This, in turn, leads to each Interrupt being routed to one of the following types of software stacks:

- CPU Drivers
- CPU Guest OSs
- CPU Trust Zone (TZ, a specialized VM)
- Firmware for any embedded Falcon or Cortex-R5, as in BPMP, SPE, SCE, and APE, with an external Interrupt from the LIC.

To support the virtualization requirements, the Non-AON GPIO Controller, in addition, also maintains different sets of Status Registers, detailing the Interrupt Status per Pin for every

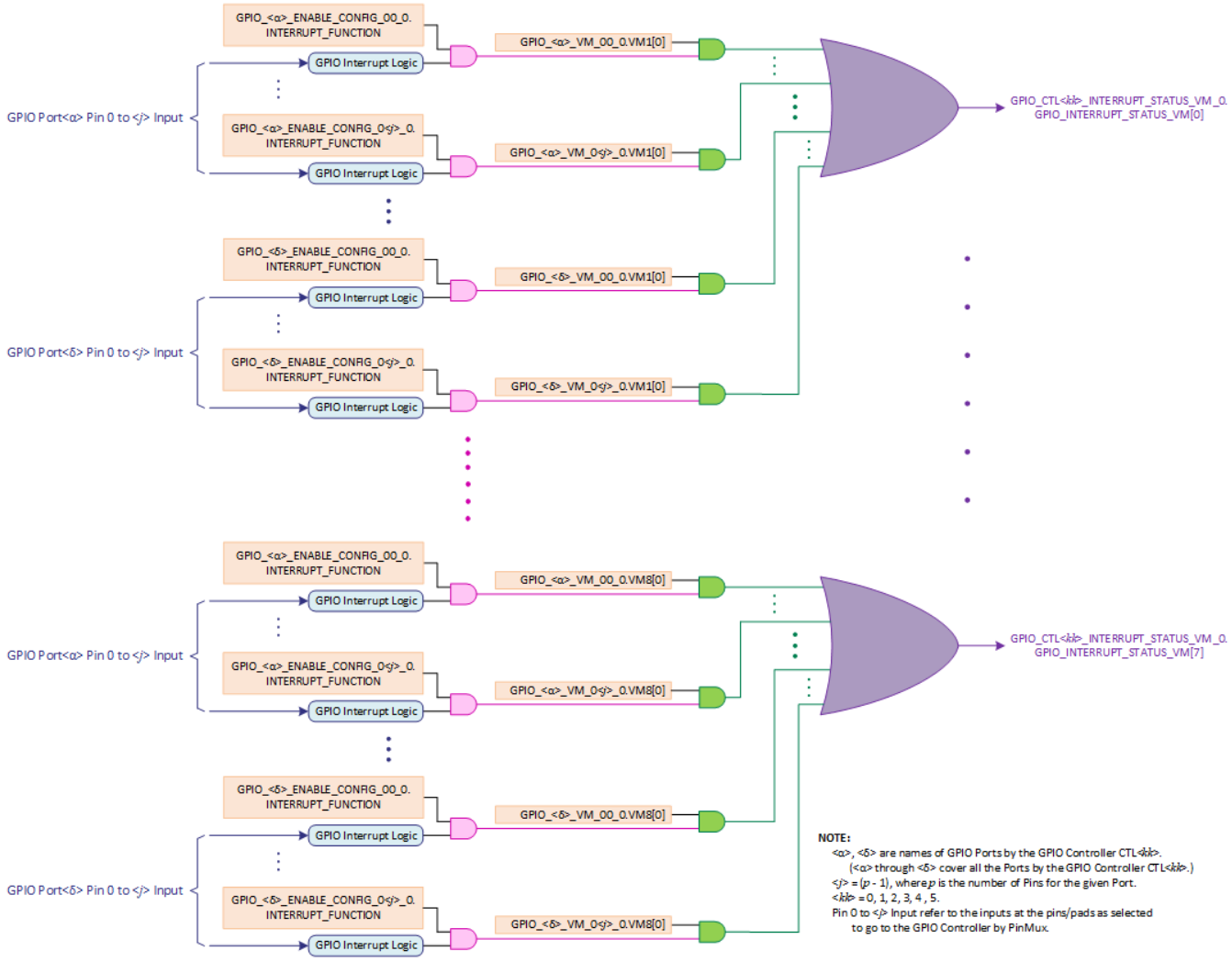
security domain which is readable by the privileged Security Master, as shown in the following diagram.

Figure 8.48 Interrupt Status per Security Master



There is also abstracted Interrupt Status for Virtualization requirements that is one Interrupt Status per Guest OS as shown in the following diagram.

Figure 8.49 VM Interrupt Status Register per Non-AON GPIO Controller



8.5.2.5 AON GPIO Controller

The AON subsystem controlled by the SPE has around 20+ GPIO Pins spread in three pin groups, namely AO, AO_HV, and SYS, and controlled by one AON GPIO Controller connected to the local APB bridge in the AON cluster. To ensure that these GPIO Pins are controlled during SC7, all the control and status for these GPIO Pins are maintained as part of the AON cluster, not in the SoC domain. Further, the AON GPIO Controller is not power-gated.

The AON GPIO controller has its own separate 64-KiB page to facilitate a flexible usage model during SC7 and run time.

8.5.2.6 APB Slave Error support

The following register access errors are handled by the GPIO Controller and reported as APB Slave error on the APB bus. The APB Master has provision to log the details of violating transfers for diagnosis purpose.

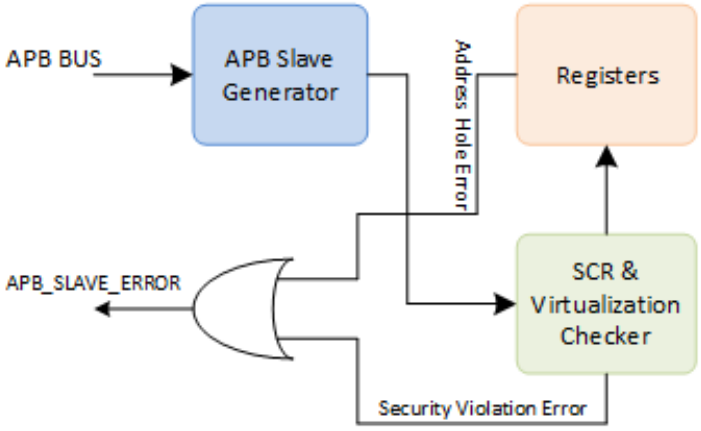
- Address Hole Errors.
- Virtualization access violations.

To facilitate first-level debugging the logic generates

- 0xDEAD_1001 in case of address hole Writes or Reads,
- 0xDEAD_1007 in case of Virtualization access violations,

The following diagram shows the high-level structure of this block.

Figure 8.50 GPIO APB Slave Error Generation



8.5.2.7 Virtualization

With the presence of Virtualization, the address translation

Guest (VA) → Intermediate (VA) → Host (PA)

happens in two phases as illustrated below.

- Page tables for Guest (VA) → Intermediate (VA) maintained by the Guest OS.
- Page tables for Intermediate (VA) → Host (PA) maintained by the Hypervisor.

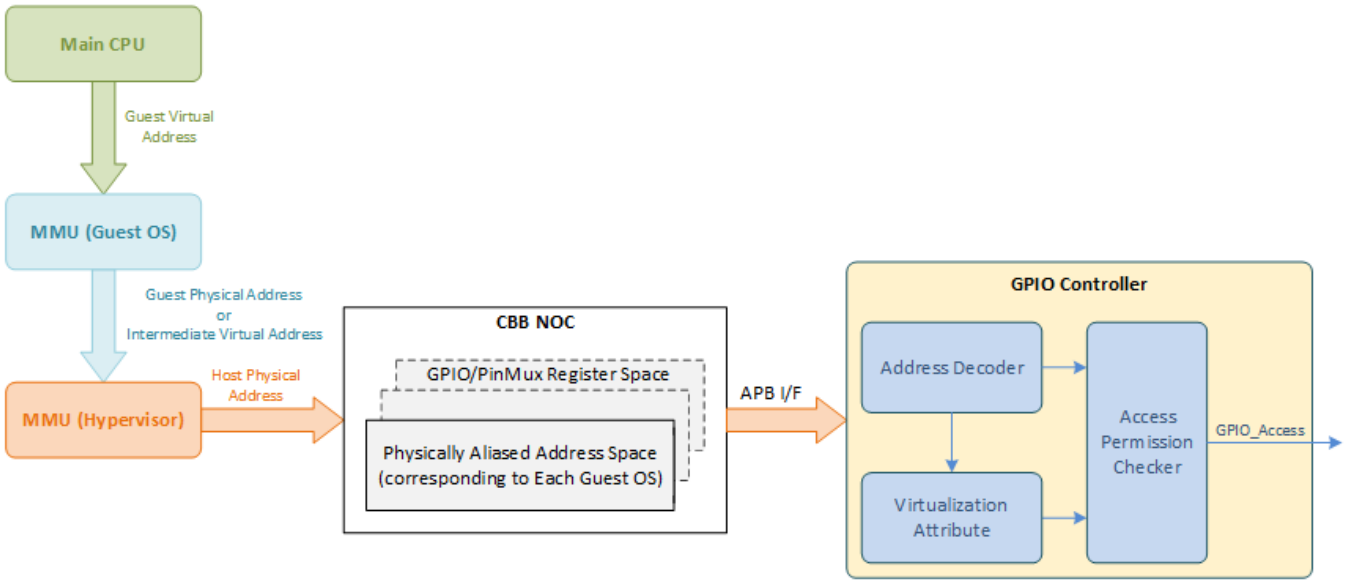
NOTE:

VA: Virtual address
PA: Physical Address

Under virtualization, the address space of the registers per GPIO Controller is replicated along with table-based mapping.

- Physically aliasing the GPIO/PinMux address space corresponding to the eight Guest OS's as supported, with Hypervisor mapping same resources in different physical addresses for the different Guest OS.
- Mapping table/registers fully controlled by Hypervisor to configure each GPIO/PinMux register to set Guest OS's access privileges.
- CBB NOC decoding physically aliased address space to same physical block, GPIO Controller.
- GPIO Controller comparing register access to decide the Guest OS's access right of the register based on Virtualization mapping register settings as explained in the **SCR and Virtualization Checker** section.

Figure 8.51 Virtualization Provisioning for GPIOs



8.5.2.7.1 Virtualization of AON and FSI GPIO Controller

The AON GPIOs are typically owned by the Sensor Processing Engine (SPE) or a single Guest OS, which does not require Virtualization support. The AUTOSAR version targeted in the FSI cluster does not support virtualization and therefore GPIO virtualization is not required. In summary, virtualization provision is not supported for the AON GPIO Controller, nor for the FSI GPIO Controllers.

8.5.3 Registers

This section provides the high-level description of GPIO registers organization, configuration, and status. The configuration details are given for one instance of GPIO Pin, whereas the status details are given for one instance of GPIO Port.

8.5.3.1 System Address Map (AMAP)

From System Address Map point of view, a 4-KiB address space is reserved for each GPIO Controller to support a maximum of eight GPIO Ports, each of which maps to a maximum of 8 GPIO Pins. Hence, each GPIO Controller maps up to 64 GPIO Pins, depending on the corresponding physical pin/pad locations and various other floorplan considerations. This 4-KiB space is repeated for each GPIO Controller in a 64-KiB space for each of the following domains:

- Non-AON
- AON
- FSI

In particular, there are

- Six GPIO Controllers (CTL0, CTL1, CTL2, CTL3, CTL4, CTL5) in the Non-AON domain,
- One GPIO Controller (AON) in the AON domain, and
- Two GPIO Controllers (FSI_CTL0, FSI_CTL1) in the FSI domain.

Further, each GPIO Controller's registers are in two categories, Configuration registers and Common registers.

The Configuration registers are responsible for

- GPIO mode (input vs output) selection and control,
- Debouncing control, and
- other features accessible by the Guest OS/VM.

The Common Control registers, contrary to what its name may suggest, are privileged registers only accessible by Hypervisor (HV) or the equivalent, but not by the Guest OS/VM, to govern

- Virtualization settings

For Non-AON GPIO Controllers, the address space is physically aliased to support virtualization. Each of these GPIO Controllers groups is mapped in a 64-KiB address page partitioned into 4-KiB sub-pages for each of the individual GPIO Controllers in the group. The 64-KiB page is physically aliased eight times to support eight VM/Guest OS access. The Common registers are mapped in its own 64-KiB address page partitioned into 4-KiB sub-pages for each of the individual GPIO Controllers in the group.

The following diagrams illustrate the System Address Map organizations of the Non-AON, AON, and FSI GPIO Controllers.

Figure 8.52 Non-AON GPIO Controllers AMAP Organization

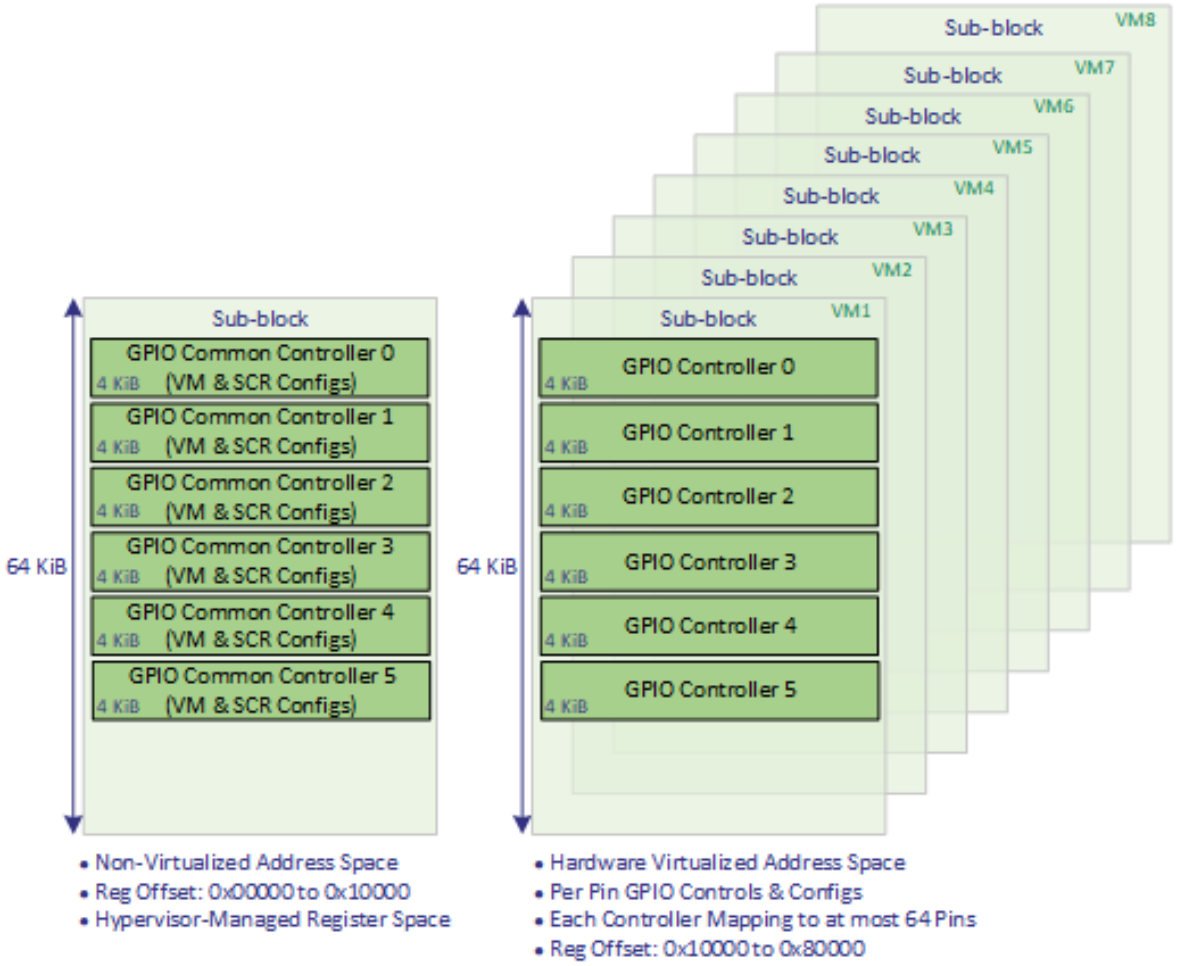


Figure 8.53 AON GPIO Controller AMAP Organization

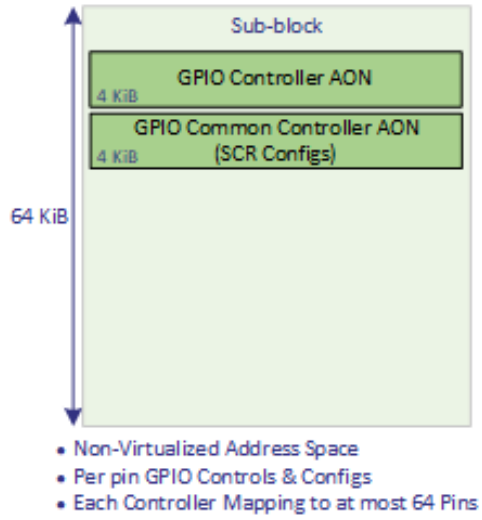
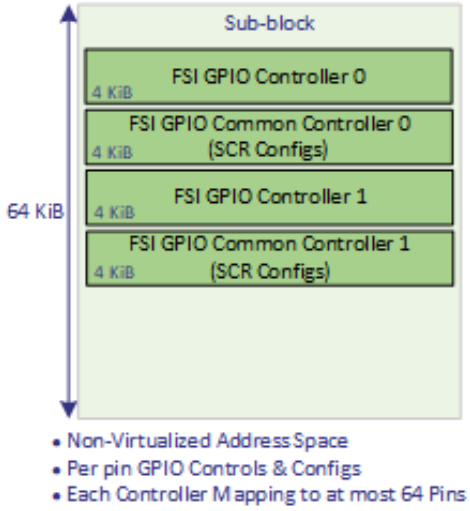


Figure 8.54 FSI GPIO Controller AMAP Organization



8.5.3.2 Registers Details

The following tables summarize all the GPIO registers for Non-AON and AON GPIO Controllers, as well as FSI GPIO Controllers.

Table 8.58 Non-AON and AON GPIO Registers Details

GPIO Control Registers	
GPIO_<*i*>_ENABLE_CONFIG_0<*j*>_0	
GPIO_<*i*>_DEBOUNCE_THRESHOLD_0<*j*>_0	
GPIO_<*i*>_INPUT_0<*j*>_0	
GPIO_<*i*>_OUTPUT_CONTROL_0<*j*>_0	
GPIO_<*i*>_OUTPUT_VALUE_0<*j*>_0	
GPIO_<*i*>_INTERRUPT_CLEAR_0<*j*>_0	
GPIO_<*i*>_INTERRUPT_STATUS_G<*jj*>_0	
GPIO Common Control Registers	
GPIO_<*ii*>_VM_0<*j*>_0	
GPIO_CTL<*kk*>_INTERRUPT_STATUS_VM_0	
GPIO_<*k*>_ICG_EN_OVERRIDE_0	
GPIO_<*i*>_INT<*jjj*>_ROUTE_MAPPING_0	
GPIO_<*i*>_SPARE_0	

NOTE:

<*i*> = A, B, C, D E, F, G, H, I, J, K, L, M, N, P, Q, R, X, Y, Z, AA, AC, AD, AE, AF, AG, BB, CC, DD, EE, GG (Non-AON and AON GPIO Ports)

<*ii*> = A, B, C, D E, F, G, H, I, J, K, L, M, N, P, Q, R, X, Y, Z, AC, AD, AE, AF, AG (non-AON GPIO Ports)

<*j*> ∈ {0, 1, 2, 3, 4, 5, 6, 7} (GPIO Pin within a GPIO Port)

<*jj*> = 0, 1, 2, 3, 4, 5, 6, 7 (Security Master)

<*jjj*> = 0, 1, 2, 3, 4, 5, 6, 7 (VM / Guest OS)

<*k*> = CTL0, CTL1, CTL2, CTL3, CTL4, CTL5, AON (non-AON and AON GPIO Controllers)

<*kk*> = 0, 1, 2, 3, 4, 5 (non-AON GPIO Controllers)

Table 8.59 FSI GPIO Registers Details

GPIO FSI Control Registers	
GPIO_<*iii*>_ENABLE_CONFIG_0<*j*>_0	
GPIO_<*iii*>_DEBOUNCE_THRESHOLD_<*j*>_0	
GPIO_<*iii*>_INPUT_0<*j*>_0	
GPIO_<*iii*>_OUTPUT_CONTROL_0<*j*>_0	
GPIO_<*iii*>_OUTPUT_VALUE_0<*j*>_0	
GPIO_<*iii*>_INTERRUPT_CLEAR_0<*j*>_0	
GPIO_<*iii*>_INTERRUPT_STATUS_G<*jj*>_0	
GPIO FSI Common Control Registers	
GPIO_FSI<*m*>_ICG_EN_OVERRIDE_0	
GPIO_<*iii*>_INT<*jjj*>_ROUTE_MAPPING_0	
GPIO_FSI<*m*>_SPARE_0	

NOTE:

<*iii*> = S, T, W, U, V (FSI GPIO Ports)

<*j*> ∈ {0, 1, 2, 3, 4, 5, 6, 7} (GPIO Pin within a GPIO Port)

<*jj*> = 0, 1, 2, 3, 4, 5, 6, 7 (Security Master)

<*jjj*> = 0, 1, 2, 3, 4, 5, 6, 7 (VM / Guest OS)

<*m*> = 0, 1 (FSI) (FSI non-AON GPIO Controllers)

8.5.3.2.1 Configuration/Status Registers

This sub-section provides the description of GPIO Configuration and Status Registers. The Configuration Registers details are given for one instance of the GPIO Pin, whereas the Status Registers details are given for one instance of the GPIO Port.

Per-Pin Configuration Registers

This sub-section covers the GPIO Configuration registers for one instance of the GPIO Pin, with the Pin number within a GPIO Port denoted by "0<*j*>".

NOTE:

There are maximally 8 GPIO Pins in one GPIO Port. And some GPIO Ports have less than 8 GPIO Pins.
Hence, <*j*> ∈ {0, 1, 2, 3, 4, 5, 6, 7}.

GPIO_<*i*>/<*iii*>_ENABLE_ CONFIG_0<*j*>_0,

where <*i*> = A, B, C, D, E, F, G, H, I, J, K, L, M, N, P, Q, R, X, Y, Z, AA, AC, AD, AE, AF, AG, BB, CC, DD, EE, GG,

<*iii*> = S, T, W, U, V, and

<*j*> ∈ {0, 1, 2, 3, 4, 5, 6, 7}.

This register, available for each GPIO Pin (<*j*>) in a GPIO Port (<*i*>, <*iii*>), configures the given GPIO Pin.

Field	Bit	Reset	Description
TIMESTAMPING_FUNCTION	7	0	This field allows the GPIO input change event to create a timestamp event. When enabled based on the settings in TRIGGER_TYPE & TRIGGER LEVEL time-stamp trigger is sent out to TSC. It is effective only when (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE) && (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.IN_OUT == IN). **0** = DISABLE. **1** = ENABLE.
INTERRUPT_FUNCTION	6	0	This field provides the Interrupt selection for the GPIO Input changes based on ▪ trigger type (TRIGGER_TYPE) ▪ trigger polarity (TRIGGER LEVEL) It is effective only when (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE). **0** = DISABLE. **1** = ENABLE. When ((GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE) && (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.INTERRUPT_FUNCTION == 1)), the Interrupt Status is updated in the Interrupt Status Register (GPIO_<*i*>_INTERRUPT_STATUS_G<*jj*>_0) for all the VM's, based on the settings of GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.TRIGGER_TYPE & GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.TRIGGER LEVEL.

Field	Bit	Reset	Description
DEBOUNCE_FUNCTION	5	0	This field provides the Debounce filter selection on the GPIO Input. It is effective only when (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE). **0** = DISABLE (no Debounce filter). **1** = ENABLE. When (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.DEBOUNCE_FUNCTION == ENABLE), the debouncing interval value is defined by (GPIO_<*i*>_DEBOUNCE_THRESHOLD_0<*j*>_0.DEBOUNCE_THRESHOLD).
TRIGGER LEVEL	4	0	This field selects the polarity of the GPIO Input trigger for the given trigger type. It is effective only when (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE). When (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.TRIGGER_TYPE == NO TRIGGER), this field is irrelevant. When (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.TRIGGER_TYPE == LEVEL), **0** = Trigger on level Low. **1** = Trigger on Level High. When (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.TRIGGER_TYPE == SINGLE EDGE), **0** = Trigger on Falling Edge. **1** = Trigger on Raising Edge. When (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.TRIGGER_TYPE == BOTH EDGES), this field is irrelevant.
TRIGGER_TYPE	3:2	0	This field selects the GPIO Input trigger type (condition). It is effective only when (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE). **0** = NO_TRIGGER. **1** = LEVEL. **2** = SINGLE EDGE. **3** = DOUBLE_EDGE (i.e. both Low-High, High-Low edges are detected as event).
IN_OUT	1	0	This field configures the GPIO for Input or Output. It is effective only when (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE). **0** = IN. **1** = OUT.
GPIO_ENABLE	0	0	This field is the GPIO functionality selection control required for output/input selection. It acts as a global qualifier for all the functions of the specific GPIO. It is also used for SLCG (Second Level Clock Gating) per pin-based GPIO logic. **0** = DISABLE. **1** = ENABLE. When Disabled, the GPIO is floated, i.e. both input and output are disabled. At this point, if PinMux selects the GPIO, the I/O is in High-Z state.

GPIO_<*i*>/<*iii*>_DEBOUNCE_THRESHOLD_0<*j*>_0

where <*i*> = A, B, C, D, E, F, G, H, I, J, K, L, M, N, P, Q, R, X, Y, Z, AA, AC, AD, AE, AF, AG, BB, CC, DD, EE, GG,

<*iii*> = S, T, W, U, V, and

<*j*> ∈ {0, 1, 2, 3, 4, 5, 6, 7}.

This register, available for each GPIO Pin (<*j*>) in a GPIO Port (<*i*>, <*iii*>), specifies the Debounce threshold of the given GPIO Pin.

Field	Bit	Reset	Description
DEBOUNCE_ THRESHOLD	7:0	0	This field holds the Debounce threshold in msec to specify the debounce interval. **0** = No Debounce. **1** ~ N-1: Debouncing interval of 1 msec to (N-1) msec. **NOTE:** The actual Debounce interval has an accuracy granularity of 1 msec. Hence, when programmed to "1", the actual Debouncing interval is between 1 msec and 2 msec.

GPIO_<*i*>/<*iii*>_INPUT_0<*j*>_0

where <*i*> = A, B, C, D, E, F, G, H, I, J, K, L, M, N, P, Q, R, X, Y, Z, AA, AC, AD, AE, AF, AG, BB, CC, DD, EE, GG,

<*iii*> = S, T, W, U, V, and

<*j*> ∈ {0, 1, 2, 3, 4, 5, 6, 7}.

This register, available for each GPIO Pin (<*j*>) in a GPIO Port (<*i*>, <*iii*>), specifies the GPIO Input value of the given GPIO Pin.

Field	Bit	Reset	Description
GPIO_IN	0	X	This field holds the GPIO Input value sampled after the specified Debouncing interval. It is effective only when (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE). It reflects the external value regardless of (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.IN_OUT) setting, so that the externally driven value is present in this field to aid debugging.

GPIO_<*i*>/<*iii*>_OUTPUT_CONTROL_0<*j*>_0

where <*i*> = A, B, C, D, E, F, G, H, I, J, K, L, M, N, P, Q, R, X, Y, Z, AA, AC, AD, AE, AF, AG, BB, CC, DD, EE, GG,

<*iii*> = S, T, W, U, V, and

<*j*> ∈ {0, 1, 2, 3, 4, 5, 6, 7}.

This register, available for each GPIO Pin (<*j*>) in a GPIO Port (<*i*>, <*iii*>), enables the given GPIO Pin to be Output.

Field	Bit	Reset	Description
GPIO_OUT_CONTROL	0	1	This field Indicates whether the GPIO as an output is Floated or Actively driven. It is effective only when ((GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE) && (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.IN_OUT == OUT)). It is floated regardless of the value specified here, When ((GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == DISABLE) || (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.IN_OUT == IN)). **0** = DRIVEN. **1** = FLOATED.

GPIO_<*i*>/<*iii*>_OUTPUT_VALUE_0<*j*>_0

where <*i*> = A, B, C, D, E, F, G, H, I, J, K, L, M, N, P, Q, R, X, Y, Z, AA, AC, AD, AE, AF, AG, BB, CC, DD, EE, GG,

<*iii*> = S, T, W, U, V, and

<*j*> ∈ {0, 1, 2, 3, 4, 5, 6, 7}.

This register, available for each GPIO Pin (<*j*>) in a GPIO Port (<*i*>, <*iii*>), specifies the GPIO Output value of the given GPIO Pin.

Field	Bit	Reset	Description
GPIO_OUT_VALUE	0	0	This field holds the GPIO output value to be driven out when (GPIO_<*i*>_OUTPUT_CONTROL_<*j*>_0.GPIO_OUT_CONTROL == DRIVEN). It is effective only when ((GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE) && (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.IN_OUT == OUT)).

GPIO_<*i*>/<*iii*>_INTERRUPT_CLEAR_0<*j*>_0

where <*i*> = A, B, C, D, E, F, G, H, I, J, K, L, M, N, P, Q, R, X, Y, Z, AA, AC, AD, AE, AF, AG, BB, CC, DD, EE, GG,

<*iii*> = S, T, W, U, V, and

<*j*> ∈ {0, 1, 2, 3, 4, 5, 6, 7}.

This register, available for each GPIO Pin (<*j*>) in a GPIO Port (<*i*>, <*iii*>), clears the Interrupt at the given GPIO Port/Pin.

Field	Bit	Reset	Description
GPIO_INTERRUPT_CLEAR	0	0	This field holds the Interrupt clear control bit for clearing the Interrupt at GPIO Pin <*j*> in GPIO Port <*i*>/<*iii*>. It is effective only when ((GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0. (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.INTERRUPT_FUNCTION == ENABLE)). **0** = DON'T CLEAR. **1** = CLEAR.

Per-Port Status Registers

This sub-section covers the GPIO Status Registers for one instance of the GPIO Port in response to all the 8 Security Masters, NVG0 through NVG7.

GPIO_<*i*>/<*iii*>_INTERRUPT_STATUS_G<*jj*>_0,

where <*i*> = A, B, C, D, E, F, G, H, I, J, K, L, M, N, P, Q, R, X, Y, Z, AA, AC, AD, AE, AF, AG, BB, CC, DD, EE, GG,

<*iii*> = S, T, W, U, V, and

<*jj*> = 0, 1, 2, 3, 4, 5, 6, 7.

This register, available for each GPIO Port (<*i*>) and Security Master (<*jj*>), indicates whether a GPIO Pin in the GPIO Port (<*i*>) has caused the assertion of the given Interrupt originated from a Security Master (<*jj*>).

Field	Bit	Reset	Description
GPIO_INTERRUPT_STATUS	[<*j*>*:0]	0	This field has as many bits as the number of GPIO Pins in the given GPIO Port, with each such bit indicating whether the corresponding GPIO Input change has resulted in Interrupt. Bit 0 is for Pin 0, Bit 1 is for Pin 1, ... Bit <*j*> is for Pin <*j*>. **0** = Interrupt not set. **1** = Interrupt set. It is effective only when ((GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE) && (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.INTERRUPT_FUNCTION == ENABLE)). The bit in this field is cleared based on GPIO_<*i*>/ <*iii*>_INTERRUPT_CLEAR_0<*j*>_0.GPIO_INTERRUPT_CLEAR. In a Virtualized environment, the status reflects only the GPIOs belonging to VM. For the GPIO bit positions not mapped to VM, the corresponding bit in this field is written with "0" by hardware for software to read this register correctly.

*: <*j*> = (*p* -1), where *p* is the number of GPIO Pins in a given GPIO Port <*i*>. And <*j*> ∈ {0, 1, 2, 3, 4, 5, 6, 7}.

8.5.3.2.2 GPIO Common Control Registers

This sub-section provides the description of GPIO Common Control registers that support Virtualization and Security Control features under the control of Hypervisor. These registers are only accessible by the Hypervisor but not by the Guest OS's (VM's) because their address space is not visible to the Guest OS's (VM's).

Common Control Registers for Virtualization Support

GPIO_<*ii*>_VM_0<*j*>_0,

where <*ii*> = A, B, C, D, E, F, G, H, I, J, K, L, M, N, P, Q, R, X, Y, Z, AC, AD, AE, AF, AG and

<*j*> ∈ {0, 1, 2, 3, 4, 5, 6, 7}.

This register, available for each Non-AON GPIO Port (<*ii*>), maps its GPIO Pin (<*j*>) to VM to specify its accessibility for each Guest OS.

Field	Bit	Reset	Description
VM8 (Guest OS7 Access Permission)	15:14	FULL	This field specifies the access permission for VM8 (Guest OS7). **00** = No Access for VM8 (Guest OS7). **01** = Read Only Access for VM8 (Guest OS7). **10** = Write Only Access for VM8 (Guest OS7). **11** = FULL access for VM8 (Guest OS7).
VM7 (Guest OS6 Access Permission)	13:12	FULL	This field specifies the access permission for VM7 (Guest OS6). **00** = No Access for VM7 (Guest OS6). **01** = Read Only Access for VM7 (Guest OS6). **10** = Write Only Access for VM7 (Guest OS6). **11** = FULL access for VM7 (Guest OS6).
VM6 (Guest OS5 Access Permission)	11:10	FULL	This field specifies the access permission for VM6 (Guest OS5). **00** = No Access for VM6 (Guest OS5). **01** = Read Only Access for VM6 (Guest OS5). **10** = Write Only Access for VM6 (Guest OS5). **11** = FULL access for VM6 (Guest OS5).
VM5 (Guest OS4 Access Permission)	9:8	FULL	This field specifies the access permission for VM5 (Guest OS4). **00** = No Access for VM5 (Guest OS4). **01** = Read Only Access for VM5 (Guest OS4). **10** = Write Only Access for VM5 (Guest OS4). **11** = FULL access for VM5 (Guest OS4).
VM4 (Guest OS3 Access Permission)	7:6	FULL	This field specifies the access permission for VM4 (Guest OS3). **00** = No Access for VM4 (Guest OS3). **01** = Read Only Access for VM4 (Guest OS3). **10** = Write Only Access for VM4 (Guest OS3). **11** = FULL access for VM4 (Guest OS3).
VM3 (Guest OS2 Access Permission)	5:4	FULL	This field specifies the access permission for VM3 (Guest OS2). **00** = No Access for VM3 (Guest OS2). **01** = Read Only Access for VM3 (Guest OS2). **10** = Write Only Access for VM3 (Guest OS2). **11** = FULL access for VM3 (Guest OS2).

Field	Bit	Reset	Description
VM2 (Guest OS1 Access Permission)	3:2	FULL	This field specifies the access permission for VM2 (Guest OS1). **00** = No Access for VM2 (Guest OS1). **01** = Read Only Access for VM2 (Guest OS1). **10** = Write Only Access for VM2 (Guest OS1). **11** = FULL access for VM2 (Guest OS1).
VM1 (Guest OS0 Access Permission)	1:0	FULL	This field specifies the access permission for VM1 (Guest OS0). **00** = No Access for VM1 (Guest OS0). **01** = Read Only Access for VM1 (Guest OS0). **10** = Write Only Access for VM1 (Guest OS0). **11** = FULL access for VM1 (Guest OS0).

GPIO_CTL<*kk*>_INTERRUPT_STATUS_VM_0,

where <*kk*> = 0, 1, 2, 3, 4, 5.

This register, available for each Non-AON GPIO Controller (<*kk*>), indicates the Interrupt Status from each of the VM (Guest OS).

Field	Bit	Reset	Description
GPIO_INTERRUPT_STATUS_VM	7:0	0	Each bit of this field Indicates the Interrupt Status whether any GPIO owned by a specific Guest OS resulted an Interrupt. Bit 0 is for VM1 (Guest OS0), Bit 1 is for VM2 (Guest OS1), Bit 2 is for VM3 (Guest OS2), Bit 3 is for VM4 (Guest OS3), Bit 4 is for VM5 (Guest OS4), Bit 5 is for VM6 (Guest OS5), Bit 6 is for VM7 (Guest OS6), Bit 7 is for VM8 (Guest OS7). **0** = Interrupt not set. **1** = Interrupt set. The bit in this field is cleared based on GPIO_<*i*>/<*iii*>_INTERRUPT_CLEAR_0<*j*>_0.GPIO_INTERRUPT_CLEAR.

GPIO_<*k*>/FSI<*m*>_ ICG_EN_OVERRIDE_0,

where <*k*> = CTL0, CTL1, CTL2, CTL3, CTL4, CTL5, AON, and

<*iii*> = 0, 1.

This register, available for each GPIO Controller (<*k*> & <*m*>), is the ICG (Internal Clock Gate) override enable configuration register for the GPIO Controller.

Field	Bit	Reset	Description
ICG_EN	0	DISABLE	This field enables ICG Override. **0** = ICG Override Disabled. **1** = ICG Override Enabled.

GPIO_<*i*>/<*iii*>_INT<*jjj*>_ROUTE_MAPPING_0,

where <*i*> = A, B, C, D, E, F, G, H, I, J, K, L, M, N, P, Q, R, X, Y, Z, AA, AC, AD, AE, AF, AG, BB, CC, DD, EE, GG,

<*iii*> = S, T, W, U, V, and

<*jjj*> = 0, 1, 2, 3, 4, 5, 6, 7. (VM / Guest OS)

This register, available for each GPIO Port (<*i*> & <*iii*>) and VM (<*jjj*>), indicates whether a GPIO Pin in the GPIO Port has caused the assertion of the given Interrupt.

Field	Bit	Default	Description
VAL	[<*j*>*:0]	0	This field has as many bits as the number of GPIO Pins in the given GPIO Port, with each such bit indicating whether the corresponding GPIO Pin has caused the assertion of the given Interrupt (<*jjj*>). **0** = No. **1** = Yes.

*: <*j*> = (*p* -1), where *p* is the number of GPIO Pins in a given GPIO Port <*i*>. And <*j*> ∈ {0, 1, 2, 3, 4, 5, 6, 7}.

8.5.4 Programming Model

The section discusses the programming model of the GPIO Controllers excluding any Security-related topics.

8.5.4.1 GPIO Pad Control Initialization

A GPIO Controller manages a few variants of MPIO/LSIO pads. This means that one such MPIO/LSIO pad can function as I^2C, UART, SPI, or GPIO. Every physical pin/pad has its control registers to fine tune the electrical characteristics and turn on the input receiver circuits, and etc. Refer to the PinMux chapter in this TRM for details on the initialization of Pad and PinMux controls for GPIO mode.

The discussions regarding GPIO input/Output/Bi-directional operations in the following sub-sections are illustrated for GPIO_CTL0 and based on the premise that proper initialization of registers in the PADCTL address space are already taken care of as needed.

8.5.4.2 GPIO Input Configuration

1. Enabling the Clocks for the specific GPIO Controller.

(CLK_RST_CONTROLLER_RST_DEV_GPIO_CTL0_0.SWR_GPIO_CTL0_RST = 0)

(CLK_RST_CONTROLLER_CLK_OUT_ENB_GPIO_CTL0_0.CLK_ENB_GPIO_CTL0 = 1)

NOTE:

Clocks are enabled by default during cold boot and few GPIOs are enabled by Boot Loader.
However, this step may be required by the run-time driver as Clocks may have been disabled as result of run-time power saving.

2. Configuring the GPIO input path.

- (GPIO_<*i*>_ENABLE_CONFIG_0<*j*>_0.IN_OUT = IN)
- Configuring GPIO_<*i*>_ENABLE_CONFIG_0<*j*>_0.TIMESTAMPING_FUNCTION/ INTERRUPT_FUNCTION per use case requirements.
- Configuring GPIO_<*i*>_ENABLE_CONFIG_0<*j*>_0.TRIGGER_LEVEL/TRIGGER_TYPE/ DEBOUNCE_FUNCTION per use case requirements.

3. Setting GPIO_<*i*>_DEBOUNCE_THRESHOLD_0<*j*>_0 with appropriate value based on the debounce requirements of the platform.

4. Ensuring that the AON TSC is configured for generating the msec pulse.

5. Setting (GPIO_<i>_ENABLE_CONFIG_0<j>.GPIO_ENABLE = ENABLE) to complete the GPIO configuration and enable GPIO.

6. Starting Polling/Reading GPIO_<*i*>_INPUT_0<*j*>.GPIO_IN to read the GPIO input value.

NOTE:

When Interrupt is enabled, Interrupt is triggered based on GPIO input processing selection.

7. Optional step:

- Polling GPIO_<*i*>_INTERRUPT_STATUS_G<*jj*>_0 for a specific Security Master (<*jj*>) when proper Interrupt triggers are configured.

8.5.4.3 GPIO Output/Bi-directional Configuration

1. Enabling e the Clocks for the specific GPIO Controller.

(CLK_RST_CONTROLLER_RST_DEV_GPIO_CTL0_0.SWR_GPIO_CTL0_RST = 0)

(CLK_RST_CONTROLLER_CLK_OUT_ENB_GPIO_CTL0_0.CLK_ENB_GPIO_CTL0 = 1)

NOTE:

Clocks are enabled by default during cold boot and few GPIOs are enabled by Boot Loader.
However, this step may be required by the run-time driver as Clocks may have been disabled as result of run-time power saving.

2. Configuring the GPIO direction.

- (GPIO_<*i*>_ENABLE_CONFIG_0<*j*>_0.IN_OUT = OUT)

NOTE:

OUT configuration does not block the input path. Hence, the GPIO so configured also serves as bi-directional.

3. Configuring the GPIO input path (for bi-directional).

- (GPIO_<*i*>_ENABLE_CONFIG_0<*j*>_0.IN_OUT = IN)
- Configuring GPIO_<*i*>_ENABLE_CONFIG_0<*j*>_0.TIMESTAMPING_FUNCTION/ INTERRUPT_FUNCTION per use case requirements.
- Configuring GPIO_<*i*>_ENABLE_CONFIG_0<*j*>_0.TRIGGER_LEVEL/TRIGGER_TYPE/ DEBOUNCE_FUNCTION per use case requirements.

4. Setting GPIO_<*i*>_DEBOUNCE_THRESHOLD_0<*j*>_0 with appropriate value based on the debounce requirements of the platform.

5. Ensuring that the AON TSC is configured for generating the msec pulse. (Not required for GPIO output mode).

6. Setting value to be driven out.

- (GPIO_<*i*>_OUTPUT_VALUE_0<*j*>_0.GPIO_OUT_VALUE = value)

7. Configuring GPO Pin to DRIVE state to drive the value actively.

- (GPIO_<*i*>_OUTPUT_CONTROL_0<*j*>_0.GPIO_OUT_CONTROL = DRIVEN)

NOTE:

To float the bus, (GPIO_<*i*>_OUTPUT_CONTROL_0<*j*>_0.GPIO_OUT_CONTROL = FLOATED).

8. Setting for Bi-directional.

- (GPIO_<*i*>_ENABLE_CONFIG_0<*j*>_0.GPIO_ENABLE = ENABLE)

9. Toggling GPIO.

- (GPIO_<*i*>_OUTPUT_VALUE_0<*j*>_0.GPIO_OUT_VALUE = updated value)

10. Starting Polling/Reading GPIO_<*i*>_INPUT_0<*j*>.GPIO_IN to read the GPIO input value.

NOTE:

This is applicable only for Bi-directional mode.
When Interrupt is enabled, Interrupt is triggered based on GPIO input processing selection.

11. Optional step:

- Polling GPIO_<*i*>_INTERRUPT_STATUS_G<*jj*>_0 for a specific Security Master (<*jj*>) when proper Interrupt triggers are configured.

8.5.4.4 GPIO Interrupts

8.5.4.4.1 Enabling Interrupts for GPIO

- Enabling the basic Interrupt functionality for a specific GPIO.
  - (GPIO_<*i*>_ENABLE_CONFIG_0<*j*>.INTERRUPT_FUNCTION = ENABLE)

8.5.4.4.2 Interrupt Routing to Multiple VMs

Each GPIO Controller has eight physical Interrupts connected to the LIC. The LIC has the capability to route the individual Interrupts to any security domain or software entities via the its CPU's external Interrupts or via the advanced vGIC (Virtualization capable Generic Interrupt Controller).

Based on the platform usage and use case requirements, software is required to identify the mapping of GPIO Pins to a Guest OS. For each of such Pins, Interrupt from the GPIO Controller managing that pin/port needs to be enabled for the Guest OS (all referred as VM) by

- Configuring the specific bit position in the given GPIO_<*i*>_INT<*jjj*>_ROUTE_MAPPING_0 register to route the Interrupt event to one of the VMs.
- Enabling the corresponding GPIO Controller in the LIC to redirect to the specific Security Master.

This ROUTE_MAPPING register, available for each GPIO Port by every GPIO Controller, is part of the common address space, so Hypervisor can configure it in a Virtualized environment.

The ROUTE_MAPPING register is typically programmed during the Boot phase once for all the GPIO Controllers. As part of the SC7 recovery, it is programmed during SC7 exit for all GPIO Controllers excepting the AON GPIO Controller. During SC7 exit, the AON registers are not reprogrammed. When different values are programmed in this register in a toggle fashion, there may be glitches on the Interrupt depending on whether the GPIO is toggled during at the same time.

For details of the LIC, refer to the Interrupt Controllers chapter in this TRM.

GPIO Interrupts and Virtualization

In a system that supports virtualization, it is desirable to have a separate Interrupt input to the vGIC for each VM that controls one or more GPIOs. When separate GPIO Interrupts go to the vGIC for each VM, the Hypervisor can rely on the Interrupt information from the vGIC to schedule the correct VM. Based on the “Interrupt Routing Bitmap” maintained by the Hypervisor, the Hypervisor maps all the GPIO's belonging to a VM to one of the 8 available Interrupt lines, then routes this Interrupt line to the VM via vGIC. Here is the flow:

Hypervisor receives the Interrupt from the GPIO Controller

1. Reads the Interrupt Status Register in vGIC.
2. Programs the vGIC to generate a virtual Interrupt for the Guest OS.
3. Schedules the Guest OS as needed.

Guest OS

1. Reads the Interrupt Status in vGIC to determine the source of the Interrupt.
2. Reads the GPIO Controller to determine which GPIO Pin caused the Interrupt.
3. Calls the appropriate ISR.

There are 8 physical Interrupts from every GPIO Controller with each such Interrupt directly assigned to the Guest. There is no need for Hypervisor to perform any sort of GPIO handling except for setting up the protection. However, when more Interrupts from a GPIO Controller are mapped to specific Security Masters like SCE/BPMP, and VMs are left with fewer Interrupts more than the number of VM's, the Hypervisor’s ISR needs to read the following registers

- GPIO_CTL<*kk*>_INTERRUPT_STATUS_VM_0.
- GPIO_<*i*>/<*iii*>_INTERRUPT_STATUS_G<*jj*>_0.

then based on the read-outs, inject the virtual IRQ to the Guest that needs attention.

For details of the vGIC, refer to the vGIC chapter in this TRM.

8.5.4.5 Enabling Security Attributes for GPIO Registers

Please ask your NVIDIA representative for details.

8.5.4.6 GPIOs in Loopback Mode

When the GPIO output is actively driven, software can immediately poll, i.e.

1. Configure the GPIO Pin as Bi-directional.
1. Drive the output value to "0" or "1".
1. Read the corresponding GPIO IN register.

When the GPIO output is weak-pullup or weak-pull down, software cannot immediately read back the GPIO input pin state. This is because the GPIO input value propagation incurs an RC delay (charge/discharge), and it takes time to reflect the GPIO output value propagated as input value in GPIO IN register. This means that software must introduce a delay before reading the GPIO IN register. To be on the safe side, we consider worst values like R = 100 KOhm and C = 10 pF resulting in an RC delay of ~1 µs. The typical wait time needs to be 5 RC delays, hence 5 µs. Further, to accommodate all the PVT (Process, Voltage, Temperature) corner cases, software is recommended to impose a 20 µs delay in such loopback scenarios. For loopbacks to different GPIOs at the board level and considering trace delays, additional margins may need to be added to the typical delay. The proper steps to read the GPIO input is thus:

- Configure the Pin as GPIO Bi-directional.
- Set Driving output control selection to FLOATED.
- (Internal pullup or board-level pullup as per platform requirements and configuration).
- Wait for 20 µs.
- Read the corresponding GPIO IN register.

8.5.4.7 Disabling Specific GPIO

When a GPIO Pin is removed from the GPIO group the following sequences need to be performed. The step to float the GPIO is explicitly added as part of the GPIO disabling sequence for full backward compatibility of hardware behavior.

- (GPIO_<*i*>_OUTPUT_CONTROL_0<*j*>_0.GPIO_OUT_CONTROL = FLOATED).
- (GPIO_<*i*>_ENABLE_CONFIG_0<*j*>.GPIO_ENABLE = DISABLE).

8.5.4.8 GPIO Programming Entities

All the registers common to a GPIO Controller are programmed by the Boot Loader in the normal system environment, and by the Hypervisor in the Virtualized system environment. Some of the registers like top-level security attribute are programmed by the Boot Loader in all the systems.

8.5.4.9 Interrupt Handling Options and Guidelines

Few additional aspects with respect to GPIO Interrupt handling with Virtualization and Security protection:

- The GPIO Pin to Interrupt associated with a CPU is decoupled from the CPU that owns the configuration control of GPIO Pin. This enables a wide variety of usage models, like some CPUs behaving only as owner for Interrupts but not for configuration, etc.
- No device (including an Interrupt controller) can be modified by more than one software Stack. Any other software stack that wants to use the device must send a message to the holding software stack to act on its behalf.

8.5.5 Platform Aspects

8.5.5.1 De-bounce Filter Threshold Selection

The GPIO Controller implements hardware filter to filter out transient signals possibly caused by GPIO used for mechanical switches and contacts. When the switch is changed, or key is pressed on a Keyboard device, multiple noisy signals are generated. For digital logic to properly interpret a signal or level change, such noisy signals need to be filtered to be a clean logic (0/1) signal. Otherwise, misinterpretation of such unfiltered signal may lead to wrong trigger notification decision for the application.

The bouncy signal duration depends on the mechanical device and type of switch. System software is required to incorporate the platform designer's assistance to specify the de-bounce filtering requirements including level/pulse trigger (signal low to high or vice versa). At times, there may be a filter on the board, or a filtered output is readily available from the switching/KB or similar device. The filter threshold depends solely on the extent of the duration required to mask out the transient state of the device's output.

8.5.5.2 GPIO Usage Restrictions in Platforms

This section summarizes the limitations that arise in the way of assigning GPIOs to different Guest OS's and Security Masters in the Platform usage scenario. The limitation arises from the fact that we have only 8 Interrupts from a GPIO Controller managing 30 to 40 GPIO Pins.

To ensure that the GPIO Pin assignments do not result in limitations such as short of Interrupts, the platform designers working on the GPIO Pin assignment or any software entity instrumenting the GPIO Pin assignment must make sure that either statically (during system Boot) or dynamically (during run-time assignment) no more than 8 Interrupt sources are assigned to one GPIO Controller. Exceeding the limitation requires the handling software to be enhanced.

An option to work around the physical Interrupt limitation is to have the GPIO Interrupts daisy-chained from one Security Master (software stack) to another provided the latency incurred is acceptable.

8.5.5.2.1 Bit Banging Use Cases

Bit banging is about using the software-driven GPIO's to generate arbitrary waveforms, e.g. PWM, I^2C signal, or any similar proprietary bus protocol waveforms. It is possible for a platform designer to use GPIOs for such purposes when in lack of real I^2C or PWM port for instance. Of course, there is a limitation on the toggle rate of the GPIO. This is because the software-managed GPIO toggle rate is governed by the GPIO controller register access time that is about 400 ns to 500 ns (round trip latency based on the assumption that cbb clock = APB clock = CPU clock = 408 MHz).

8.5.6 GPIO Control Registers

NOTE:

All Non-AON, i.e. GPIO_<*ii*>, Control Registers share the same base address under the name GPIO_CTL,

where <*ii*> = A, B, C, D, E, F, G, H, I, J, K, L, M, N, P, Q, R, X, Y, Z, AC, AD, AE, AF, and AG.

All AON, i.e. GPIO_<*iii*>, Control Registers share the same base address under the name AON_GPIO_0,

where <*iii*> = AA, BB, CC, DD, EE, and GG.

For the base address of Non-AON GPIO Control Registers (GPIO_CTL) and AON GPIO Control Registers (AON_GPIO_0), please refer to the System Address Map in this TRM.

NOTE:

The binary “x” is turned to “0” for the Reset and PROD hex value. Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as “x” or “Undefined” actually means that the bit could be either “0” or “1” after Reset, whereas a bit PROD value as “x” or “Don’t-care” means that either “0” or “1” can be written to the bit by software Initialization.

GPIO_<*i*>_ENABLE_CONFIG_0<*j*>_0,

where <*i*> = A, B, C, D, E, F, G, H, I, J, K, L, M, N, P, Q, R, X, Y, Z, AA, AC, AD, AE, AF, AG, BB, CC, DD, EE, GG and

<*j*> ∈ {0, 1, 2, 3, 4, 5, 6, 7}.

GPIO_A_ENABLE_CONFIG_00_0
Offset: 0x10000

GPIO_A_ENABLE_CONFIG_01_0
Offset: 0x10020

GPIO_A_ENABLE_CONFIG_02_0
Offset: 0x10040

GPIO_A_ENABLE_CONFIG_03_0
Offset: 0x10060

GPIO_A_ENABLE_CONFIG_04_0
Offset: 0x10080

GPIO_A_ENABLE_CONFIG_05_0
Offset: 0x100a0

GPIO_A_ENABLE_CONFIG_06_0
Offset: 0x100c0

GPIO_A_ENABLE_CONFIG_07_0
Offset: 0x100e0

GPIO_AC_ENABLE_CONFIG_00_0
Offset: 0x10200

GPIO_AC_ENABLE_CONFIG_01_0
Offset: 0x10220

GPIO_AC_ENABLE_CONFIG_02_0
Offset: 0x10240

GPIO_AC_ENABLE_CONFIG_03_0
Offset: 0x10260

GPIO_AC_ENABLE_CONFIG_04_0
Offset: 0x10280

GPIO_AC_ENABLE_CONFIG_05_0
Offset: 0x102a0

GPIO_AC_ENABLE_CONFIG_06_0
Offset: 0x102c0

GPIO_AC_ENABLE_CONFIG_07_0
Offset: 0x102e0

GPIO_AD_ENABLE_CONFIG_00_0
Offset: 0x10400

GPIO_AD_ENABLE_CONFIG_01_0
Offset: 0x10420

GPIO_AD_ENABLE_CONFIG_02_0
Offset: 0x10440

GPIO_AD_ENABLE_CONFIG_03_0
Offset: 0x10460

GPIO_B_ENABLE_CONFIG_00_0
Offset: 0x10600

GPIO_X_ENABLE_CONFIG_00_0
Offset: 0x11000

GPIO_X_ENABLE_CONFIG_01_0
Offset: 0x11020

GPIO_X_ENABLE_CONFIG_02_0
Offset: 0x11040

GPIO_X_ENABLE_CONFIG_03_0
Offset: 0x11060

GPIO_X_ENABLE_CONFIG_04_0
Offset: 0x11080

GPIO_X_ENABLE_CONFIG_05_0
Offset: 0x110a0

GPIO_X_ENABLE_CONFIG_06_0
Offset: 0x110c0

GPIO_X_ENABLE_CONFIG_07_0
Offset: 0x110e0

GPIO_Y_ENABLE_CONFIG_00_0
Offset: 0x11200

GPIO_Y_ENABLE_CONFIG_01_0
Offset: 0x11220

GPIO_Y_ENABLE_CONFIG_02_0
Offset: 0x11240

GPIO_Y_ENABLE_CONFIG_03_0
Offset: 0x11260

GPIO_Y_ENABLE_CONFIG_04_0
Offset: 0x11280

GPIO_Y_ENABLE_CONFIG_05_0
Offset: 0x112a0

GPIO_Y_ENABLE_CONFIG_06_0
Offset: 0x112c0

GPIO_Y_ENABLE_CONFIG_07_0
Offset: 0x112e0

GPIO_Z_ENABLE_CONFIG_00_0
Offset: 0x11400

GPIO_Z_ENABLE_CONFIG_01_0
Offset: 0x11420

GPIO_Z_ENABLE_CONFIG_02_0
Offset: 0x11440

GPIO_Z_ENABLE_CONFIG_03_0
Offset: 0x11460

GPIO_Z_ENABLE_CONFIG_04_0
Offset: 0x11480

GPIO_Z_ENABLE_CONFIG_05_0
Offset: 0x114a0

GPIO_Z_ENABLE_CONFIG_06_0
Offset: 0x114c0

GPIO_Z_ENABLE_CONFIG_07_0
Offset: 0x114e0

GPIO_M_ENABLE_CONFIG_00_0
Offset: 0x12000

GPIO_M_ENABLE_CONFIG_01_0
Offset: 0x12020

GPIO_M_ENABLE_CONFIG_02_0
Offset: 0x12040

GPIO_M_ENABLE_CONFIG_03_0
Offset: 0x12060

GPIO_M_ENABLE_CONFIG_04_0
Offset: 0x12080

GPIO_M_ENABLE_CONFIG_05_0
Offset: 0x120a0

GPIO_M_ENABLE_CONFIG_06_0
Offset: 0x120c0

GPIO_M_ENABLE_CONFIG_07_0
Offset: 0x120e0

GPIO_N_ENABLE_CONFIG_00_0
Offset: 0x12200

GPIO_N_ENABLE_CONFIG_01_0
Offset: 0x12220

GPIO_N_ENABLE_CONFIG_02_0
Offset: 0x12240

GPIO_N_ENABLE_CONFIG_03_0
Offset: 0x12260

GPIO_N_ENABLE_CONFIG_04_0
Offset: 0x12280

GPIO_N_ENABLE_CONFIG_05_0
Offset: 0x122a0

GPIO_N_ENABLE_CONFIG_06_0
Offset: 0x122c0

GPIO_N_ENABLE_CONFIG_07_0
Offset: 0x122e0

GPIO_P_ENABLE_CONFIG_00_0
Offset: 0x12400

GPIO_P_ENABLE_CONFIG_01_0
Offset: 0x12420

GPIO_P_ENABLE_CONFIG_02_0
Offset: 0x12440

GPIO_P_ENABLE_CONFIG_03_0
Offset: 0x12460

GPIO_P_ENABLE_CONFIG_04_0
Offset: 0x12480

GPIO_P_ENABLE_CONFIG_05_0
Offset: 0x124a0

GPIO_P_ENABLE_CONFIG_06_0
Offset: 0x124c0

GPIO_P_ENABLE_CONFIG_07_0
Offset: 0x124e0

GPIO_Q_ENABLE_CONFIG_00_0
Offset: 0x12600

GPIO_Q_ENABLE_CONFIG_01_0
Offset: 0x12620

GPIO_Q_ENABLE_CONFIG_02_0
Offset: 0x12640

GPIO_Q_ENABLE_CONFIG_03_0
Offset: 0x12660

GPIO_Q_ENABLE_CONFIG_04_0
Offset: 0x12680

GPIO_Q_ENABLE_CONFIG_05_0
Offset: 0x126a0

GPIO_Q_ENABLE_CONFIG_06_0
Offset: 0x126c0

GPIO_Q_ENABLE_CONFIG_07_0
Offset: 0x126e0

GPIO_R_ENABLE_CONFIG_00_0
Offset: 0x12800

GPIO_R_ENABLE_CONFIG_01_0
Offset: 0x12820

GPIO_R_ENABLE_CONFIG_02_0
Offset: 0x12840

GPIO_R_ENABLE_CONFIG_03_0
Offset: 0x12860

GPIO_R_ENABLE_CONFIG_04_0
Offset: 0x12880

GPIO_R_ENABLE_CONFIG_05_0
Offset: 0x128a0

GPIO_K_ENABLE_CONFIG_00_0
Offset: 0x13000

GPIO_K_ENABLE_CONFIG_01_0
Offset: 0x13020

GPIO_K_ENABLE_CONFIG_02_0
Offset: 0x13040

GPIO_K_ENABLE_CONFIG_03_0
Offset: 0x13060

GPIO_K_ENABLE_CONFIG_04_0
Offset: 0x13080

GPIO_K_ENABLE_CONFIG_05_0
Offset: 0x130a0

GPIO_K_ENABLE_CONFIG_06_0
Offset: 0x130c0

GPIO_K_ENABLE_CONFIG_07_0
Offset: 0x130e0

GPIO_L_ENABLE_CONFIG_00_0
Offset: 0x13200

GPIO_L_ENABLE_CONFIG_01_0
Offset: 0x13220

GPIO_L_ENABLE_CONFIG_02_0
Offset: 0x13240

GPIO_L_ENABLE_CONFIG_03_0
Offset: 0x13260

GPIO_AG_ENABLE_CONFIG_00_0
Offset: 0x13400

GPIO_AG_ENABLE_CONFIG_01_0
Offset: 0x13420

GPIO_AG_ENABLE_CONFIG_02_0
Offset: 0x13440

GPIO_AG_ENABLE_CONFIG_03_0
Offset: 0x13460

GPIO_AG_ENABLE_CONFIG_04_0
Offset: 0x13480

GPIO_AG_ENABLE_CONFIG_05_0
Offset: 0x134a0

GPIO_AG_ENABLE_CONFIG_06_0
Offset: 0x134c0

GPIO_AG_ENABLE_CONFIG_07_0
Offset: 0x134e0

GPIO_AE_ENABLE_CONFIG_00_0
Offset: 0x13600

GPIO_AE_ENABLE_CONFIG_01_0
Offset: 0x13620

GPIO_AF_ENABLE_CONFIG_00_0
Offset: 0x13800

GPIO_AF_ENABLE_CONFIG_01_0
Offset: 0x13820

GPIO_AF_ENABLE_CONFIG_02_0
Offset: 0x13840

GPIO_AF_ENABLE_CONFIG_03_0
Offset: 0x13860

GPIO_G_ENABLE_CONFIG_00_0
Offset: 0x14000

GPIO_G_ENABLE_CONFIG_01_0
Offset: 0x14020

GPIO_G_ENABLE_CONFIG_02_0
Offset: 0x14040

GPIO_G_ENABLE_CONFIG_03_0
Offset: 0x14060

GPIO_G_ENABLE_CONFIG_04_0
Offset: 0x14080

GPIO_G_ENABLE_CONFIG_05_0
Offset: 0x140a0

GPIO_G_ENABLE_CONFIG_06_0
Offset: 0x140c0

GPIO_G_ENABLE_CONFIG_07_0
Offset: 0x140e0

GPIO_H_ENABLE_CONFIG_00_0
Offset: 0x14200

GPIO_H_ENABLE_CONFIG_01_0
Offset: 0x14220

GPIO_H_ENABLE_CONFIG_02_0
Offset: 0x14240

GPIO_H_ENABLE_CONFIG_03_0
Offset: 0x14260

GPIO_H_ENABLE_CONFIG_04_0
Offset: 0x14280

GPIO_H_ENABLE_CONFIG_05_0
Offset: 0x142a0

GPIO_H_ENABLE_CONFIG_06_0
Offset: 0x142c0

GPIO_H_ENABLE_CONFIG_07_0
Offset: 0x142e0

GPIO_I_ENABLE_CONFIG_00_0
Offset: 0x14400

GPIO_I_ENABLE_CONFIG_01_0
Offset: 0x14420

GPIO_I_ENABLE_CONFIG_02_0
Offset: 0x14440

GPIO_I_ENABLE_CONFIG_03_0
Offset: 0x14460

GPIO_I_ENABLE_CONFIG_04_0
Offset: 0x14480

GPIO_I_ENABLE_CONFIG_05_0
Offset: 0x144a0

GPIO_I_ENABLE_CONFIG_06_0
Offset: 0x144c0

GPIO_J_ENABLE_CONFIG_00_0
Offset: 0x15000

GPIO_J_ENABLE_CONFIG_01_0
Offset: 0x15020

GPIO_J_ENABLE_CONFIG_02_0
Offset: 0x15040

GPIO_J_ENABLE_CONFIG_03_0
Offset: 0x15060

GPIO_J_ENABLE_CONFIG_04_0
Offset: 0x15080

GPIO_J_ENABLE_CONFIG_05_0
Offset: 0x150a0

GPIO_C_ENABLE_CONFIG_00_0
Offset: 0x15200

GPIO_C_ENABLE_CONFIG_01_0
Offset: 0x15220

GPIO_C_ENABLE_CONFIG_02_0
Offset: 0x15240

GPIO_C_ENABLE_CONFIG_03_0
Offset: 0x15260

GPIO_C_ENABLE_CONFIG_04_0
Offset: 0x15280

GPIO_C_ENABLE_CONFIG_05_0
Offset: 0x152a0

GPIO_C_ENABLE_CONFIG_06_0
Offset: 0x152c0

GPIO_C_ENABLE_CONFIG_07_0
Offset: 0x152e0

GPIO_D_ENABLE_CONFIG_00_0
Offset: 0x15400

GPIO_D_ENABLE_CONFIG_01_0
Offset: 0x15420

GPIO_D_ENABLE_CONFIG_02_0
Offset: 0x15440

GPIO_D_ENABLE_CONFIG_03_0
Offset: 0x15460

GPIO_E_ENABLE_CONFIG_00_0
Offset: 0x15600

GPIO_E_ENABLE_CONFIG_01_0
Offset: 0x15620

GPIO_E_ENABLE_CONFIG_02_0
Offset: 0x15640

GPIO_E_ENABLE_CONFIG_03_0
Offset: 0x15660

GPIO_E_ENABLE_CONFIG_04_0
Offset: 0x15680

GPIO_E_ENABLE_CONFIG_05_0
Offset: 0x156a0

GPIO_E_ENABLE_CONFIG_06_0
Offset: 0x156c0

GPIO_E_ENABLE_CONFIG_07_0
Offset: 0x156e0

GPIO_F_ENABLE_CONFIG_00_0
Offset: 0x15800

GPIO_F_ENABLE_CONFIG_01_0
Offset: 0x15820

GPIO_F_ENABLE_CONFIG_02_0
Offset: 0x15840

GPIO_F_ENABLE_CONFIG_03_0
Offset: 0x15860

GPIO_F_ENABLE_CONFIG_04_0
Offset: 0x15880

GPIO_F_ENABLE_CONFIG_05_0
Offset: 0x158a0

GPIO_EE_ENABLE_CONFIG_00_0
Offset: 0x1000

GPIO_EE_ENABLE_CONFIG_01_0
Offset: 0x1020

GPIO_EE_ENABLE_CONFIG_02_0
Offset: 0x1040

GPIO_EE_ENABLE_CONFIG_03_0
Offset: 0x1060

GPIO_EE_ENABLE_CONFIG_04_0
Offset: 0x1080

GPIO_EE_ENABLE_CONFIG_05_0
Offset: 0x10a0

GPIO_EE_ENABLE_CONFIG_06_0
Offset: 0x10c0

GPIO_EE_ENABLE_CONFIG_07_0
Offset: 0x10e0

GPIO_GG_ENABLE_CONFIG_00_0
Offset: 0x1200

GPIO_CC_ENABLE_CONFIG_00_0
Offset: 0x1400

GPIO_CC_ENABLE_CONFIG_01_0
Offset: 0x1420

GPIO_CC_ENABLE_CONFIG_02_0
Offset: 0x1440

GPIO_CC_ENABLE_CONFIG_03_0
Offset: 0x1460

GPIO_CC_ENABLE_CONFIG_04_0
Offset: 0x1480

GPIO_CC_ENABLE_CONFIG_05_0
Offset: 0x14a0

GPIO_CC_ENABLE_CONFIG_06_0
Offset: 0x14c0

GPIO_CC_ENABLE_CONFIG_07_0
Offset: 0x14e0

GPIO_DD_ENABLE_CONFIG_00_0
Offset: 0x1600

GPIO_DD_ENABLE_CONFIG_01_0
Offset: 0x1620

GPIO_DD_ENABLE_CONFIG_02_0
Offset: 0x1640

GPIO_AA_ENABLE_CONFIG_00_0
Offset: 0x1800

GPIO_AA_ENABLE_CONFIG_01_0
Offset: 0x1820

GPIO_AA_ENABLE_CONFIG_02_0
Offset: 0x1840

GPIO_AA_ENABLE_CONFIG_03_0
Offset: 0x1860

GPIO_AA_ENABLE_CONFIG_04_0
Offset: 0x1880

GPIO_AA_ENABLE_CONFIG_05_0
Offset: 0x18a0

GPIO_AA_ENABLE_CONFIG_06_0
Offset: 0x18c0

GPIO_AA_ENABLE_CONFIG_07_0
Offset: 0x18e0

GPIO_BB_ENABLE_CONFIG_00_0
Offset: 0x1a00

GPIO_BB_ENABLE_CONFIG_01_0
Offset: 0x1a20

GPIO_BB_ENABLE_CONFIG_02_0
Offset: 0x1a40

GPIO_BB_ENABLE_CONFIG_03_0
Offset: 0x1a60

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GPIO_<*i*>_SCR_0<*j*>_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7	0x0	**TIMESTAMPING_FUNCTION**: This field allows the GPIO input change event to create a timestamp event. When enabled based on the settings in TRIGGER_TYPE & TRIGGER LEVEL time-stamp trigger is sent out to TSC. It is effective only when (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE) && (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.IN_OUT == IN). **0** = DISABLE. **1** = ENABLE.
6	0x0	**INTERRUPT_FUNCTION**: This field provides the Interrupt selection for the GPIO Input changes based on trigger type (TRIGGER_TYPE) trigger polarity (TRIGGER LEVEL) It is effective only when (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE). **0** = DISABLE. **1** = ENABLE. When ((GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE) && (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.INTERRUPT_FUNCTION == 1)), the Interrupt Status is updated in the Interrupt Status Register (GPIO_<*i*>_INTERRUPT_STATUS_G<*jj*>_0) for all the VM's, based on the settings of GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.TRIGGER_TYPE & GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.TRIGGER LEVEL.

Bit	Reset	Description
5	0x0	**DEBOUNCE_FUNCTION**: This field provides the Debounce filter selection on the GPIO Input. It is effective only when (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE). **0** = DISABLE (no Debounce filter). **1** = ENABLE. When (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.DEBOUNCE_FUNCTION == ENABLE), the debouncing interval value is defined by (GPIO_<*i*>_DEBOUNCE_THRESHOLD_0<*j*>_0.DEBOUNCE_THRESHOLD).
4	0x0	**TRIGGER_LEVEL**: This field selects the polarity of the GPIO Input trigger for the given trigger type. It is effective only when (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE). When (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.TRIGGER_TYPE == NO TRIGGER), this field is irrelevant. When (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.TRIGGER_TYPE == LEVEL), **0** = Trigger on level Low. **1** = Trigger on Level High. When (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.TRIGGER_TYPE == SINGLE EDGE), **0** = Trigger on Falling Edge. **1** = Trigger on Raising Edge. When (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.TRIGGER_TYPE == BOTH EDGES), this field is irrelevant.
3:2	0x0	**TRIGGER_TYPE**: This field selects the GPIO Input trigger type (condition). It is effective only when (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE). **0** = NO_TRIGGER. **1** = LEVEL. **2** = SINGLE EDGE. **3** = DOUBLE_EDGE (i.e. both Low-High, High-Low edges are detected as event).
1	0x0	**IN_OUT**: This field configures the GPIO for Input or Output. It is effective only when (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE). **0** = IN. **1** = OUT.
0	0x0	**GPIO_ENABLE**: This field is the GPIO functionality selection control required for output/input selection. It acts as a global qualifier for all the functions of the specific GPIO. It is also used for SLCG (Second Level Clock Gating) per pin-based GPIO logic. **0** = DISABLE. **1** = ENABLE. When disabled, the GPIO is floated, i.e. both input and output are disabled. At this point, If PinMux selects the GPIO, the I/O is in High-Z state.

GPIO_<*i*>_DEBOUNCE_THRESHOLD_0<*j*>_0,

where <*i*> = A, B, C, D, E, F, G, H, I, J, K, L, M, N, P, Q, R, X, Y, Z, AA, AC, AD, AE, AF, AG, BB, CC, DD, EE, GG and

<*j*> ∈ {0, 1, 2, 3, 4, 5, 6, 7}.

GPIO_A_DEBOUNCE_THRESHOLD_00_0
Offset: 0x10004

GPIO_A_DEBOUNCE_THRESHOLD_01_0
Offset: 0x10024

GPIO_A_DEBOUNCE_THRESHOLD_02_0
Offset: 0x10044

GPIO_A_DEBOUNCE_THRESHOLD_03_0
Offset: 0x10064

GPIO_A_DEBOUNCE_THRESHOLD_04_0
Offset: 0x10084

GPIO_A_DEBOUNCE_THRESHOLD_05_0
Offset: 0x100a4

GPIO_A_DEBOUNCE_THRESHOLD_06_0
Offset: 0x100c4

GPIO_A_DEBOUNCE_THRESHOLD_07_0
Offset: 0x100e4

GPIO_AC_DEBOUNCE_THRESHOLD_00_0
Offset: 0x10204

GPIO_AC_DEBOUNCE_THRESHOLD_01_0
Offset: 0x10224

GPIO_AC_DEBOUNCE_THRESHOLD_02_0
Offset: 0x10244

GPIO_AC_DEBOUNCE_THRESHOLD_03_0
Offset: 0x10264

GPIO_AC_DEBOUNCE_THRESHOLD_04_0
Offset: 0x10284

GPIO_AC_DEBOUNCE_THRESHOLD_05_0
Offset: 0x102a4

GPIO_AC_DEBOUNCE_THRESHOLD_06_0
Offset: 0x102c4

GPIO_AC_DEBOUNCE_THRESHOLD_07_0
Offset: 0x102e4

GPIO_B_DEBOUNCE_THRESHOLD_00_0
Offset: 0x10604

GPIO_X_DEBOUNCE_THRESHOLD_00_0
Offset: 0x11004

GPIO_X_DEBOUNCE_THRESHOLD_01_0
Offset: 0x11024

GPIO_X_DEBOUNCE_THRESHOLD_02_0
Offset: 0x11044

GPIO_X_DEBOUNCE_THRESHOLD_03_0
Offset: 0x11064

GPIO_X_DEBOUNCE_THRESHOLD_04_0
Offset: 0x11084

GPIO_X_DEBOUNCE_THRESHOLD_05_0
Offset: 0x110a4

GPIO_X_DEBOUNCE_THRESHOLD_06_0
Offset: 0x110c4

GPIO_X_DEBOUNCE_THRESHOLD_07_0
Offset: 0x110e4

GPIO_Y_DEBOUNCE_THRESHOLD_00_0
Offset: 0x11204

GPIO_Y_DEBOUNCE_THRESHOLD_01_0
Offset: 0x11224

GPIO_Y_DEBOUNCE_THRESHOLD_02_0
Offset: 0x11244

GPIO_Y_DEBOUNCE_THRESHOLD_03_0
Offset: 0x11264

GPIO_Y_DEBOUNCE_THRESHOLD_04_0
Offset: 0x11284

GPIO_Y_DEBOUNCE_THRESHOLD_05_0
Offset: 0x112a4

GPIO_Y_DEBOUNCE_THRESHOLD_06_0
Offset: 0x112c4

GPIO_Y_DEBOUNCE_THRESHOLD_07_0
Offset: 0x112e4

GPIO_Z_DEBOUNCE_THRESHOLD_00_0
Offset: 0x11404

GPIO_Z_DEBOUNCE_THRESHOLD_01_0
Offset: 0x11424

GPIO_Z_DEBOUNCE_THRESHOLD_02_0
Offset: 0x11444

GPIO_Z_DEBOUNCE_THRESHOLD_03_0
Offset: 0x11464

GPIO_Z_DEBOUNCE_THRESHOLD_04_0
Offset: 0x11484

GPIO_Z_DEBOUNCE_THRESHOLD_05_0
Offset: 0x114a4

GPIO_Z_DEBOUNCE_THRESHOLD_06_0
Offset: 0x114c4

GPIO_Z_DEBOUNCE_THRESHOLD_07_0
Offset: 0x114e4

GPIO_M_DEBOUNCE_THRESHOLD_00_0
Offset: 0x12004

GPIO_M_DEBOUNCE_THRESHOLD_01_0
Offset: 0x12024

GPIO_M_DEBOUNCE_THRESHOLD_02_0
Offset: 0x12044

GPIO_M_DEBOUNCE_THRESHOLD_03_0
Offset: 0x12064

GPIO_M_DEBOUNCE_THRESHOLD_04_0
Offset: 0x12084

GPIO_M_DEBOUNCE_THRESHOLD_05_0
Offset: 0x120a4

GPIO_M_DEBOUNCE_THRESHOLD_06_0
Offset: 0x120c4

GPIO_M_DEBOUNCE_THRESHOLD_07_0
Offset: 0x120e4

GPIO_N_DEBOUNCE_THRESHOLD_00_0
Offset: 0x12204

GPIO_N_DEBOUNCE_THRESHOLD_01_0
Offset: 0x12224

GPIO_N_DEBOUNCE_THRESHOLD_02_0
Offset: 0x12244

GPIO_N_DEBOUNCE_THRESHOLD_03_0
Offset: 0x12264

GPIO_N_DEBOUNCE_THRESHOLD_04_0
Offset: 0x12284

GPIO_N_DEBOUNCE_THRESHOLD_05_0
Offset: 0x122a4

GPIO_N_DEBOUNCE_THRESHOLD_06_0
Offset: 0x122c4

GPIO_N_DEBOUNCE_THRESHOLD_07_0
Offset: 0x122e4

GPIO_P_DEBOUNCE_THRESHOLD_00_0
Offset: 0x12404

GPIO_P_DEBOUNCE_THRESHOLD_01_0
Offset: 0x12424

GPIO_P_DEBOUNCE_THRESHOLD_02_0
Offset: 0x12444

GPIO_P_DEBOUNCE_THRESHOLD_03_0
Offset: 0x12464

GPIO_P_DEBOUNCE_THRESHOLD_04_0
Offset: 0x12484

GPIO_P_DEBOUNCE_THRESHOLD_05_0
Offset: 0x124a4

GPIO_P_DEBOUNCE_THRESHOLD_06_0
Offset: 0x124c4

GPIO_P_DEBOUNCE_THRESHOLD_07_0
Offset: 0x124e4

GPIO_Q_DEBOUNCE_THRESHOLD_00_0
Offset: 0x12604

GPIO_Q_DEBOUNCE_THRESHOLD_01_0
Offset: 0x12624

GPIO_Q_DEBOUNCE_THRESHOLD_02_0
Offset: 0x12644

GPIO_Q_DEBOUNCE_THRESHOLD_03_0
Offset: 0x12664

GPIO_Q_DEBOUNCE_THRESHOLD_04_0
Offset: 0x12684

GPIO_Q_DEBOUNCE_THRESHOLD_05_0
Offset: 0x126a4

GPIO_Q_DEBOUNCE_THRESHOLD_06_0
Offset: 0x126c4

GPIO_Q_DEBOUNCE_THRESHOLD_07_0
Offset: 0x126e4

GPIO_R_DEBOUNCE_THRESHOLD_00_0
Offset: 0x12804

GPIO_R_DEBOUNCE_THRESHOLD_01_0
Offset: 0x12824

GPIO_R_DEBOUNCE_THRESHOLD_02_0
Offset: 0x12844

GPIO_R_DEBOUNCE_THRESHOLD_03_0
Offset: 0x12864

GPIO_R_DEBOUNCE_THRESHOLD_04_0
Offset: 0x12884

GPIO_R_DEBOUNCE_THRESHOLD_05_0
Offset: 0x128a4

GPIO_K_DEBOUNCE_THRESHOLD_00_0
Offset: 0x13004

GPIO_K_DEBOUNCE_THRESHOLD_01_0
Offset: 0x13024

GPIO_K_DEBOUNCE_THRESHOLD_02_0
Offset: 0x13044

GPIO_K_DEBOUNCE_THRESHOLD_03_0
Offset: 0x13064

GPIO_K_DEBOUNCE_THRESHOLD_04_0
Offset: 0x13084

GPIO_K_DEBOUNCE_THRESHOLD_05_0
Offset: 0x130a4

GPIO_K_DEBOUNCE_THRESHOLD_06_0
Offset: 0x130c4

GPIO_K_DEBOUNCE_THRESHOLD_07_0
Offset: 0x130e4

GPIO_L_DEBOUNCE_THRESHOLD_00_0
Offset: 0x13204

GPIO_L_DEBOUNCE_THRESHOLD_01_0
Offset: 0x13224

GPIO_L_DEBOUNCE_THRESHOLD_02_0
Offset: 0x13244

GPIO_L_DEBOUNCE_THRESHOLD_03_0
Offset: 0x13264

GPIO_AG_DEBOUNCE_THRESHOLD_00_0
Offset: 0x13404

GPIO_AG_DEBOUNCE_THRESHOLD_01_0
Offset: 0x13424

GPIO_AG_DEBOUNCE_THRESHOLD_02_0
Offset: 0x13444

GPIO_AG_DEBOUNCE_THRESHOLD_03_0
Offset: 0x13464

GPIO_AG_DEBOUNCE_THRESHOLD_04_0
Offset: 0x13484

GPIO_AG_DEBOUNCE_THRESHOLD_05_0
Offset: 0x134a4

GPIO_AG_DEBOUNCE_THRESHOLD_06_0
Offset: 0x134c4

GPIO_AG_DEBOUNCE_THRESHOLD_07_0
Offset: 0x134e4

GPIO_AE_DEBOUNCE_THRESHOLD_00_0
Offset: 0x13604

GPIO_AE_DEBOUNCE_THRESHOLD_01_0
Offset: 0x13624

GPIO_AF_DEBOUNCE_THRESHOLD_00_0
Offset: 0x13804

GPIO_AF_DEBOUNCE_THRESHOLD_01_0
Offset: 0x13824

GPIO_AF_DEBOUNCE_THRESHOLD_02_0
Offset: 0x13844

GPIO_AF_DEBOUNCE_THRESHOLD_03_0
Offset: 0x13864

GPIO_G_DEBOUNCE_THRESHOLD_00_0
Offset: 0x14004

GPIO_G_DEBOUNCE_THRESHOLD_01_0
Offset: 0x14024

GPIO_G_DEBOUNCE_THRESHOLD_02_0
Offset: 0x14044

GPIO_G_DEBOUNCE_THRESHOLD_03_0
Offset: 0x14064

GPIO_G_DEBOUNCE_THRESHOLD_04_0
Offset: 0x14084

GPIO_G_DEBOUNCE_THRESHOLD_05_0
Offset: 0x140a4

GPIO_G_DEBOUNCE_THRESHOLD_06_0
Offset: 0x140c4

GPIO_G_DEBOUNCE_THRESHOLD_07_0
Offset: 0x140e4

GPIO_H_DEBOUNCE_THRESHOLD_00_0
Offset: 0x14204

GPIO_H_DEBOUNCE_THRESHOLD_01_0
Offset: 0x14224

GPIO_H_DEBOUNCE_THRESHOLD_02_0
Offset: 0x14244

GPIO_H_DEBOUNCE_THRESHOLD_03_0
Offset: 0x14264

GPIO_H_DEBOUNCE_THRESHOLD_04_0
Offset: 0x14284

GPIO_H_DEBOUNCE_THRESHOLD_05_0
Offset: 0x142a4

GPIO_H_DEBOUNCE_THRESHOLD_06_0
Offset: 0x142c4

GPIO_H_DEBOUNCE_THRESHOLD_07_0
Offset: 0x142e4

GPIO_I_DEBOUNCE_THRESHOLD_00_0
Offset: 0x14404

GPIO_I_DEBOUNCE_THRESHOLD_01_0
Offset: 0x14424

GPIO_I_DEBOUNCE_THRESHOLD_02_0
Offset: 0x14444

GPIO_I_DEBOUNCE_THRESHOLD_03_0
Offset: 0x14464

GPIO_I_DEBOUNCE_THRESHOLD_04_0
Offset: 0x14484

GPIO_I_DEBOUNCE_THRESHOLD_05_0
Offset: 0x144a4

GPIO_I_DEBOUNCE_THRESHOLD_06_0
Offset: 0x144c4

GPIO_J_DEBOUNCE_THRESHOLD_00_0
Offset: 0x15004

GPIO_J_DEBOUNCE_THRESHOLD_01_0
Offset: 0x15024

GPIO_J_DEBOUNCE_THRESHOLD_02_0
Offset: 0x15044

GPIO_J_DEBOUNCE_THRESHOLD_03_0
Offset: 0x15064

GPIO_J_DEBOUNCE_THRESHOLD_04_0
Offset: 0x15084

GPIO_J_DEBOUNCE_THRESHOLD_05_0
Offset: 0x150a4

GPIO_C_DEBOUNCE_THRESHOLD_00_0
Offset: 0x15204

GPIO_C_DEBOUNCE_THRESHOLD_01_0
Offset: 0x15224

GPIO_C_DEBOUNCE_THRESHOLD_02_0
Offset: 0x15244

GPIO_C_DEBOUNCE_THRESHOLD_03_0
Offset: 0x15264

GPIO_C_DEBOUNCE_THRESHOLD_04_0
Offset: 0x15284

GPIO_C_DEBOUNCE_THRESHOLD_05_0
Offset: 0x152a4

GPIO_C_DEBOUNCE_THRESHOLD_06_0
Offset: 0x152c4

GPIO_C_DEBOUNCE_THRESHOLD_07_0
Offset: 0x152e4

GPIO_D_DEBOUNCE_THRESHOLD_00_0
Offset: 0x15404

GPIO_D_DEBOUNCE_THRESHOLD_01_0
Offset: 0x15424

GPIO_D_DEBOUNCE_THRESHOLD_02_0
Offset: 0x15444

GPIO_D_DEBOUNCE_THRESHOLD_03_0
Offset: 0x15464

GPIO_E_DEBOUNCE_THRESHOLD_00_0
Offset: 0x15604

GPIO_E_DEBOUNCE_THRESHOLD_01_0
Offset: 0x15624

GPIO_E_DEBOUNCE_THRESHOLD_02_0
Offset: 0x15644

GPIO_E_DEBOUNCE_THRESHOLD_03_0
Offset: 0x15664

GPIO_E_DEBOUNCE_THRESHOLD_04_0
Offset: 0x15684

GPIO_E_DEBOUNCE_THRESHOLD_05_0
Offset: 0x156a4

GPIO_E_DEBOUNCE_THRESHOLD_06_0
Offset: 0x156c4

GPIO_E_DEBOUNCE_THRESHOLD_07_0
Offset: 0x156e4

GPIO_F_DEBOUNCE_THRESHOLD_00_0
Offset: 0x15804

GPIO_F_DEBOUNCE_THRESHOLD_01_0
Offset: 0x15824

GPIO_F_DEBOUNCE_THRESHOLD_02_0
Offset: 0x15844

GPIO_F_DEBOUNCE_THRESHOLD_03_0
Offset: 0x15864

GPIO_F_DEBOUNCE_THRESHOLD_04_0
Offset: 0x15884

GPIO_F_DEBOUNCE_THRESHOLD_05_0
Offset: 0x158a4

GPIO_EE_DEBOUNCE_THRESHOLD_00_0
Offset: 0x1004

GPIO_EE_DEBOUNCE_THRESHOLD_01_0
Offset: 0x1024

GPIO_EE_DEBOUNCE_THRESHOLD_02_0
Offset: 0x1044

GPIO_EE_DEBOUNCE_THRESHOLD_03_0
Offset: 0x1064

GPIO_EE_DEBOUNCE_THRESHOLD_04_0
Offset: 0x1084

GPIO_EE_DEBOUNCE_THRESHOLD_05_0
Offset: 0x10a4

GPIO_EE_DEBOUNCE_THRESHOLD_06_0
Offset: 0x10c4

GPIO_EE_DEBOUNCE_THRESHOLD_07_0
Offset: 0x10e4

GPIO_GG_DEBOUNCE_THRESHOLD_00_0
Offset: 0x1204

GPIO_CC_DEBOUNCE_THRESHOLD_00_0
Offset: 0x1404

GPIO_CC_DEBOUNCE_THRESHOLD_01_0
Offset: 0x1424

GPIO_CC_DEBOUNCE_THRESHOLD_02_0
Offset: 0x1444

GPIO_CC_DEBOUNCE_THRESHOLD_03_0
Offset: 0x1464

GPIO_CC_DEBOUNCE_THRESHOLD_04_0
Offset: 0x1484

GPIO_CC_DEBOUNCE_THRESHOLD_05_0
Offset: 0x14a4

GPIO_CC_DEBOUNCE_THRESHOLD_06_0
Offset: 0x14c4

GPIO_CC_DEBOUNCE_THRESHOLD_07_0
Offset: 0x14e4

GPIO_DD_DEBOUNCE_THRESHOLD_00_0
Offset: 0x1604

GPIO_DD_DEBOUNCE_THRESHOLD_01_0
Offset: 0x1624

GPIO_DD_DEBOUNCE_THRESHOLD_02_0
Offset: 0x1644

GPIO_AA_DEBOUNCE_THRESHOLD_00_0
Offset: 0x1804

GPIO_AA_DEBOUNCE_THRESHOLD_01_0
Offset: 0x1824

GPIO_AA_DEBOUNCE_THRESHOLD_02_0
Offset: 0x1844

GPIO_AA_DEBOUNCE_THRESHOLD_03_0
Offset: 0x1864

GPIO_AA_DEBOUNCE_THRESHOLD_04_0
Offset: 0x1884

GPIO_AA_DEBOUNCE_THRESHOLD_05_0
Offset: 0x18a4

GPIO_AA_DEBOUNCE_THRESHOLD_06_0
Offset: 0x18c4

GPIO_AA_DEBOUNCE_THRESHOLD_07_0
Offset: 0x18e4

GPIO_BB_DEBOUNCE_THRESHOLD_00_0
Offset: 0x1a04

GPIO_BB_DEBOUNCE_THRESHOLD_01_0
Offset: 0x1a24

GPIO_BB_DEBOUNCE_THRESHOLD_02_0
Offset: 0x1a44

GPIO_BB_DEBOUNCE_THRESHOLD_03_0
Offset: 0x1a64

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GPIO_<*i*>_SCR_0<*j*>_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**DEBOUNCE_THRESHOLD**: This field holds the Debounce threshold in msec to specify the debounce interval. **0** = No Debounce. **1** ~ N-1: Debouncing interval of 1 msec to (N-1) msec. **NOTE:** The actual Debounce interval has an accuracy granularity of 1 msec. Hence, when programmed to "1", the actual Debouncing interval is between 1 msec and 2 msec.

GPIO_<*i*>_INPUT_0<*j*>_0,

where <*i*> = A, B, C, D, E, F, G, H, I, J, K, L, M, N, P, Q, R, X, Y, Z, AA, AC, AD, AE, AF, AG, BB, CC, DD, EE, GG and

<*j*> ∈ {0, 1, 2, 3, 4, 5, 6, 7}.

GPIO_A_INPUT_00_0
Offset: 0x10008

GPIO_A_INPUT_01_0
Offset: 0x10028

GPIO_A_INPUT_02_0
Offset: 0x10048

GPIO_A_INPUT_03_0
Offset: 0x10068

GPIO_A_INPUT_04_0
Offset: 0x10088

GPIO_A_INPUT_05_0
Offset: 0x100a8

GPIO_A_INPUT_06_0
Offset: 0x100c8

GPIO_A_INPUT_07_0
Offset: 0x100e8

GPIO_AC_INPUT_00_0
Offset: 0x10208

GPIO_AC_INPUT_01_0
Offset: 0x10228

GPIO_AC_INPUT_02_0
Offset: 0x10248

GPIO_AC_INPUT_03_0
Offset: 0x10268

GPIO_AC_INPUT_04_0
Offset: 0x10288

GPIO_AC_INPUT_05_0
Offset: 0x102a8

GPIO_AC_INPUT_06_0
Offset: 0x102c8

GPIO_AC_INPUT_07_0
Offset: 0x102e8

GPIO_AD_INPUT_00_0
Offset: 0x10408

GPIO_AD_INPUT_01_0
Offset: 0x10428

GPIO_AD_INPUT_02_0
Offset: 0x10448

GPIO_AD_INPUT_03_0
Offset: 0x10468

GPIO_B_INPUT_00_0
Offset: 0x10608

GPIO_X_INPUT_00_0
Offset: 0x11008

GPIO_X_INPUT_01_0
Offset: 0x11028

GPIO_X_INPUT_02_0
Offset: 0x11048

GPIO_X_INPUT_03_0
Offset: 0x11068

GPIO_X_INPUT_04_0
Offset: 0x11088

GPIO_X_INPUT_05_0
Offset: 0x110a8

GPIO_X_INPUT_06_0
Offset: 0x110c8

GPIO_X_INPUT_07_0
Offset: 0x110e8

GPIO_Y_INPUT_00_0
Offset: 0x11208

GPIO_Y_INPUT_01_0
Offset: 0x11228

GPIO_Y_INPUT_02_0
Offset: 0x11248

GPIO_Y_INPUT_03_0
Offset: 0x11268

GPIO_Y_INPUT_04_0
Offset: 0x11288

GPIO_Y_INPUT_05_0
Offset: 0x112a8

GPIO_Y_INPUT_06_0
Offset: 0x112c8

GPIO_Y_INPUT_07_0
Offset: 0x112e8

GPIO_Z_INPUT_00_0
Offset: 0x11408

GPIO_Z_INPUT_01_0
Offset: 0x11428

GPIO_Z_INPUT_02_0
Offset: 0x11448

GPIO_Z_INPUT_03_0
Offset: 0x11468

GPIO_Z_INPUT_04_0
Offset: 0x11488

GPIO_Z_INPUT_05_0
Offset: 0x114a8

GPIO_Z_INPUT_06_0
Offset: 0x114c8

GPIO_Z_INPUT_07_0
Offset: 0x114e8

GPIO_M_INPUT_00_0
Offset: 0x12008

GPIO_M_INPUT_01_0
Offset: 0x12028

GPIO_M_INPUT_02_0
Offset: 0x12048

GPIO_M_INPUT_03_0
Offset: 0x12068

GPIO_M_INPUT_04_0
Offset: 0x12088

GPIO_M_INPUT_05_0
Offset: 0x120a8

GPIO_M_INPUT_06_0
Offset: 0x120c8

GPIO_M_INPUT_07_0
Offset: 0x120e8

GPIO_N_INPUT_00_0
Offset: 0x12208

GPIO_N_INPUT_01_0
Offset: 0x12228

GPIO_N_INPUT_02_0
Offset: 0x12248

GPIO_N_INPUT_03_0
Offset: 0x12268

GPIO_N_INPUT_04_0
Offset: 0x12288

GPIO_N_INPUT_05_0
Offset: 0x122a8

GPIO_N_INPUT_06_0
Offset: 0x122c8

GPIO_N_INPUT_07_0
Offset: 0x122e8

GPIO_P_INPUT_00_0
Offset: 0x12408

GPIO_P_INPUT_01_0
Offset: 0x12428

GPIO_P_INPUT_02_0
Offset: 0x12448

GPIO_P_INPUT_03_0
Offset: 0x12468

GPIO_P_INPUT_04_0
Offset: 0x12488

GPIO_P_INPUT_05_0
Offset: 0x124a8

GPIO_P_INPUT_06_0
Offset: 0x124c8

GPIO_P_INPUT_07_0
Offset: 0x124e8

GPIO_Q_INPUT_00_0
Offset: 0x12608

GPIO_Q_INPUT_01_0
Offset: 0x12628

GPIO_Q_INPUT_02_0
Offset: 0x12648

GPIO_Q_INPUT_03_0
Offset: 0x12668

GPIO_Q_INPUT_04_0
Offset: 0x12688

GPIO_Q_INPUT_05_0
Offset: 0x126a8

GPIO_Q_INPUT_06_0
Offset: 0x126c8

GPIO_Q_INPUT_07_0
Offset: 0x126e8

GPIO_R_INPUT_00_0
Offset: 0x12808

GPIO_R_INPUT_01_0
Offset: 0x12828

GPIO_R_INPUT_02_0
Offset: 0x12848

GPIO_R_INPUT_03_0
Offset: 0x12868

GPIO_R_INPUT_04_0
Offset: 0x12888

GPIO_R_INPUT_05_0
Offset: 0x128a8

GPIO_K_INPUT_00_0
Offset: 0x13008

GPIO_K_INPUT_01_0
Offset: 0x13028

GPIO_K_INPUT_02_0
Offset: 0x13048

GPIO_K_INPUT_03_0
Offset: 0x13068

GPIO_K_INPUT_04_0
Offset: 0x13088

GPIO_K_INPUT_05_0
Offset: 0x130a8

GPIO_K_INPUT_06_0
Offset: 0x130c8

GPIO_K_INPUT_07_0
Offset: 0x130e8

GPIO_L_INPUT_00_0
Offset: 0x13208

GPIO_L_INPUT_01_0
Offset: 0x13228

GPIO_L_INPUT_02_0
Offset: 0x13248

GPIO_L_INPUT_03_0
Offset: 0x13268

GPIO_AG_INPUT_00_0
Offset: 0x13408

GPIO_AG_INPUT_01_0
Offset: 0x13428

GPIO_AG_INPUT_02_0
Offset: 0x13448

GPIO_AG_INPUT_03_0
Offset: 0x13468

GPIO_AG_INPUT_04_0
Offset: 0x13488

GPIO_AG_INPUT_05_0
Offset: 0x134a8

GPIO_AG_INPUT_06_0
Offset: 0x134c8

GPIO_AG_INPUT_07_0
Offset: 0x134e8

GPIO_AE_INPUT_00_0
Offset: 0x13608

GPIO_AE_INPUT_01_0
Offset: 0x13628

GPIO_AF_INPUT_00_0
Offset: 0x13808

GPIO_AF_INPUT_01_0
Offset: 0x13828

GPIO_AF_INPUT_02_0
Offset: 0x13848

GPIO_AF_INPUT_03_0
Offset: 0x13868

GPIO_G_INPUT_00_0
Offset: 0x14008

GPIO_G_INPUT_01_0
Offset: 0x14028

GPIO_G_INPUT_02_0
Offset: 0x14048

GPIO_G_INPUT_03_0
Offset: 0x14068

GPIO_G_INPUT_04_0
Offset: 0x14088

GPIO_G_INPUT_05_0
Offset: 0x140a8

GPIO_G_INPUT_06_0
Offset: 0x140c8

GPIO_G_INPUT_07_0
Offset: 0x140e8

GPIO_H_INPUT_00_0
Offset: 0x14208

GPIO_H_INPUT_01_0
Offset: 0x14228

GPIO_H_INPUT_02_0
Offset: 0x14248

GPIO_H_INPUT_03_0
Offset: 0x14268

GPIO_H_INPUT_04_0
Offset: 0x14288

GPIO_H_INPUT_05_0
Offset: 0x142a8

GPIO_H_INPUT_06_0
Offset: 0x142c8

GPIO_H_INPUT_07_0
Offset: 0x142e8

GPIO_I_INPUT_00_0
Offset: 0x14408

GPIO_I_INPUT_01_0
Offset: 0x14428

GPIO_I_INPUT_02_0
Offset: 0x14448

GPIO_I_INPUT_03_0
Offset: 0x14468

GPIO_I_INPUT_04_0
Offset: 0x14488

GPIO_I_INPUT_05_0
Offset: 0x144a8

GPIO_I_INPUT_06_0
Offset: 0x144c8

GPIO_J_INPUT_00_0
Offset: 0x15008

GPIO_J_INPUT_01_0
Offset: 0x15028

GPIO_J_INPUT_02_0
Offset: 0x15048

GPIO_J_INPUT_03_0
Offset: 0x15068

GPIO_J_INPUT_04_0
Offset: 0x15088

GPIO_J_INPUT_05_0
Offset: 0x150a8

GPIO_C_INPUT_00_0
Offset: 0x15208

GPIO_C_INPUT_01_0
Offset: 0x15228

GPIO_C_INPUT_02_0
Offset: 0x15248

GPIO_C_INPUT_03_0
Offset: 0x15268

GPIO_C_INPUT_04_0
Offset: 0x15288

GPIO_C_INPUT_05_0
Offset: 0x152a8

GPIO_C_INPUT_06_0
Offset: 0x152c8

GPIO_C_INPUT_07_0
Offset: 0x152e8

GPIO_D_INPUT_00_0
Offset: 0x15408

GPIO_D_INPUT_01_0
Offset: 0x15428

GPIO_D_INPUT_02_0
Offset: 0x15448

GPIO_D_INPUT_03_0
Offset: 0x15468

GPIO_E_INPUT_00_0
Offset: 0x15608

GPIO_E_INPUT_01_0
Offset: 0x15628

GPIO_E_INPUT_02_0
Offset: 0x15648

GPIO_E_INPUT_03_0
Offset: 0x15668

GPIO_E_INPUT_04_0
Offset: 0x15688

GPIO_E_INPUT_05_0
Offset: 0x156a8

GPIO_E_INPUT_06_0
Offset: 0x156c8

GPIO_E_INPUT_07_0
Offset: 0x156e8

GPIO_F_INPUT_00_0
Offset: 0x15808

GPIO_F_INPUT_01_0
Offset: 0x15828

GPIO_F_INPUT_02_0
Offset: 0x15848

GPIO_F_INPUT_03_0
Offset: 0x15868

GPIO_F_INPUT_04_0
Offset: 0x15888

GPIO_F_INPUT_05_0
Offset: 0x158a8

GPIO_EE_INPUT_00_0
Offset: 0x1008

GPIO_EE_INPUT_01_0
Offset: 0x1028

GPIO_EE_INPUT_02_0
Offset: 0x1048

GPIO_EE_INPUT_03_0
Offset: 0x1068

GPIO_EE_INPUT_04_0
Offset: 0x1088

GPIO_EE_INPUT_05_0
Offset: 0x10a8

GPIO_EE_INPUT_06_0
Offset: 0x10c8

GPIO_EE_INPUT_07_0
Offset: 0x10e8

GPIO_GG_INPUT_00_0
Offset: 0x1208

GPIO_CC_INPUT_00_0
Offset: 0x1408

GPIO_CC_INPUT_01_0
Offset: 0x1428

GPIO_CC_INPUT_02_0
Offset: 0x1448

GPIO_CC_INPUT_03_0
Offset: 0x1468

GPIO_CC_INPUT_04_0
Offset: 0x1488

GPIO_CC_INPUT_05_0
Offset: 0x14a8

GPIO_CC_INPUT_06_0
Offset: 0x14c8

GPIO_CC_INPUT_07_0
Offset: 0x14e8

GPIO_DD_INPUT_00_0
Offset: 0x1608

GPIO_DD_INPUT_01_0
Offset: 0x1628

GPIO_DD_INPUT_02_0
Offset: 0x1648

GPIO_AA_INPUT_00_0
Offset: 0x1808

GPIO_AA_INPUT_01_0
Offset: 0x1828

GPIO_AA_INPUT_02_0
Offset: 0x1848

GPIO_AA_INPUT_03_0
Offset: 0x1868

GPIO_AA_INPUT_04_0
Offset: 0x1888

GPIO_AA_INPUT_05_0
Offset: 0x18a8

GPIO_AA_INPUT_06_0
Offset: 0x18c8

GPIO_AA_INPUT_07_0
Offset: 0x18e8

GPIO_BB_INPUT_00_0
Offset: 0x1a08

GPIO_BB_INPUT_01_0
Offset: 0x1a28

GPIO_BB_INPUT_02_0
Offset: 0x1a48

GPIO_BB_INPUT_03_0
Offset: 0x1a68

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: GPIO_<*i*>_SCR_0<*j*>_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
0	X	**GPIO_IN**: This field holds the GPIO Input value sampled after the specified Debouncing interval. It is effective only when (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE). It reflects the external value regardless of (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.IN_OUT) setting, so that the externally driven value is present in this field to aid debugging.

GPIO_<*i*>_OUTPUT_CONTROL_0<*j*>_0,

where <*i*> = A, B, C, D, E, F, G, H, I, J, K, L, M, N, P, Q, R, X, Y, Z, AA, AC, AD, AE, AF, AG, BB, CC, DD, EE, GG and

<*j*> ∈ {0, 1, 2, 3, 4, 5, 6, 7}.

GPIO_A_OUTPUT_CONTROL_00_0
Offset: 0x1000c

GPIO_A_OUTPUT_CONTROL_01_0
Offset: 0x1002c

GPIO_A_OUTPUT_CONTROL_02_0
Offset: 0x1004c

GPIO_A_OUTPUT_CONTROL_03_0
Offset: 0x1006c

GPIO_A_OUTPUT_CONTROL_04_0
Offset: 0x1008c

GPIO_A_OUTPUT_CONTROL_05_0
Offset: 0x100ac

GPIO_A_OUTPUT_CONTROL_06_0
Offset: 0x100cc

GPIO_A_OUTPUT_CONTROL_07_0
Offset: 0x100ec

GPIO_AC_OUTPUT_CONTROL_00_0
Offset: 0x1020c

GPIO_AC_OUTPUT_CONTROL_01_0
Offset: 0x1022c

GPIO_AC_OUTPUT_CONTROL_02_0
Offset: 0x1024c

GPIO_AC_OUTPUT_CONTROL_03_0
Offset: 0x1026c

GPIO_AC_OUTPUT_CONTROL_04_0
Offset: 0x1028c

GPIO_AC_OUTPUT_CONTROL_05_0
Offset: 0x102ac

GPIO_AC_OUTPUT_CONTROL_06_0
Offset: 0x102cc

GPIO_AC_OUTPUT_CONTROL_07_0
Offset: 0x102ec

GPIO_AD_OUTPUT_CONTROL_00_0
Offset: 0x1040c

GPIO_AD_OUTPUT_CONTROL_01_0
Offset: 0x1042c

GPIO_AD_OUTPUT_CONTROL_02_0
Offset: 0x1044c

GPIO_AD_OUTPUT_CONTROL_03_0
Offset: 0x1046c

GPIO_B_OUTPUT_CONTROL_00_0
Offset: 0x1060c

GPIO_X_OUTPUT_CONTROL_00_0
Offset: 0x1100c

GPIO_X_OUTPUT_CONTROL_01_0
Offset: 0x1102c

GPIO_X_OUTPUT_CONTROL_02_0
Offset: 0x1104c

GPIO_X_OUTPUT_CONTROL_03_0
Offset: 0x1106c

GPIO_X_OUTPUT_CONTROL_04_0
Offset: 0x1108c

GPIO_X_OUTPUT_CONTROL_05_0
Offset: 0x110ac

GPIO_X_OUTPUT_CONTROL_06_0
Offset: 0x110cc

GPIO_X_OUTPUT_CONTROL_07_0
Offset: 0x110ec

GPIO_Y_OUTPUT_CONTROL_00_0
Offset: 0x1120c

GPIO_Y_OUTPUT_CONTROL_01_0
Offset: 0x1122c

GPIO_Y_OUTPUT_CONTROL_02_0
Offset: 0x1124c

GPIO_Y_OUTPUT_CONTROL_03_0
Offset: 0x1126c

GPIO_Y_OUTPUT_CONTROL_04_0
Offset: 0x1128c

GPIO_Y_OUTPUT_CONTROL_05_0
Offset: 0x112ac

GPIO_Y_OUTPUT_CONTROL_06_0
Offset: 0x112cc

GPIO_Y_OUTPUT_CONTROL_07_0
Offset: 0x112ec

GPIO_Z_OUTPUT_CONTROL_00_0
Offset: 0x1140c

GPIO_Z_OUTPUT_CONTROL_01_0
Offset: 0x1142c

GPIO_Z_OUTPUT_CONTROL_02_0
Offset: 0x1144c

GPIO_Z_OUTPUT_CONTROL_03_0
Offset: 0x1146c

GPIO_Z_OUTPUT_CONTROL_04_0
Offset: 0x1148c

GPIO_Z_OUTPUT_CONTROL_05_0
Offset: 0x114ac

GPIO_Z_OUTPUT_CONTROL_06_0
Offset: 0x114cc

GPIO_Z_OUTPUT_CONTROL_07_0
Offset: 0x114ec

GPIO_M_OUTPUT_CONTROL_00_0
Offset: 0x1200c

GPIO_M_OUTPUT_CONTROL_01_0
Offset: 0x1202c

GPIO_M_OUTPUT_CONTROL_02_0
Offset: 0x1204c

GPIO_M_OUTPUT_CONTROL_03_0
Offset: 0x1206c

GPIO_M_OUTPUT_CONTROL_04_0
Offset: 0x1208c

GPIO_M_OUTPUT_CONTROL_05_0
Offset: 0x120ac

GPIO_M_OUTPUT_CONTROL_06_0
Offset: 0x120cc

GPIO_M_OUTPUT_CONTROL_07_0
Offset: 0x120ec

GPIO_N_OUTPUT_CONTROL_00_0
Offset: 0x1220c

GPIO_N_OUTPUT_CONTROL_01_0
Offset: 0x1222c

GPIO_N_OUTPUT_CONTROL_02_0
Offset: 0x1224c

GPIO_N_OUTPUT_CONTROL_03_0
Offset: 0x1226c

GPIO_N_OUTPUT_CONTROL_04_0
Offset: 0x1228c

GPIO_N_OUTPUT_CONTROL_05_0
Offset: 0x122ac

GPIO_N_OUTPUT_CONTROL_06_0
Offset: 0x122cc

GPIO_N_OUTPUT_CONTROL_07_0
Offset: 0x122ec

GPIO_P_OUTPUT_CONTROL_00_0
Offset: 0x1240c

GPIO_P_OUTPUT_CONTROL_01_0
Offset: 0x1242c

GPIO_P_OUTPUT_CONTROL_02_0
Offset: 0x1244c

GPIO_P_OUTPUT_CONTROL_03_0
Offset: 0x1246c

GPIO_P_OUTPUT_CONTROL_04_0
Offset: 0x1248c

GPIO_P_OUTPUT_CONTROL_05_0
Offset: 0x124ac

GPIO_P_OUTPUT_CONTROL_06_0
Offset: 0x124cc

GPIO_P_OUTPUT_CONTROL_07_0
Offset: 0x124ec

GPIO_Q_OUTPUT_CONTROL_00_0
Offset: 0x1260c

GPIO_Q_OUTPUT_CONTROL_01_0
Offset: 0x1262c

GPIO_Q_OUTPUT_CONTROL_02_0
Offset: 0x1264c

GPIO_Q_OUTPUT_CONTROL_03_0
Offset: 0x1266c

GPIO_Q_OUTPUT_CONTROL_04_0
Offset: 0x1268c

GPIO_Q_OUTPUT_CONTROL_05_0
Offset: 0x126ac

GPIO_Q_OUTPUT_CONTROL_06_0
Offset: 0x126cc

GPIO_Q_OUTPUT_CONTROL_07_0
Offset: 0x126ec

GPIO_R_OUTPUT_CONTROL_00_0
Offset: 0x1280c

GPIO_R_OUTPUT_CONTROL_01_0
Offset: 0x1282c

GPIO_R_OUTPUT_CONTROL_02_0
Offset: 0x1284c

GPIO_R_OUTPUT_CONTROL_03_0
Offset: 0x1286c

GPIO_R_OUTPUT_CONTROL_04_0
Offset: 0x1288c

GPIO_R_OUTPUT_CONTROL_05_0
Offset: 0x128ac

GPIO_K_OUTPUT_CONTROL_00_0
Offset: 0x1300c

GPIO_K_OUTPUT_CONTROL_01_0
Offset: 0x1302c

GPIO_K_OUTPUT_CONTROL_02_0
Offset: 0x1304c

GPIO_K_OUTPUT_CONTROL_03_0
Offset: 0x1306c

GPIO_K_OUTPUT_CONTROL_04_0
Offset: 0x1308c

GPIO_K_OUTPUT_CONTROL_05_0
Offset: 0x130ac

GPIO_K_OUTPUT_CONTROL_06_0
Offset: 0x130cc

GPIO_K_OUTPUT_CONTROL_07_0
Offset: 0x130ec

GPIO_L_OUTPUT_CONTROL_00_0
Offset: 0x1320c

GPIO_L_OUTPUT_CONTROL_01_0
Offset: 0x1322c

GPIO_L_OUTPUT_CONTROL_02_0
Offset: 0x1324c

GPIO_L_OUTPUT_CONTROL_03_0
Offset: 0x1326c

GPIO_AG_OUTPUT_CONTROL_00_0
Offset: 0x1340c

GPIO_AG_OUTPUT_CONTROL_01_0
Offset: 0x1342c

GPIO_AG_OUTPUT_CONTROL_02_0
Offset: 0x1344c

GPIO_AG_OUTPUT_CONTROL_03_0
Offset: 0x1346c

GPIO_AG_OUTPUT_CONTROL_04_0
Offset: 0x1348c

GPIO_AG_OUTPUT_CONTROL_05_0
Offset: 0x134ac

GPIO_AG_OUTPUT_CONTROL_06_0
Offset: 0x134cc

GPIO_AG_OUTPUT_CONTROL_07_0
Offset: 0x134ec

GPIO_AE_OUTPUT_CONTROL_00_0
Offset: 0x1360c

GPIO_AE_OUTPUT_CONTROL_01_0
Offset: 0x1362c

GPIO_AF_OUTPUT_CONTROL_00_0
Offset: 0x1380c

GPIO_AF_OUTPUT_CONTROL_01_0
Offset: 0x1382c

GPIO_AF_OUTPUT_CONTROL_02_0
Offset: 0x1384c

GPIO_AF_OUTPUT_CONTROL_03_0
Offset: 0x1386c

GPIO_G_OUTPUT_CONTROL_00_0
Offset: 0x1400c

GPIO_G_OUTPUT_CONTROL_01_0
Offset: 0x1402c

GPIO_G_OUTPUT_CONTROL_02_0
Offset: 0x1404c

GPIO_G_OUTPUT_CONTROL_03_0
Offset: 0x1406c

GPIO_G_OUTPUT_CONTROL_04_0
Offset: 0x1408c

GPIO_G_OUTPUT_CONTROL_05_0
Offset: 0x140ac

GPIO_G_OUTPUT_CONTROL_06_0
Offset: 0x140cc

GPIO_G_OUTPUT_CONTROL_07_0
Offset: 0x140ec

GPIO_H_OUTPUT_CONTROL_00_0
Offset: 0x1420c

GPIO_H_OUTPUT_CONTROL_01_0
Offset: 0x1422c

GPIO_H_OUTPUT_CONTROL_02_0
Offset: 0x1424c

GPIO_H_OUTPUT_CONTROL_03_0
Offset: 0x1426c

GPIO_H_OUTPUT_CONTROL_04_0
Offset: 0x1428c

GPIO_H_OUTPUT_CONTROL_05_0
Offset: 0x142ac

GPIO_H_OUTPUT_CONTROL_06_0
Offset: 0x142cc

GPIO_H_OUTPUT_CONTROL_07_0
Offset: 0x142ec

GPIO_I_OUTPUT_CONTROL_00_0
Offset: 0x1440c

GPIO_I_OUTPUT_CONTROL_01_0
Offset: 0x1442c

GPIO_I_OUTPUT_CONTROL_02_0
Offset: 0x1444c

GPIO_I_OUTPUT_CONTROL_03_0
Offset: 0x1446c

GPIO_I_OUTPUT_CONTROL_04_0
Offset: 0x1448c

GPIO_I_OUTPUT_CONTROL_05_0
Offset: 0x144ac

GPIO_I_OUTPUT_CONTROL_06_0
Offset: 0x144cc

GPIO_J_OUTPUT_CONTROL_00_0
Offset: 0x1500c

GPIO_J_OUTPUT_CONTROL_01_0
Offset: 0x1502c

GPIO_J_OUTPUT_CONTROL_02_0
Offset: 0x1504c

GPIO_J_OUTPUT_CONTROL_03_0
Offset: 0x1506c

GPIO_J_OUTPUT_CONTROL_04_0
Offset: 0x1508c

GPIO_J_OUTPUT_CONTROL_05_0
Offset: 0x150ac

GPIO_C_OUTPUT_CONTROL_00_0
Offset: 0x1520c

GPIO_C_OUTPUT_CONTROL_01_0
Offset: 0x1522c

GPIO_C_OUTPUT_CONTROL_02_0
Offset: 0x1524c

GPIO_C_OUTPUT_CONTROL_03_0
Offset: 0x1526c

GPIO_C_OUTPUT_CONTROL_04_0
Offset: 0x1528c

GPIO_C_OUTPUT_CONTROL_05_0
Offset: 0x152ac

GPIO_C_OUTPUT_CONTROL_06_0
Offset: 0x152cc

GPIO_C_OUTPUT_CONTROL_07_0
Offset: 0x152ec

GPIO_D_OUTPUT_CONTROL_00_0
Offset: 0x1540c

GPIO_D_OUTPUT_CONTROL_01_0
Offset: 0x1542c

GPIO_D_OUTPUT_CONTROL_02_0
Offset: 0x1544c

GPIO_D_OUTPUT_CONTROL_03_0
Offset: 0x1546c

GPIO_E_OUTPUT_CONTROL_00_0
Offset: 0x1560c

GPIO_E_OUTPUT_CONTROL_01_0
Offset: 0x1562c

GPIO_E_OUTPUT_CONTROL_02_0
Offset: 0x1564c

GPIO_E_OUTPUT_CONTROL_03_0
Offset: 0x1566c

GPIO_E_OUTPUT_CONTROL_04_0
Offset: 0x1568c

GPIO_E_OUTPUT_CONTROL_05_0
Offset: 0x156ac

GPIO_E_OUTPUT_CONTROL_06_0
Offset: 0x156cc

GPIO_E_OUTPUT_CONTROL_07_0
Offset: 0x156ec

GPIO_F_OUTPUT_CONTROL_00_0
Offset: 0x1580c

GPIO_F_OUTPUT_CONTROL_01_0
Offset: 0x1582c

GPIO_F_OUTPUT_CONTROL_02_0
Offset: 0x1584c

GPIO_F_OUTPUT_CONTROL_03_0
Offset: 0x1586c

GPIO_F_OUTPUT_CONTROL_04_0
Offset: 0x1588c

GPIO_F_OUTPUT_CONTROL_05_0
Offset: 0x158ac

GPIO_EE_OUTPUT_CONTROL_00_0
Offset: 0x100c

GPIO_EE_OUTPUT_CONTROL_01_0
Offset: 0x102c

GPIO_EE_OUTPUT_CONTROL_02_0
Offset: 0x104c

GPIO_EE_OUTPUT_CONTROL_03_0
Offset: 0x106c

GPIO_EE_OUTPUT_CONTROL_04_0
Offset: 0x108c

GPIO_EE_OUTPUT_CONTROL_05_0
Offset: 0x10ac

GPIO_EE_OUTPUT_CONTROL_06_0
Offset: 0x10cc

GPIO_EE_OUTPUT_CONTROL_07_0
Offset: 0x10ec

GPIO_GG_OUTPUT_CONTROL_00_0
Offset: 0x120c

GPIO_CC_OUTPUT_CONTROL_00_0
Offset: 0x140c

GPIO_CC_OUTPUT_CONTROL_01_0
Offset: 0x142c

GPIO_CC_OUTPUT_CONTROL_02_0
Offset: 0x144c

GPIO_CC_OUTPUT_CONTROL_03_0
Offset: 0x146c

GPIO_CC_OUTPUT_CONTROL_04_0
Offset: 0x148c

GPIO_CC_OUTPUT_CONTROL_05_0
Offset: 0x14ac

GPIO_CC_OUTPUT_CONTROL_06_0
Offset: 0x14cc

GPIO_CC_OUTPUT_CONTROL_07_0
Offset: 0x14ec

GPIO_DD_OUTPUT_CONTROL_00_0
Offset: 0x160c

GPIO_DD_OUTPUT_CONTROL_01_0
Offset: 0x162c

GPIO_DD_OUTPUT_CONTROL_02_0
Offset: 0x164c

GPIO_AA_OUTPUT_CONTROL_00_0
Offset: 0x180c

GPIO_AA_OUTPUT_CONTROL_01_0
Offset: 0x182c

GPIO_AA_OUTPUT_CONTROL_02_0
Offset: 0x184c

GPIO_AA_OUTPUT_CONTROL_03_0
Offset: 0x186c

GPIO_AA_OUTPUT_CONTROL_04_0
Offset: 0x188c

GPIO_AA_OUTPUT_CONTROL_05_0
Offset: 0x18ac

GPIO_AA_OUTPUT_CONTROL_06_0
Offset: 0x18cc

GPIO_AA_OUTPUT_CONTROL_07_0
Offset: 0x18ec

GPIO_BB_OUTPUT_CONTROL_00_0
Offset: 0x1a0c

GPIO_BB_OUTPUT_CONTROL_01_0
Offset: 0x1a2c

GPIO_BB_OUTPUT_CONTROL_02_0
Offset: 0x1a4c

GPIO_BB_OUTPUT_CONTROL_03_0
Offset: 0x1a6c

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GPIO_<*i*>_SCR_0<*j*>_0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	0x1	**GPIO_OUT_CONTROL**: This field Indicates whether the GPIO as an output is Floated or Actively driven. It is effective only when ((GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE) && (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.IN_OUT == OUT)). It is floated regardless of the value specified here, When ((GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == DISABLE) || (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.IN_OUT == IN)). **0** = DRIVEN. **1** = FLOATED.

GPIO_<*i*>_OUTPUT_VALUE_0<*j*>_0,

where <*i*> = A, B, C, D, E, F, G, H, I, J, K, L, M, N, P, Q, R, X, Y, Z, AA, AC, AD, AE, AF, AG, BB, CC, DD, EE, GG and

<*j*> ∈ {0, 1, 2, 3, 4, 5, 6, 7}.

GPIO_A_OUTPUT_VALUE_00_0
Offset: 0x10010

GPIO_A_OUTPUT_VALUE_01_0
Offset: 0x10030

GPIO_A_OUTPUT_VALUE_02_0
Offset: 0x10050

GPIO_A_OUTPUT_VALUE_03_0
Offset: 0x10070

GPIO_A_OUTPUT_VALUE_04_0
Offset: 0x10090

GPIO_A_OUTPUT_VALUE_05_0
Offset: 0x100b0

GPIO_A_OUTPUT_VALUE_06_0
Offset: 0x100d0

GPIO_A_OUTPUT_VALUE_07_0
Offset: 0x100f0

GPIO_AC_OUTPUT_VALUE_00_0
Offset: 0x10210

GPIO_AC_OUTPUT_VALUE_01_0
Offset: 0x10230

GPIO_AC_OUTPUT_VALUE_02_0
Offset: 0x10250

GPIO_AC_OUTPUT_VALUE_03_0
Offset: 0x10270

GPIO_AC_OUTPUT_VALUE_04_0
Offset: 0x10290

GPIO_AC_OUTPUT_VALUE_05_0
Offset: 0x102b0

GPIO_AC_OUTPUT_VALUE_06_0
Offset: 0x102d0

GPIO_AC_OUTPUT_VALUE_07_0
Offset: 0x102f0

GPIO_AD_OUTPUT_VALUE_00_0
Offset: 0x10410

GPIO_AD_OUTPUT_VALUE_01_0
Offset: 0x10430

GPIO_AD_OUTPUT_VALUE_02_0
Offset: 0x10450

GPIO_AD_OUTPUT_VALUE_03_0
Offset: 0x10470

GPIO_B_OUTPUT_VALUE_00_0
Offset: 0x10610

GPIO_X_OUTPUT_VALUE_00_0
Offset: 0x11010

GPIO_X_OUTPUT_VALUE_01_0
Offset: 0x11030

GPIO_X_OUTPUT_VALUE_02_0
Offset: 0x11050

GPIO_X_OUTPUT_VALUE_03_0
Offset: 0x11070

GPIO_X_OUTPUT_VALUE_04_0
Offset: 0x11090

GPIO_X_OUTPUT_VALUE_05_0
Offset: 0x110b0

GPIO_X_OUTPUT_VALUE_06_0
Offset: 0x110d0

GPIO_X_OUTPUT_VALUE_07_0
Offset: 0x110f0

GPIO_Y_OUTPUT_VALUE_00_0
Offset: 0x11210

GPIO_Y_OUTPUT_VALUE_01_0
Offset: 0x11230

GPIO_Y_OUTPUT_VALUE_02_0
Offset: 0x11250

GPIO_Y_OUTPUT_VALUE_03_0
Offset: 0x11270

GPIO_Y_OUTPUT_VALUE_04_0
Offset: 0x11290

GPIO_Y_OUTPUT_VALUE_05_0
Offset: 0x112b0

GPIO_Y_OUTPUT_VALUE_06_0
Offset: 0x112d0

GPIO_Y_OUTPUT_VALUE_07_0
Offset: 0x112f0

GPIO_Z_OUTPUT_VALUE_00_0
Offset: 0x11410

GPIO_Z_OUTPUT_VALUE_01_0
Offset: 0x11430

GPIO_Z_OUTPUT_VALUE_02_0
Offset: 0x11450

GPIO_Z_OUTPUT_VALUE_03_0
Offset: 0x11470

GPIO_Z_OUTPUT_VALUE_04_0
Offset: 0x11490

GPIO_Z_OUTPUT_VALUE_05_0
Offset: 0x114b0

GPIO_Z_OUTPUT_VALUE_06_0
Offset: 0x114d0

GPIO_Z_OUTPUT_VALUE_07_0
Offset: 0x114f0

GPIO_M_OUTPUT_VALUE_00_0
Offset: 0x12010

GPIO_M_OUTPUT_VALUE_01_0
Offset: 0x12030

GPIO_M_OUTPUT_VALUE_02_0
Offset: 0x12050

GPIO_M_OUTPUT_VALUE_03_0
Offset: 0x12070

GPIO_M_OUTPUT_VALUE_04_0
Offset: 0x12090

GPIO_M_OUTPUT_VALUE_05_0
Offset: 0x120b0

GPIO_M_OUTPUT_VALUE_06_0
Offset: 0x120d0

GPIO_M_OUTPUT_VALUE_07_0
Offset: 0x120f0

GPIO_N_OUTPUT_VALUE_00_0
Offset: 0x12210

GPIO_N_OUTPUT_VALUE_01_0
Offset: 0x12230

GPIO_N_OUTPUT_VALUE_02_0
Offset: 0x12250

GPIO_N_OUTPUT_VALUE_03_0
Offset: 0x12270

GPIO_N_OUTPUT_VALUE_04_0
Offset: 0x12290

GPIO_N_OUTPUT_VALUE_05_0
Offset: 0x122b0

GPIO_N_OUTPUT_VALUE_06_0
Offset: 0x122d0

GPIO_N_OUTPUT_VALUE_07_0
Offset: 0x122f0

GPIO_P_OUTPUT_VALUE_00_0
Offset: 0x12410

GPIO_P_OUTPUT_VALUE_01_0
Offset: 0x12430

GPIO_P_OUTPUT_VALUE_02_0
Offset: 0x12450

GPIO_P_OUTPUT_VALUE_03_0
Offset: 0x12470

GPIO_P_OUTPUT_VALUE_04_0
Offset: 0x12490

GPIO_P_OUTPUT_VALUE_05_0
Offset: 0x124b0

GPIO_P_OUTPUT_VALUE_06_0
Offset: 0x124d0

GPIO_P_OUTPUT_VALUE_07_0
Offset: 0x124f0

GPIO_Q_OUTPUT_VALUE_00_0
Offset: 0x12610

GPIO_Q_OUTPUT_VALUE_01_0
Offset: 0x12630

GPIO_Q_OUTPUT_VALUE_02_0
Offset: 0x12650

GPIO_Q_OUTPUT_VALUE_03_0
Offset: 0x12670

GPIO_Q_OUTPUT_VALUE_04_0
Offset: 0x12690

GPIO_Q_OUTPUT_VALUE_05_0
Offset: 0x126b0

GPIO_Q_OUTPUT_VALUE_06_0
Offset: 0x126d0

GPIO_Q_OUTPUT_VALUE_07_0
Offset: 0x126f0

GPIO_R_OUTPUT_VALUE_00_0
Offset: 0x12810

GPIO_R_OUTPUT_VALUE_01_0
Offset: 0x12830

GPIO_R_OUTPUT_VALUE_02_0
Offset: 0x12850

GPIO_R_OUTPUT_VALUE_03_0
Offset: 0x12870

GPIO_R_OUTPUT_VALUE_04_0
Offset: 0x12890

GPIO_R_OUTPUT_VALUE_05_0
Offset: 0x128b0

GPIO_K_OUTPUT_VALUE_00_0
Offset: 0x13010

GPIO_K_OUTPUT_VALUE_01_0
Offset: 0x13030

GPIO_K_OUTPUT_VALUE_02_0
Offset: 0x13050

GPIO_K_OUTPUT_VALUE_03_0
Offset: 0x13070

GPIO_K_OUTPUT_VALUE_04_0
Offset: 0x13090

GPIO_K_OUTPUT_VALUE_05_0
Offset: 0x130b0

GPIO_K_OUTPUT_VALUE_06_0
Offset: 0x130d0

GPIO_K_OUTPUT_VALUE_07_0
Offset: 0x130f0

GPIO_L_OUTPUT_VALUE_00_0
Offset: 0x13210

GPIO_L_OUTPUT_VALUE_01_0
Offset: 0x13230

GPIO_L_OUTPUT_VALUE_02_0
Offset: 0x13250

GPIO_L_OUTPUT_VALUE_03_0
Offset: 0x13270

GPIO_AG_OUTPUT_VALUE_00_0
Offset: 0x13410

GPIO_AG_OUTPUT_VALUE_01_0
Offset: 0x13430

GPIO_AG_OUTPUT_VALUE_02_0
Offset: 0x13450

GPIO_AG_OUTPUT_VALUE_03_0
Offset: 0x13470

GPIO_AG_OUTPUT_VALUE_04_0
Offset: 0x13490

GPIO_AG_OUTPUT_VALUE_05_0
Offset: 0x134b0

GPIO_AG_OUTPUT_VALUE_06_0
Offset: 0x134d0

GPIO_AG_OUTPUT_VALUE_07_0
Offset: 0x134f0

GPIO_AE_OUTPUT_VALUE_00_0
Offset: 0x13610

GPIO_AE_OUTPUT_VALUE_01_0
Offset: 0x13630

GPIO_AF_OUTPUT_VALUE_00_0
Offset: 0x13810

GPIO_AF_OUTPUT_VALUE_01_0
Offset: 0x13830

GPIO_AF_OUTPUT_VALUE_02_0
Offset: 0x13850

GPIO_AF_OUTPUT_VALUE_03_0
Offset: 0x13870

GPIO_G_OUTPUT_VALUE_00_0
Offset: 0x14010

GPIO_G_OUTPUT_VALUE_01_0
Offset: 0x14030

GPIO_G_OUTPUT_VALUE_02_0
Offset: 0x14050

GPIO_G_OUTPUT_VALUE_03_0
Offset: 0x14070

GPIO_G_OUTPUT_VALUE_04_0
Offset: 0x14090

GPIO_G_OUTPUT_VALUE_05_0
Offset: 0x140b0

GPIO_G_OUTPUT_VALUE_06_0
Offset: 0x140d0

GPIO_G_OUTPUT_VALUE_07_0
Offset: 0x140f0

GPIO_H_OUTPUT_VALUE_00_0
Offset: 0x14210

GPIO_H_OUTPUT_VALUE_01_0
Offset: 0x14230

GPIO_H_OUTPUT_VALUE_02_0
Offset: 0x14250

GPIO_H_OUTPUT_VALUE_03_0
Offset: 0x14270

GPIO_H_OUTPUT_VALUE_04_0
Offset: 0x14290

GPIO_H_OUTPUT_VALUE_05_0
Offset: 0x142b0

GPIO_H_OUTPUT_VALUE_06_0
Offset: 0x142d0

GPIO_H_OUTPUT_VALUE_07_0
Offset: 0x142f0

GPIO_I_OUTPUT_VALUE_00_0
Offset: 0x14410

GPIO_I_OUTPUT_VALUE_01_0
Offset: 0x14430

GPIO_I_OUTPUT_VALUE_02_0
Offset: 0x14450

GPIO_I_OUTPUT_VALUE_03_0
Offset: 0x14470

GPIO_I_OUTPUT_VALUE_04_0
Offset: 0x14490

GPIO_I_OUTPUT_VALUE_05_0
Offset: 0x144b0

GPIO_I_OUTPUT_VALUE_06_0
Offset: 0x144d0

GPIO_J_OUTPUT_VALUE_00_0
Offset: 0x15010

GPIO_J_OUTPUT_VALUE_01_0
Offset: 0x15030

GPIO_J_OUTPUT_VALUE_02_0
Offset: 0x15050

GPIO_J_OUTPUT_VALUE_03_0
Offset: 0x15070

GPIO_J_OUTPUT_VALUE_04_0
Offset: 0x15090

GPIO_J_OUTPUT_VALUE_05_0
Offset: 0x150b0

GPIO_C_OUTPUT_VALUE_00_0
Offset: 0x15210

GPIO_C_OUTPUT_VALUE_01_0
Offset: 0x15230

GPIO_C_OUTPUT_VALUE_02_0
Offset: 0x15250

GPIO_C_OUTPUT_VALUE_03_0
Offset: 0x15270

GPIO_C_OUTPUT_VALUE_04_0
Offset: 0x15290

GPIO_C_OUTPUT_VALUE_05_0
Offset: 0x152b0

GPIO_C_OUTPUT_VALUE_06_0
Offset: 0x152d0

GPIO_C_OUTPUT_VALUE_07_0
Offset: 0x152f0

GPIO_D_OUTPUT_VALUE_00_0
Offset: 0x15410

GPIO_D_OUTPUT_VALUE_01_0
Offset: 0x15430

GPIO_D_OUTPUT_VALUE_02_0
Offset: 0x15450

GPIO_D_OUTPUT_VALUE_03_0
Offset: 0x15470

GPIO_E_OUTPUT_VALUE_00_0
Offset: 0x15610

GPIO_E_OUTPUT_VALUE_01_0
Offset: 0x15630

GPIO_E_OUTPUT_VALUE_02_0
Offset: 0x15650

GPIO_E_OUTPUT_VALUE_03_0
Offset: 0x15670

GPIO_E_OUTPUT_VALUE_04_0
Offset: 0x15690

GPIO_E_OUTPUT_VALUE_05_0
Offset: 0x156b0

GPIO_E_OUTPUT_VALUE_06_0
Offset: 0x156d0

GPIO_E_OUTPUT_VALUE_07_0
Offset: 0x156f0

GPIO_F_OUTPUT_VALUE_00_0
Offset: 0x15810

GPIO_F_OUTPUT_VALUE_01_0
Offset: 0x15830

GPIO_F_OUTPUT_VALUE_02_0
Offset: 0x15850

GPIO_F_OUTPUT_VALUE_03_0
Offset: 0x15870

GPIO_F_OUTPUT_VALUE_04_0
Offset: 0x15890

GPIO_F_OUTPUT_VALUE_05_0
Offset: 0x158b0

GPIO_EE_OUTPUT_VALUE_00_0
Offset: 0x1010

GPIO_EE_OUTPUT_VALUE_01_0
Offset: 0x1030

GPIO_EE_OUTPUT_VALUE_02_0
Offset: 0x1050

GPIO_EE_OUTPUT_VALUE_03_0
Offset: 0x1070

GPIO_EE_OUTPUT_VALUE_04_0
Offset: 0x1090

GPIO_EE_OUTPUT_VALUE_05_0
Offset: 0x10b0

GPIO_EE_OUTPUT_VALUE_06_0
Offset: 0x10d0

GPIO_EE_OUTPUT_VALUE_07_0
Offset: 0x10f0

GPIO_GG_OUTPUT_VALUE_00_0
Offset: 0x1210

GPIO_CC_OUTPUT_VALUE_00_0
Offset: 0x1410

GPIO_CC_OUTPUT_VALUE_01_0
Offset: 0x1430

GPIO_CC_OUTPUT_VALUE_02_0
Offset: 0x1450

GPIO_CC_OUTPUT_VALUE_03_0
Offset: 0x1470

GPIO_CC_OUTPUT_VALUE_04_0
Offset: 0x1490

GPIO_CC_OUTPUT_VALUE_05_0
Offset: 0x14b0

GPIO_CC_OUTPUT_VALUE_06_0
Offset: 0x14d0

GPIO_CC_OUTPUT_VALUE_07_0
Offset: 0x14f0

GPIO_DD_OUTPUT_VALUE_00_0
Offset: 0x1610

GPIO_DD_OUTPUT_VALUE_01_0
Offset: 0x1630

GPIO_DD_OUTPUT_VALUE_02_0
Offset: 0x1650

GPIO_AA_OUTPUT_VALUE_00_0
Offset: 0x1810

GPIO_AA_OUTPUT_VALUE_01_0
Offset: 0x1830

GPIO_AA_OUTPUT_VALUE_02_0
Offset: 0x1850

GPIO_AA_OUTPUT_VALUE_03_0
Offset: 0x1870

GPIO_AA_OUTPUT_VALUE_04_0
Offset: 0x1890

GPIO_AA_OUTPUT_VALUE_05_0
Offset: 0x18b0

GPIO_AA_OUTPUT_VALUE_06_0
Offset: 0x18d0

GPIO_AA_OUTPUT_VALUE_07_0
Offset: 0x18f0

GPIO_BB_OUTPUT_VALUE_00_0
Offset: 0x1a10

GPIO_BB_OUTPUT_VALUE_01_0
Offset: 0x1a30

GPIO_BB_OUTPUT_VALUE_02_0
Offset: 0x1a50

GPIO_BB_OUTPUT_VALUE_03_0
Offset: 0x1a70

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GPIO_<*i*>_SCR_0<*j*>_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**GPIO_OUT_VAL**: This field holds the GPIO output value to be driven out when (GPIO_<*i*>_OUTPUT_CONTROL_<*j*>_0.GPIO_OUT_CONTROL == DRIVEN). It is effective only when ((GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE) && (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.IN_OUT == OUT)).

GPIO_<*i*>_INTERRUPT_CLEAR_0<*j*>_0,

where <*i*> = A, B, C, D, E, F, G, H, I, J, K, L, M, N, P, Q, R, X, Y, Z, AA, AC, AD, AE, AF, AG, BB, CC, DD, EE, GG and

<*j*> ∈ {0, 1, 2, 3, 4, 5, 6, 7}.

GPIO_A_INTERRUPT_CLEAR_00_0
Offset: 0x10014

GPIO_A_INTERRUPT_CLEAR_01_0
Offset: 0x10034

GPIO_A_INTERRUPT_CLEAR_02_0
Offset: 0x10054

GPIO_A_INTERRUPT_CLEAR_03_0
Offset: 0x10074

GPIO_A_INTERRUPT_CLEAR_04_0
Offset: 0x10094

GPIO_A_INTERRUPT_CLEAR_05_0
Offset: 0x100b4

GPIO_A_INTERRUPT_CLEAR_06_0
Offset: 0x100d4

GPIO_A_INTERRUPT_CLEAR_07_0
Offset: 0x100f4
Read/Write: R/W

GPIO_AC_INTERRUPT_CLEAR_00_0
Offset: 0x10214

GPIO_AC_INTERRUPT_CLEAR_01_0
Offset: 0x10234

GPIO_AC_INTERRUPT_CLEAR_02_0
Offset: 0x10254

GPIO_AC_INTERRUPT_CLEAR_03_0
Offset: 0x10274

GPIO_AC_INTERRUPT_CLEAR_04_0
Offset: 0x10294

GPIO_AC_INTERRUPT_CLEAR_05_0
Offset: 0x102b4

GPIO_AC_INTERRUPT_CLEAR_06_0
Offset: 0x102d4

GPIO_AC_INTERRUPT_CLEAR_07_0
Offset: 0x102f4

GPIO_AD_INTERRUPT_CLEAR_00_0
Offset: 0x10414

GPIO_AD_INTERRUPT_CLEAR_01_0
Offset: 0x10434

GPIO_AD_INTERRUPT_CLEAR_02_0
Offset: 0x10454

GPIO_AD_INTERRUPT_CLEAR_03_0
Offset: 0x10474

GPIO_B_INTERRUPT_CLEAR_00_0
Offset: 0x10614

GPIO_X_INTERRUPT_CLEAR_00_0
Offset: 0x11014

GPIO_X_INTERRUPT_CLEAR_01_0
Offset: 0x11034

GPIO_X_INTERRUPT_CLEAR_02_0
Offset: 0x11054

GPIO_X_INTERRUPT_CLEAR_03_0
Offset: 0x11074

GPIO_X_INTERRUPT_CLEAR_04_0
Offset: 0x11094

GPIO_X_INTERRUPT_CLEAR_05_0
Offset: 0x110b4

GPIO_X_INTERRUPT_CLEAR_06_0
Offset: 0x110d4

GPIO_X_INTERRUPT_CLEAR_07_0
Offset: 0x110f4

GPIO_Y_INTERRUPT_CLEAR_00_0
Offset: 0x11214

GPIO_Y_INTERRUPT_CLEAR_01_0
Offset: 0x11234

GPIO_Y_INTERRUPT_CLEAR_02_0
Offset: 0x11254

GPIO_Y_INTERRUPT_CLEAR_03_0
Offset: 0x11274

GPIO_Y_INTERRUPT_CLEAR_04_0
Offset: 0x11294

GPIO_Y_INTERRUPT_CLEAR_05_0
Offset: 0x112b4

GPIO_Y_INTERRUPT_CLEAR_06_0
Offset: 0x112d4

GPIO_Y_INTERRUPT_CLEAR_07_0
Offset: 0x112f4

GPIO_Z_INTERRUPT_CLEAR_00_0
Offset: 0x11414

GPIO_Z_INTERRUPT_CLEAR_01_0
Offset: 0x11434

GPIO_Z_INTERRUPT_CLEAR_02_0
Offset: 0x11454

GPIO_Z_INTERRUPT_CLEAR_03_0
Offset: 0x11474

GPIO_Z_INTERRUPT_CLEAR_04_0
Offset: 0x11494

GPIO_Z_INTERRUPT_CLEAR_05_0
Offset: 0x114b4

GPIO_Z_INTERRUPT_CLEAR_06_0
Offset: 0x114d4

GPIO_Z_INTERRUPT_CLEAR_07_0
Offset: 0x114f4

GPIO_M_INTERRUPT_CLEAR_00_0
Offset: 0x12014

GPIO_M_INTERRUPT_CLEAR_01_0
Offset: 0x12034

GPIO_M_INTERRUPT_CLEAR_02_0
Offset: 0x12054

GPIO_M_INTERRUPT_CLEAR_03_0
Offset: 0x12074

GPIO_M_INTERRUPT_CLEAR_04_0
Offset: 0x12094

GPIO_M_INTERRUPT_CLEAR_05_0
Offset: 0x120b4

GPIO_M_INTERRUPT_CLEAR_06_0
Offset: 0x120d4

GPIO_M_INTERRUPT_CLEAR_07_0
Offset: 0x120f4

GPIO_N_INTERRUPT_CLEAR_00_0
Offset: 0x12214

GPIO_N_INTERRUPT_CLEAR_01_0
Offset: 0x12234

GPIO_N_INTERRUPT_CLEAR_02_0
Offset: 0x12254

GPIO_N_INTERRUPT_CLEAR_03_0
Offset: 0x12274

GPIO_N_INTERRUPT_CLEAR_04_0
Offset: 0x12294

GPIO_N_INTERRUPT_CLEAR_05_0
Offset: 0x122b4

GPIO_N_INTERRUPT_CLEAR_06_0
Offset: 0x122d4

GPIO_N_INTERRUPT_CLEAR_07_0
Offset: 0x122f4

GPIO_P_INTERRUPT_CLEAR_00_0
Offset: 0x12414

GPIO_P_INTERRUPT_CLEAR_01_0
Offset: 0x12434

GPIO_P_INTERRUPT_CLEAR_02_0
Offset: 0x12454

GPIO_P_INTERRUPT_CLEAR_03_0
Offset: 0x12474

GPIO_P_INTERRUPT_CLEAR_04_0
Offset: 0x12494

GPIO_P_INTERRUPT_CLEAR_05_0
Offset: 0x124b4

GPIO_P_INTERRUPT_CLEAR_06_0
Offset: 0x124d4

GPIO_P_INTERRUPT_CLEAR_07_0
Offset: 0x124f4

GPIO_Q_INTERRUPT_CLEAR_00_0
Offset: 0x12614

GPIO_Q_INTERRUPT_CLEAR_01_0
Offset: 0x12634

GPIO_Q_INTERRUPT_CLEAR_02_0
Offset: 0x12654

GPIO_Q_INTERRUPT_CLEAR_03_0
Offset: 0x12674

GPIO_Q_INTERRUPT_CLEAR_04_0
Offset: 0x12694

GPIO_Q_INTERRUPT_CLEAR_05_0
Offset: 0x126b4

GPIO_Q_INTERRUPT_CLEAR_06_0
Offset: 0x126d4

GPIO_Q_INTERRUPT_CLEAR_07_0
Offset: 0x126f4

GPIO_R_INTERRUPT_CLEAR_00_0
Offset: 0x12814

GPIO_R_INTERRUPT_CLEAR_01_0
Offset: 0x12834

GPIO_R_INTERRUPT_CLEAR_02_0
Offset: 0x12854

GPIO_R_INTERRUPT_CLEAR_03_0
Offset: 0x12874

GPIO_R_INTERRUPT_CLEAR_04_0
Offset: 0x12894

GPIO_R_INTERRUPT_CLEAR_05_0
Offset: 0x128b4

GPIO_K_INTERRUPT_CLEAR_00_0
Offset: 0x13014

GPIO_K_INTERRUPT_CLEAR_01_0
Offset: 0x13034

GPIO_K_INTERRUPT_CLEAR_02_0
Offset: 0x13054

GPIO_K_INTERRUPT_CLEAR_03_0
Offset: 0x13074

GPIO_K_INTERRUPT_CLEAR_04_0
Offset: 0x13094

GPIO_K_INTERRUPT_CLEAR_05_0
Offset: 0x130b4

GPIO_K_INTERRUPT_CLEAR_06_0
Offset: 0x130d4

GPIO_K_INTERRUPT_CLEAR_07_0
Offset: 0x130f4

GPIO_L_INTERRUPT_CLEAR_00_0
Offset: 0x13214

GPIO_L_INTERRUPT_CLEAR_01_0
Offset: 0x13234

GPIO_L_INTERRUPT_CLEAR_02_0
Offset: 0x13254

GPIO_L_INTERRUPT_CLEAR_03_0
Offset: 0x13274

GPIO_AG_INTERRUPT_CLEAR_00_0
Offset: 0x13414

GPIO_AG_INTERRUPT_CLEAR_01_0
Offset: 0x13434

GPIO_AG_INTERRUPT_CLEAR_02_0
Offset: 0x13454

GPIO_AG_INTERRUPT_CLEAR_03_0
Offset: 0x13474

GPIO_AG_INTERRUPT_CLEAR_04_0
Offset: 0x13494

GPIO_AG_INTERRUPT_CLEAR_05_0
Offset: 0x134b4

GPIO_AG_INTERRUPT_CLEAR_06_0
Offset: 0x134d4

GPIO_AG_INTERRUPT_CLEAR_07_0
Offset: 0x134f4

GPIO_AE_INTERRUPT_CLEAR_00_0
Offset: 0x13614

GPIO_AE_INTERRUPT_CLEAR_01_0
Offset: 0x13634

GPIO_AF_INTERRUPT_CLEAR_00_0
Offset: 0x13814

GPIO_AF_INTERRUPT_CLEAR_01_0
Offset: 0x13834

GPIO_AF_INTERRUPT_CLEAR_02_0
Offset: 0x13854

GPIO_AF_INTERRUPT_CLEAR_03_0
Offset: 0x13874

GPIO_G_INTERRUPT_CLEAR_00_0
Offset: 0x14014

GPIO_G_INTERRUPT_CLEAR_01_0
Offset: 0x14034

GPIO_G_INTERRUPT_CLEAR_02_0
Offset: 0x14054

GPIO_G_INTERRUPT_CLEAR_03_0
Offset: 0x14074

GPIO_G_INTERRUPT_CLEAR_04_0
Offset: 0x14094

GPIO_G_INTERRUPT_CLEAR_05_0
Offset: 0x140b4

GPIO_G_INTERRUPT_CLEAR_06_0
Offset: 0x140d4

GPIO_G_INTERRUPT_CLEAR_07_0
Offset: 0x140f4

GPIO_H_INTERRUPT_CLEAR_00_0
Offset: 0x14214

GPIO_H_INTERRUPT_CLEAR_01_0
Offset: 0x14234

GPIO_H_INTERRUPT_CLEAR_02_0
Offset: 0x14254

GPIO_H_INTERRUPT_CLEAR_03_0
Offset: 0x14274

GPIO_H_INTERRUPT_CLEAR_04_0
Offset: 0x14294

GPIO_H_INTERRUPT_CLEAR_05_0
Offset: 0x142b4

GPIO_H_INTERRUPT_CLEAR_06_0
Offset: 0x142d4

GPIO_H_INTERRUPT_CLEAR_07_0
Offset: 0x142f4

GPIO_I_INTERRUPT_CLEAR_00_0
Offset: 0x14414

GPIO_I_INTERRUPT_CLEAR_01_0
Offset: 0x14434

GPIO_I_INTERRUPT_CLEAR_02_0
Offset: 0x14454

GPIO_I_INTERRUPT_CLEAR_03_0
Offset: 0x14474

GPIO_I_INTERRUPT_CLEAR_04_0
Offset: 0x14494

GPIO_I_INTERRUPT_CLEAR_05_0
Offset: 0x144b4

GPIO_I_INTERRUPT_CLEAR_06_0
Offset: 0x144d4

GPIO_J_INTERRUPT_CLEAR_00_0
Offset: 0x15014

GPIO_J_INTERRUPT_CLEAR_01_0
Offset: 0x15034

GPIO_J_INTERRUPT_CLEAR_02_0
Offset: 0x15054

GPIO_J_INTERRUPT_CLEAR_03_0
Offset: 0x15074

GPIO_J_INTERRUPT_CLEAR_04_0
Offset: 0x15094

GPIO_J_INTERRUPT_CLEAR_05_0
Offset: 0x150b4

GPIO_C_INTERRUPT_CLEAR_00_0
Offset: 0x15214

GPIO_C_INTERRUPT_CLEAR_01_0
Offset: 0x15234

GPIO_C_INTERRUPT_CLEAR_02_0
Offset: 0x15254

GPIO_C_INTERRUPT_CLEAR_03_0
Offset: 0x15274

GPIO_C_INTERRUPT_CLEAR_04_0
Offset: 0x15294

GPIO_C_INTERRUPT_CLEAR_05_0
Offset: 0x152b4

GPIO_C_INTERRUPT_CLEAR_06_0
Offset: 0x152d4

GPIO_C_INTERRUPT_CLEAR_07_0
Offset: 0x152f4

GPIO_D_INTERRUPT_CLEAR_00_0
Offset: 0x15414

GPIO_D_INTERRUPT_CLEAR_01_0
Offset: 0x15434

GPIO_D_INTERRUPT_CLEAR_02_0
Offset: 0x15454

GPIO_D_INTERRUPT_CLEAR_03_0
Offset: 0x15474

GPIO_E_INTERRUPT_CLEAR_00_0
Offset: 0x15614

GPIO_E_INTERRUPT_CLEAR_01_0
Offset: 0x15634

GPIO_E_INTERRUPT_CLEAR_02_0
Offset: 0x15654

GPIO_E_INTERRUPT_CLEAR_03_0
Offset: 0x15674

GPIO_E_INTERRUPT_CLEAR_04_0
Offset: 0x15694

GPIO_E_INTERRUPT_CLEAR_05_0
Offset: 0x156b4

GPIO_E_INTERRUPT_CLEAR_06_0
Offset: 0x156d4

GPIO_E_INTERRUPT_CLEAR_07_0
Offset: 0x156f4

GPIO_F_INTERRUPT_CLEAR_00_0
Offset: 0x15814

GPIO_F_INTERRUPT_CLEAR_01_0
Offset: 0x15834

GPIO_F_INTERRUPT_CLEAR_02_0
Offset: 0x15854

GPIO_F_INTERRUPT_CLEAR_03_0
Offset: 0x15874

GPIO_F_INTERRUPT_CLEAR_04_0
Offset: 0x15894

GPIO_F_INTERRUPT_CLEAR_05_0
Offset: 0x158b4

GPIO_EE_INTERRUPT_CLEAR_00_0
Offset: 0x1014

GPIO_EE_INTERRUPT_CLEAR_01_0
Offset: 0x1034

GPIO_EE_INTERRUPT_CLEAR_02_0
Offset: 0x1054

GPIO_EE_INTERRUPT_CLEAR_03_0
Offset: 0x1074

GPIO_EE_INTERRUPT_CLEAR_04_0
Offset: 0x1094

GPIO_EE_INTERRUPT_CLEAR_05_0
Offset: 0x10b4

GPIO_EE_INTERRUPT_CLEAR_06_0
Offset: 0x10d4

GPIO_EE_INTERRUPT_CLEAR_07_0
Offset: 0x10f4

GPIO_GG_INTERRUPT_CLEAR_00_0
Offset: 0x1214

GPIO_CC_INTERRUPT_CLEAR_00_0
Offset: 0x1414

GPIO_CC_INTERRUPT_CLEAR_01_0
Offset: 0x1434

GPIO_CC_INTERRUPT_CLEAR_02_0
Offset: 0x1454

GPIO_CC_INTERRUPT_CLEAR_03_0
Offset: 0x1474

GPIO_CC_INTERRUPT_CLEAR_04_0
Offset: 0x1494

GPIO_CC_INTERRUPT_CLEAR_05_0
Offset: 0x14b4

GPIO_CC_INTERRUPT_CLEAR_06_0
Offset: 0x14d4

GPIO_CC_INTERRUPT_CLEAR_07_0
Offset: 0x14f4

GPIO_DD_INTERRUPT_CLEAR_00_0
Offset: 0x1614

GPIO_DD_INTERRUPT_CLEAR_01_0
Offset: 0x1634

GPIO_DD_INTERRUPT_CLEAR_02_0
Offset: 0x1654

GPIO_AA_INTERRUPT_CLEAR_00_0
Offset: 0x1814

GPIO_AA_INTERRUPT_CLEAR_01_0
Offset: 0x1834

GPIO_AA_INTERRUPT_CLEAR_02_0

Offset: 0x1854

GPIO_AA_INTERRUPT_CLEAR_03_0
Offset: 0x1874

GPIO_AA_INTERRUPT_CLEAR_04_0
Offset: 0x1894

GPIO_AA_INTERRUPT_CLEAR_05_0
Offset: 0x18b4

GPIO_AA_INTERRUPT_CLEAR_06_0
Offset: 0x18d4

GPIO_AA_INTERRUPT_CLEAR_07_0
Offset: 0x18f4

GPIO_BB_INTERRUPT_CLEAR_00_0
Offset: 0x1a14

GPIO_BB_INTERRUPT_CLEAR_01_0
Offset: 0x1a34

GPIO_BB_INTERRUPT_CLEAR_02_0
Offset: 0x1a54

GPIO_BB_INTERRUPT_CLEAR_03_0
Offset: 0x1a74

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GPIO_<*i*>_SCR_0<*j*>_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**GPIO_INTERRUPT_CLEAR**: This field holds the Interrupt clear control bit for clearing the Interrupt at GPIO Pin <*j*> in GPIO Port <*i*>/<*iii*>. It is effective only when ((GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0. (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.INTERRUPT_FUNCTION == ENABLE)). **0** = DON'T CLEAR. **1** = CLEAR.

GPIO_<*i*>_INTERRUPT_STATUS_G<*jj*>_0,

where <*i*> = A, B, C, D, E, F, G, H, I, J, K, L, M, N, P, Q, R, X, Y, Z, AA, AC, AD, AE, AF, AG, BB, CC, DD, EE, GG and

<*jj*> = 0, 1, 2, 3, 4, 5, 6, 7.

GPIO_A_INTERRUPT_STATUS_G0_0
Offset: 0x10100

GPIO_A_INTERRUPT_STATUS_G1_0
Offset: 0x10104

GPIO_A_INTERRUPT_STATUS_G2_0
Offset: 0x10108

GPIO_A_INTERRUPT_STATUS_G3_0
Offset: 0x1010c

GPIO_A_INTERRUPT_STATUS_G4_0
Offset: 0x10110

GPIO_A_INTERRUPT_STATUS_G5_0
Offset: 0x10114

GPIO_A_INTERRUPT_STATUS_G6_0
Offset: 0x10118

GPIO_A_INTERRUPT_STATUS_G7_0
Offset: 0x1011c

GPIO_AC_INTERRUPT_STATUS_G0_0
Offset: 0x10300

GPIO_AC_INTERRUPT_STATUS_G1_0
Offset: 0x10304

GPIO_AC_INTERRUPT_STATUS_G2_0
Offset: 0x10308

GPIO_AC_INTERRUPT_STATUS_G3_0
Offset: 0x1030c

GPIO_AC_INTERRUPT_STATUS_G4_0
Offset: 0x10310

GPIO_AC_INTERRUPT_STATUS_G5_0
Offset: 0x10314

GPIO_AC_INTERRUPT_STATUS_G6_0
Offset: 0x10318

GPIO_AC_INTERRUPT_STATUS_G7_0
Offset: 0x1031c

GPIO_X_INTERRUPT_STATUS_G0_0
Offset: 0x11100

GPIO_X_INTERRUPT_STATUS_G1_0
Offset: 0x11104

GPIO_X_INTERRUPT_STATUS_G2_0
Offset: 0x11108

GPIO_X_INTERRUPT_STATUS_G3_0
Offset: 0x1110c

GPIO_X_INTERRUPT_STATUS_G4_0
Offset: 0x11110

GPIO_X_INTERRUPT_STATUS_G5_0
Offset: 0x11114

GPIO_X_INTERRUPT_STATUS_G6_0
Offset: 0x11118

GPIO_X_INTERRUPT_STATUS_G7_0
Offset: 0x1111c

GPIO_Y_INTERRUPT_STATUS_G0_0
Offset: 0x11300

GPIO_Y_INTERRUPT_STATUS_G1_0
Offset: 0x11304

GPIO_Y_INTERRUPT_STATUS_G2_0
Offset: 0x11308

GPIO_Y_INTERRUPT_STATUS_G3_0
Offset: 0x1130c

GPIO_Y_INTERRUPT_STATUS_G4_0
Offset: 0x11310

GPIO_Y_INTERRUPT_STATUS_G5_0
Offset: 0x11314

GPIO_Y_INTERRUPT_STATUS_G6_0
Offset: 0x11318

GPIO_Y_INTERRUPT_STATUS_G7_0
Offset: 0x1131c

GPIO_Z_INTERRUPT_STATUS_G0_0
Offset: 0x11500

GPIO_Z_INTERRUPT_STATUS_G1_0
Offset: 0x11504

GPIO_Z_INTERRUPT_STATUS_G2_0
Offset: 0x11508

GPIO_Z_INTERRUPT_STATUS_G3_0
Offset: 0x1150c

GPIO_Z_INTERRUPT_STATUS_G4_0
Offset: 0x11510

GPIO_Z_INTERRUPT_STATUS_G5_0
Offset: 0x11514

GPIO_Z_INTERRUPT_STATUS_G6_0
Offset: 0x11518

GPIO_Z_INTERRUPT_STATUS_G7_0
Offset: 0x1151c

GPIO_M_INTERRUPT_STATUS_G0_0
Offset: 0x12100

GPIO_M_INTERRUPT_STATUS_G1_0
Offset: 0x12104

GPIO_M_INTERRUPT_STATUS_G2_0
Offset: 0x12108

GPIO_M_INTERRUPT_STATUS_G3_0
Offset: 0x1210c

GPIO_M_INTERRUPT_STATUS_G4_0
Offset: 0x12110

GPIO_M_INTERRUPT_STATUS_G5_0
Offset: 0x12114

GPIO_M_INTERRUPT_STATUS_G6_0
Offset: 0x12118

GPIO_M_INTERRUPT_STATUS_G7_0
Offset: 0x1211c

GPIO_N_INTERRUPT_STATUS_G0_0
Offset: 0x12300

GPIO_N_INTERRUPT_STATUS_G1_0
Offset: 0x12304

GPIO_N_INTERRUPT_STATUS_G2_0
Offset: 0x12308

GPIO_N_INTERRUPT_STATUS_G3_0
Offset: 0x1230c

GPIO_N_INTERRUPT_STATUS_G4_0
Offset: 0x12310

GPIO_N_INTERRUPT_STATUS_G5_0
Offset: 0x12314

GPIO_N_INTERRUPT_STATUS_G6_0
Offset: 0x12318

GPIO_N_INTERRUPT_STATUS_G7_0
Offset: 0x1231c

GPIO_P_INTERRUPT_STATUS_G0_0
Offset: 0x12500

GPIO_P_INTERRUPT_STATUS_G1_0
Offset: 0x12504

GPIO_P_INTERRUPT_STATUS_G2_0
Offset: 0x12508

GPIO_P_INTERRUPT_STATUS_G3_0
Offset: 0x1250c

GPIO_P_INTERRUPT_STATUS_G4_0
Offset: 0x12510

GPIO_P_INTERRUPT_STATUS_G5_0
Offset: 0x12514

GPIO_P_INTERRUPT_STATUS_G6_0
Offset: 0x12518

GPIO_P_INTERRUPT_STATUS_G7_0
Offset: 0x1251c

GPIO_Q_INTERRUPT_STATUS_G0_0
Offset: 0x12700

GPIO_Q_INTERRUPT_STATUS_G1_0
Offset: 0x12704

GPIO_Q_INTERRUPT_STATUS_G2_0
Offset: 0x12708

GPIO_Q_INTERRUPT_STATUS_G3_0
Offset: 0x1270c

GPIO_Q_INTERRUPT_STATUS_G4_0
Offset: 0x12710

GPIO_Q_INTERRUPT_STATUS_G5_0
Offset: 0x12714

GPIO_Q_INTERRUPT_STATUS_G6_0
Offset: 0x12718

GPIO_Q_INTERRUPT_STATUS_G7_0
Offset: 0x1271c

GPIO_K_INTERRUPT_STATUS_G0_0
Offset: 0x13100

GPIO_K_INTERRUPT_STATUS_G1_0
Offset: 0x13104

GPIO_K_INTERRUPT_STATUS_G2_0
Offset: 0x13108

GPIO_K_INTERRUPT_STATUS_G3_0
Offset: 0x1310c

GPIO_K_INTERRUPT_STATUS_G4_0
Offset: 0x13110

GPIO_K_INTERRUPT_STATUS_G5_0
Offset: 0x13114

GPIO_K_INTERRUPT_STATUS_G6_0
Offset: 0x13118

GPIO_K_INTERRUPT_STATUS_G7_0
Offset: 0x1311c

GPIO_AG_INTERRUPT_STATUS_G0_0
Offset: 0x13500

GPIO_AG_INTERRUPT_STATUS_G1_0
Offset: 0x13504

GPIO_AG_INTERRUPT_STATUS_G2_0
Offset: 0x13508

GPIO_AG_INTERRUPT_STATUS_G3_0
Offset: 0x1350c

GPIO_AG_INTERRUPT_STATUS_G4_0
Offset: 0x13510

GPIO_AG_INTERRUPT_STATUS_G5_0
Offset: 0x13514

GPIO_AG_INTERRUPT_STATUS_G6_0
Offset: 0x13518

GPIO_AG_INTERRUPT_STATUS_G7_0
Offset: 0x1351c

GPIO_G_INTERRUPT_STATUS_G0_0
Offset: 0x14100

GPIO_G_INTERRUPT_STATUS_G1_0
Offset: 0x14104

GPIO_G_INTERRUPT_STATUS_G2_0
Offset: 0x14108

GPIO_G_INTERRUPT_STATUS_G3_0
Offset: 0x1410c

GPIO_G_INTERRUPT_STATUS_G4_0
Offset: 0x14110

GPIO_G_INTERRUPT_STATUS_G5_0
Offset: 0x14114

GPIO_G_INTERRUPT_STATUS_G6_0
Offset: 0x14118

GPIO_G_INTERRUPT_STATUS_G7_0
Offset: 0x1411c

GPIO_H_INTERRUPT_STATUS_G0_0
Offset: 0x14300

GPIO_H_INTERRUPT_STATUS_G1_0
Offset: 0x14304

GPIO_H_INTERRUPT_STATUS_G2_0
Offset: 0x14308

GPIO_H_INTERRUPT_STATUS_G3_0
Offset: 0x1430c

GPIO_H_INTERRUPT_STATUS_G4_0
Offset: 0x14310

GPIO_H_INTERRUPT_STATUS_G5_0
Offset: 0x14314

GPIO_H_INTERRUPT_STATUS_G6_0
Offset: 0x14318

GPIO_H_INTERRUPT_STATUS_G7_0
Offset: 0x1431c

GPIO_C_INTERRUPT_STATUS_G0_0
Offset: 0x15300

GPIO_C_INTERRUPT_STATUS_G1_0
Offset: 0x15304

GPIO_C_INTERRUPT_STATUS_G2_0
Offset: 0x15308

GPIO_C_INTERRUPT_STATUS_G3_0
Offset: 0x1530c

GPIO_C_INTERRUPT_STATUS_G4_0
Offset: 0x15310

GPIO_C_INTERRUPT_STATUS_G5_0
Offset: 0x15314

GPIO_C_INTERRUPT_STATUS_G6_0
Offset: 0x15318

GPIO_C_INTERRUPT_STATUS_G7_0
Offset: 0x1531c

GPIO_E_INTERRUPT_STATUS_G0_0
Offset: 0x15700

GPIO_E_INTERRUPT_STATUS_G1_0
Offset: 0x15704

GPIO_E_INTERRUPT_STATUS_G2_0
Offset: 0x15708

GPIO_E_INTERRUPT_STATUS_G3_0
Offset: 0x1570c

GPIO_E_INTERRUPT_STATUS_G4_0
Offset: 0x15710

GPIO_E_INTERRUPT_STATUS_G5_0
Offset: 0x15714

GPIO_E_INTERRUPT_STATUS_G6_0
Offset: 0x15718

GPIO_E_INTERRUPT_STATUS_G7_0
Offset: 0x1571c

GPIO_EE_INTERRUPT_STATUS_G0_0
Offset: 0x1100

GPIO_EE_INTERRUPT_STATUS_G1_0
Offset: 0x1104

GPIO_EE_INTERRUPT_STATUS_G2_0
Offset: 0x1108

GPIO_EE_INTERRUPT_STATUS_G3_0
Offset: 0x110c

GPIO_EE_INTERRUPT_STATUS_G4_0
Offset: 0x1110

GPIO_EE_INTERRUPT_STATUS_G5_0
Offset: 0x1114

GPIO_EE_INTERRUPT_STATUS_G6_0
Offset: 0x1118

GPIO_EE_INTERRUPT_STATUS_G7_0
Offset: 0x111c

GPIO_CC_INTERRUPT_STATUS_G0_0
Offset: 0x1500

GPIO_CC_INTERRUPT_STATUS_G1_0
Offset: 0x1504

GPIO_CC_INTERRUPT_STATUS_G2_0
Offset: 0x1508

GPIO_CC_INTERRUPT_STATUS_G3_0
Offset: 0x150c

GPIO_CC_INTERRUPT_STATUS_G4_0
Offset: 0x1510

GPIO_CC_INTERRUPT_STATUS_G5_0
Offset: 0x1514

GPIO_CC_INTERRUPT_STATUS_G6_0
Offset: 0x1518

GPIO_CC_INTERRUPT_STATUS_G7_0
Offset: 0x151c

GPIO_AA_INTERRUPT_STATUS_G0_0
Offset: 0x1900

GPIO_AA_INTERRUPT_STATUS_G1_0
Offset: 0x1904

GPIO_AA_INTERRUPT_STATUS_G2_0
Offset: 0x1908

GPIO_AA_INTERRUPT_STATUS_G3_0
Offset: 0x190c

GPIO_AA_INTERRUPT_STATUS_G4_0
Offset: 0x1910

GPIO_AA_INTERRUPT_STATUS_G5_0
Offset: 0x1914

GPIO_AA_INTERRUPT_STATUS_G6_0
Offset: 0x1918

GPIO_AA_INTERRUPT_STATUS_G7_0
Offset: 0x191c

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**GPIO_INTERRUPT_STATUS**: This field has as many bits as the number of GPIO Pins in the given GPIO Port, with each such bit indicating whether the corresponding GPIO Input change has resulted in Interrupt. Bit 0 is for Pin 0, Bit 1 is for Pin 1, ... Bit <*j*> is for Pin <*j*>. **0** = Interrupt not set. **1** = Interrupt set. It is effective only when ((GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE) && (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.INTERRUPT_FUNCTION == ENABLE)). The bit in this field is cleared based on GPIO_<*i*>/ <*iii*>_INTERRUPT_CLEAR_0<*j*>_0.GPIO_INTERRUPT_CLEAR. In a Virtualized environment, the status reflects only the GPIOs belonging to VM. For the GPIO bit positions not mapped to VM, the corresponding bit in this field is written with "0" by hardware for software to read this register correctly.

GPIO_I_INTERRUPT_STATUS_G0_0
Offset: 0x14500

GPIO_I_INTERRUPT_STATUS_G1_0
Offset: 0x14504

GPIO_I_INTERRUPT_STATUS_G2_0
Offset: 0x14508

GPIO_I_INTERRUPT_STATUS_G3_0
Offset: 0x1450c

GPIO_I_INTERRUPT_STATUS_G4_0
Offset: 0x14510

GPIO_I_INTERRUPT_STATUS_G5_0
Offset: 0x14514

GPIO_I_INTERRUPT_STATUS_G6_0
Offset: 0x14518

GPIO_I_INTERRUPT_STATUS_G7_0
Offset: 0x1451c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000,0000)

Bit	Reset	Description
6:0	0x0	**GPIO_INTERRUPT_STATUS**: This field has as many bits as the number of GPIO Pins in the given GPIO Port, with each such bit indicating whether the corresponding GPIO Input change has resulted in Interrupt. Bit 0 is for Pin 0, Bit 1 is for Pin 1, ... Bit <*j*> is for Pin <*j*>. **0** = Interrupt not set. **1** = Interrupt set. It is effective only when ((GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE) && (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.INTERRUPT_FUNCTION == ENABLE)). The bit in this field is cleared based on GPIO_<*i*>/ <*iii*>_INTERRUPT_CLEAR_0<*j*>_0.GPIO_INTERRUPT_CLEAR. In a Virtualized environment, the status reflects only the GPIOs belonging to VM. For the GPIO bit positions not mapped to VM, the corresponding bit in this field is written with "0" by hardware for software to read this register correctly.

GPIO_R_INTERRUPT_STATUS_G0_0
Offset: 0x12900

GPIO_R_INTERRUPT_STATUS_G1_0
Offset: 0x12904

GPIO_R_INTERRUPT_STATUS_G2_0
Offset: 0x12908

GPIO_R_INTERRUPT_STATUS_G3_0
Offset: 0x1290c

GPIO_R_INTERRUPT_STATUS_G4_0
Offset: 0x12910

GPIO_R_INTERRUPT_STATUS_G5_0
Offset: 0x12914

GPIO_R_INTERRUPT_STATUS_G6_0
Offset: 0x12918

GPIO_R_INTERRUPT_STATUS_G7_0
Offset: 0x1291c

GPIO_J_INTERRUPT_STATUS_G0_0
Offset: 0x15100

GPIO_J_INTERRUPT_STATUS_G1_0
Offset: 0x15104

GPIO_J_INTERRUPT_STATUS_G2_0
Offset: 0x15108

GPIO_J_INTERRUPT_STATUS_G3_0
Offset: 0x1510c

GPIO_J_INTERRUPT_STATUS_G4_0
Offset: 0x15110

GPIO_J_INTERRUPT_STATUS_G5_0
Offset: 0x15114

GPIO_J_INTERRUPT_STATUS_G6_0
Offset: 0x15118

GPIO_J_INTERRUPT_STATUS_G7_0
Offset: 0x1511c

GPIO_F_INTERRUPT_STATUS_G0_0
Offset: 0x15900

GPIO_F_INTERRUPT_STATUS_G1_0
Offset: 0x15904

GPIO_F_INTERRUPT_STATUS_G2_0
Offset: 0x15908

GPIO_F_INTERRUPT_STATUS_G3_0
Offset: 0x1590c

GPIO_F_INTERRUPT_STATUS_G4_0
Offset: 0x15910

GPIO_F_INTERRUPT_STATUS_G5_0
Offset: 0x15914

GPIO_F_INTERRUPT_STATUS_G6_0
Offset: 0x15918

GPIO_F_INTERRUPT_STATUS_G7_0
Offset: 0x1591c

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**GPIO_INTERRUPT_STATUS**: This field has as many bits as the number of GPIO Pins in the given GPIO Port, with each such bit indicating whether the corresponding GPIO Input change has resulted in Interrupt. Bit 0 is for Pin 0, Bit 1 is for Pin 1, ... Bit <*j*> is for Pin <*j*>. **0** = Interrupt not set. **1** = Interrupt set. It is effective only when ((GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE) && (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.INTERRUPT_FUNCTION == ENABLE)). The bit in this field is cleared based on GPIO_<*i*>/ <*iii*>_INTERRUPT_CLEAR_0<*j*>_0.GPIO_INTERRUPT_CLEAR. In a Virtualized environment, the status reflects only the GPIOs belonging to VM. For the GPIO bit positions not mapped to VM, the corresponding bit in this field is written with "0" by hardware for software to read this register correctly.

GPIO_L_INTERRUPT_STATUS_G0_0
Offset: 0x13300

GPIO_L_INTERRUPT_STATUS_G1_0
Offset: 0x13304

GPIO_L_INTERRUPT_STATUS_G2_0
Offset: 0x13308

GPIO_L_INTERRUPT_STATUS_G3_0
Offset: 0x1330c

GPIO_L_INTERRUPT_STATUS_G4_0
Offset: 0x13310

GPIO_L_INTERRUPT_STATUS_G5_0
Offset: 0x13314

GPIO_L_INTERRUPT_STATUS_G6_0
Offset: 0x13318

GPIO_L_INTERRUPT_STATUS_G7_0
Offset: 0x1331c

GPIO_D_INTERRUPT_STATUS_G0_0
Offset: 0x15500

GPIO_D_INTERRUPT_STATUS_G1_0
Offset: 0x15504

GPIO_D_INTERRUPT_STATUS_G2_0
Offset: 0x15508

GPIO_D_INTERRUPT_STATUS_G3_0
Offset: 0x1550c

GPIO_D_INTERRUPT_STATUS_G4_0
Offset: 0x15510

GPIO_D_INTERRUPT_STATUS_G5_0
Offset: 0x15514

GPIO_D_INTERRUPT_STATUS_G6_0
Offset: 0x15518

GPIO_D_INTERRUPT_STATUS_G7_0
Offset: 0x1551c

GPIO_AF_INTERRUPT_STATUS_G0_0
Offset: 0x13900

GPIO_AF_INTERRUPT_STATUS_G1_0
Offset: 0x13904

GPIO_AF_INTERRUPT_STATUS_G2_0
Offset: 0x13908

GPIO_AF_INTERRUPT_STATUS_G3_0
Offset: 0x1390c

GPIO_AF_INTERRUPT_STATUS_G4_0
Offset: 0x13910

GPIO_AF_INTERRUPT_STATUS_G5_0
Offset: 0x13914

GPIO_AF_INTERRUPT_STATUS_G6_0
Offset: 0x13918

GPIO_AF_INTERRUPT_STATUS_G7_0
Offset: 0x1391c

GPIO_AD_INTERRUPT_STATUS_G0_0
Offset: 0x10500

GPIO_AD_INTERRUPT_STATUS_G1_0
Offset: 0x10504

GPIO_AD_INTERRUPT_STATUS_G2_0
Offset: 0x10508

GPIO_AD_INTERRUPT_STATUS_G3_0
Offset: 0x1050c

GPIO_AD_INTERRUPT_STATUS_G4_0
Offset: 0x10510

GPIO_AD_INTERRUPT_STATUS_G5_0
Offset: 0x10514

GPIO_AD_INTERRUPT_STATUS_G6_0
Offset: 0x10518

GPIO_AD_INTERRUPT_STATUS_G7_0
Offset: 0x1051c

GPIO_BB_INTERRUPT_STATUS_G0_0
Offset: 0x1b00

GPIO_BB_INTERRUPT_STATUS_G1_0
Offset: 0x1b04

GPIO_BB_INTERRUPT_STATUS_G2_0
Offset: 0x1b08

GPIO_BB_INTERRUPT_STATUS_G3_0
Offset: 0x1b0c

GPIO_BB_INTERRUPT_STATUS_G4_0
Offset: 0x1b10

GPIO_BB_INTERRUPT_STATUS_G5_0
Offset: 0x1b14

GPIO_BB_INTERRUPT_STATUS_G6_0
Offset: 0x1b18

GPIO_BB_INTERRUPT_STATUS_G7_0
Offset: 0x1b1c

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	0x0	**GPIO_INTERRUPT_STATUS**: This field has as many bits as the number of GPIO Pins in the given GPIO Port, with each such bit indicating whether the corresponding GPIO Input change has resulted in Interrupt. Bit 0 is for Pin 0, Bit 1 is for Pin 1, ... Bit <*j*> is for Pin <*j*>. **0** = Interrupt not set. **1** = Interrupt set. It is effective only when ((GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE) && (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.INTERRUPT_FUNCTION == ENABLE)). The bit in this field is cleared based on GPIO_<*i*>/ <*iii*>_INTERRUPT_CLEAR_0<*j*>_0.GPIO_INTERRUPT_CLEAR. In a Virtualized environment, the status reflects only the GPIOs belonging to VM. For the GPIO bit positions not mapped to VM, the corresponding bit in this field is written with "0" by hardware for software to read this register correctly.

GPIO_DD_INTERRUPT_STATUS_G0_0
Offset: 0x1700

GPIO_DD_INTERRUPT_STATUS_G1_0
Offset: 0x1704

GPIO_DD_INTERRUPT_STATUS_G2_0
Offset: 0x1708

GPIO_DD_INTERRUPT_STATUS_G3_0
Offset: 0x170c

GPIO_DD_INTERRUPT_STATUS_G4_0
Offset: 0x1710

GPIO_DD_INTERRUPT_STATUS_G5_0
Offset: 0x1714

GPIO_DD_INTERRUPT_STATUS_G6_0
Offset: 0x1718

GPIO_DD_INTERRUPT_STATUS_G7_0
Offset: 0x171c

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	0x0	**GPIO_INTERRUPT_STATUS**: This field has as many bits as the number of GPIO Pins in the given GPIO Port, with each such bit indicating whether the corresponding GPIO Input change has resulted in Interrupt. Bit 0 is for Pin 0, Bit 1 is for Pin 1, ... Bit <*j*> is for Pin <*j*>. **0** = Interrupt not set. **1** = Interrupt set. It is effective only when ((GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE) && (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.INTERRUPT_FUNCTION == ENABLE)). The bit in this field is cleared based on GPIO_<*i*>/ <*iii*>_INTERRUPT_CLEAR_0<*j*>_0.GPIO_INTERRUPT_CLEAR. In a Virtualized environment, the status reflects only the GPIOs belonging to VM. For the GPIO bit positions not mapped to VM, the corresponding bit in this field is written with "0" by hardware for software to read this register correctly.

GPIO_AE_INTERRUPT_STATUS_G0_0
Offset: 0x13700

GPIO_AE_INTERRUPT_STATUS_G1_0
Offset: 0x13704

GPIO_AE_INTERRUPT_STATUS_G2_0
Offset: 0x13708

GPIO_AE_INTERRUPT_STATUS_G3_0
Offset: 0x1370c

GPIO_AE_INTERRUPT_STATUS_G4_0
Offset: 0x13710

GPIO_AE_INTERRUPT_STATUS_G5_0
Offset: 0x13714

GPIO_AE_INTERRUPT_STATUS_G6_0
Offset: 0x13718

GPIO_AE_INTERRUPT_STATUS_G7_0
Offset: 0x1371c

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**GPIO_INTERRUPT_STATUS**: This field has as many bits as the number of GPIO Pins in the given GPIO Port, with each such bit indicating whether the corresponding GPIO Input change has resulted in Interrupt. Bit 0 is for Pin 0, Bit 1 is for Pin 1, ... Bit <*j*> is for Pin <*j*>. **0** = Interrupt not set. **1** = Interrupt set. It is effective only when ((GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE) && (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.INTERRUPT_FUNCTION == ENABLE)). The bit in this field is cleared based on GPIO_<*i*>/ <*iii*>_INTERRUPT_CLEAR_0<*j*>_0.GPIO_INTERRUPT_CLEAR. In a Virtualized environment, the status reflects only the GPIOs belonging to VM. For the GPIO bit positions not mapped to VM, the corresponding bit in this field is written with "0" by hardware for software to read this register correctly.

GPIO_B_INTERRUPT_STATUS_G0_0
Offset: 0x10700

GPIO_B_INTERRUPT_STATUS_G1_0
Offset: 0x10704

GPIO_B_INTERRUPT_STATUS_G2_0
Offset: 0x10708

GPIO_B_INTERRUPT_STATUS_G3_0
Offset: 0x1070c

GPIO_B_INTERRUPT_STATUS_G4_0
Offset: 0x10710

GPIO_B_INTERRUPT_STATUS_G5_0
Offset: 0x10714

GPIO_B_INTERRUPT_STATUS_G6_0
Offset: 0x10718

GPIO_B_INTERRUPT_STATUS_G7_0
Offset: 0x1071c

GPIO_GG_INTERRUPT_STATUS_G0_0
Offset: 0x1300

GPIO_GG_INTERRUPT_STATUS_G1_0
Offset: 0x1304

GPIO_GG_INTERRUPT_STATUS_G2_0
Offset: 0x1308

GPIO_GG_INTERRUPT_STATUS_G3_0
Offset: 0x130c

GPIO_GG_INTERRUPT_STATUS_G4_0
Offset: 0x1310

GPIO_GG_INTERRUPT_STATUS_G5_0
Offset: 0x1314

GPIO_GG_INTERRUPT_STATUS_G6_0
Offset: 0x1318

GPIO_GG_INTERRUPT_STATUS_G7_0
Offset: 0x131c

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**GPIO_INTERRUPT_STATUS**: This field has as many bits as the number of GPIO Pins in the given GPIO Port, with each such bit indicating whether the corresponding GPIO Input change has resulted in Interrupt. Bit 0 is for Pin 0, Bit 1 is for Pin 1, ... Bit <*j*> is for Pin <*j*>. **0** = Interrupt not set. **1** = Interrupt set. It is effective only when ((GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE) && (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.INTERRUPT_FUNCTION == ENABLE)). The bit in this field is cleared based on GPIO_<*i*>/ <*iii*>_INTERRUPT_CLEAR_0<*j*>_0.GPIO_INTERRUPT_CLEAR. In a Virtualized environment, the status reflects only the GPIOs belonging to VM. For the GPIO bit positions not mapped to VM, the corresponding bit in this field is written with "0" by hardware for software to read this register correctly.

8.5.7 GPIO Common Control Registers

NOTE:

All Non-AON Common Control Registers, i.e. GPIO_<*ii*>,

where <*ii*> = A, B, C, D, E, F, G, H, I, J, K, L, M, N, P, Q, R, X, Y, Z, AC, AD, AE, AF, AG,

share the same base address under the Block Name GPI_CTL_COMMON,

All AON Common Control Registers, i.e. GPIO_<*iii*>,

where <*iii*> = AA, BB, CC, DD, EE, and GG,

share the same base address under the Block Name AON_GPIO_0,

For the base address of Non-AON GPIO Common Control Registers (GPI_CTL_COMMON) and AON GPIO Common Control Registers (AON_GPIO_0), please refer to the System Address Map in this TRM.

NOTE:

The binary "x" is turned to "0" for the Reset and PROD hex value. Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as "x" or "Undefined" actually means that the bit could be either "0" or "1" after Reset, whereas a bit PROD value as "x" or "Don't-care" means that either "0" or "1" can be written to the bit by software Initialization.

GPIO_<*ii*>_VM_0<*j*>_0,

where <*ii*> = A, B, C, D, E, F, G, H, I, J, K, L, M, N, P, Q, R, X, Y, Z, AC, AD, AE, AF, AG and

<*j*> ∈ {0, 1, 2, 3, 4, 5, 6, 7}.

<*kk*> = 0, when <*ii*> = A, B, AC, AD,
= 1, when <*ii*> = X, Y, Z,
= 2, when <*ii*> = M, N P, Q, R,
= 3, when <*ii*> = K, , AE, AF, AG,
= 4, when <*ii*> = G, H, I,
= 5, when <*ii*> = C, D, E, F, J.

GPIO_A_VM_00_0
Offset: 0x0

GPIO_A_VM_01_0
Offset: 0x8

GPIO_A_VM_02_0
Offset: 0x10

GPIO_A_VM_03_0
Offset: 0x18

GPIO_A_VM_04_0
Offset: 0x20

GPIO_A_VM_05_0
Offset: 0x28

GPIO_A_VM_06_0
Offset: 0x30

GPIO_A_VM_07_0
Offset: 0x38

GPIO_AC_VM_00_0
Offset: 0x40

GPIO_AC_VM_01_0
Offset: 0x48

GPIO_AC_VM_02_0
Offset: 0x50

GPIO_AC_VM_03_0
Offset: 0x58

GPIO_AC_VM_04_0
Offset: 0x60

GPIO_AC_VM_05_0
Offset: 0x68

GPIO_AC_VM_06_0
Offset: 0x70

GPIO_AC_VM_07_0
Offset: 0x78

GPIO_AD_VM_00_0
Offset: 0x80

GPIO_AD_VM_01_0
Offset: 0x88

GPIO_AD_VM_02_0
Offset: 0x90

GPIO_AD_VM_03_0
Offset: 0x98

GPIO_B_VM_00_0
Offset: 0xc0

GPIO_X_VM_00_0
Offset: 0x1000

GPIO_X_VM_01_0
Offset: 0x1008

GPIO_X_VM_02_0
Offset: 0x1010

GPIO_X_VM_03_0
Offset: 0x1018

GPIO_X_VM_04_0
Offset: 0x1020

GPIO_X_VM_05_0
Offset: 0x1028

GPIO_X_VM_06_0
Offset: 0x1030

GPIO_X_VM_07_0
Offset: 0x1038

GPIO_Y_VM_00_0
Offset: 0x1040

GPIO_Y_VM_01_0
Offset: 0x1048

GPIO_Y_VM_02_0
Offset: 0x1050

GPIO_Y_VM_03_0
Offset: 0x1058

GPIO_Y_VM_04_0
Offset: 0x1060

GPIO_Y_VM_05_0
Offset: 0x1068

GPIO_Y_VM_06_0
Offset: 0x1070

GPIO_Y_VM_07_0
Offset: 0x1078

GPIO_Z_VM_00_0
Offset: 0x1080

GPIO_Z_VM_01_0
Offset: 0x1088

GPIO_Z_VM_02_0
Offset: 0x1090

GPIO_Z_VM_03_0
Offset: 0x1098

GPIO_Z_VM_04_0
Offset: 0x10a0

GPIO_Z_VM_05_0
Offset: 0x10a8

GPIO_Z_VM_06_0
Offset: 0x10b0

GPIO_Z_VM_07_0
Offset: 0x10b8

GPIO_M_VM_00_0
Offset: 0x2000

GPIO_M_VM_01_0
Offset: 0x2008

GPIO_M_VM_02_0
Offset: 0x2010

GPIO_M_VM_03_0
Offset: 0x2018

GPIO_M_VM_04_0
Offset: 0x2020

GPIO_M_VM_05_0
Offset: 0x2028

GPIO_M_VM_06_0
Offset: 0x2030

GPIO_M_VM_07_0
Offset: 0x2038

GPIO_N_VM_00_0
Offset: 0x2040

GPIO_N_VM_01_0
Offset: 0x2048

GPIO_N_VM_02_0
Offset: 0x2050

GPIO_N_VM_03_0
Offset: 0x2058

GPIO_N_VM_04_0
Offset: 0x2060

GPIO_N_VM_05_0
Offset: 0x2068

GPIO_N_VM_06_0
Offset: 0x2070

GPIO_N_VM_07_0
Offset: 0x2078

GPIO_P_VM_00_0
Offset: 0x2080

GPIO_P_VM_01_0
Offset: 0x2088

GPIO_P_VM_02_0
Offset: 0x2090

GPIO_P_VM_03_0
Offset: 0x2098

GPIO_P_VM_04_0
Offset: 0x20a0

GPIO_P_VM_05_0
Offset: 0x20a8

GPIO_P_VM_06_0
Offset: 0x20b0

GPIO_P_VM_07_0
Offset: 0x20b8

GPIO_Q_VM_00_0
Offset: 0x20c0

GPIO_Q_VM_01_0
Offset: 0x20c8

GPIO_Q_VM_02_0
Offset: 0x20d0

GPIO_Q_VM_03_0
Offset: 0x20d8

GPIO_Q_VM_04_0
Offset: 0x20e0

GPIO_Q_VM_05_0
Offset: 0x20e8

GPIO_Q_VM_06_0
Offset: 0x20f0

GPIO_Q_VM_07_0
Offset: 0x20f8

GPIO_R_VM_00_0
Offset: 0x2100

GPIO_R_VM_01_0
Offset: 0x2108

GPIO_R_VM_02_0
Offset: 0x2110

GPIO_R_VM_03_0
Offset: 0x2118

GPIO_R_VM_04_0
Offset: 0x2120

GPIO_R_VM_05_0
Offset: 0x2128

GPIO_K_VM_00_0
Offset: 0x3000

GPIO_K_VM_01_0
Offset: 0x3008

GPIO_K_VM_02_0
Offset: 0x3010

GPIO_K_VM_03_0
Offset: 0x3018

GPIO_K_VM_04_0
Offset: 0x3020

GPIO_K_VM_05_0
Offset: 0x3028

GPIO_K_VM_06_0
Offset: 0x3030

GPIO_K_VM_07_0
Offset: 0x3038

GPIO_L_VM_00_0
Offset: 0x3040

GPIO_L_VM_01_0
Offset: 0x3048

GPIO_L_VM_02_0
Offset: 0x3050

GPIO_L_VM_03_0
Offset: 0x3058

GPIO_AG_VM_00_0
Offset: 0x3080

GPIO_AG_VM_01_0
Offset: 0x3088

GPIO_AG_VM_02_0
Offset: 0x3090

GPIO_AG_VM_03_0
Offset: 0x3098

GPIO_AG_VM_04_0
Offset: 0x30a0

GPIO_AG_VM_05_0
Offset: 0x30a8

GPIO_AG_VM_06_0
Offset: 0x30b0

GPIO_AG_VM_07_0
Offset: 0x30b8

GPIO_AE_VM_00_0
Offset: 0x30c0

GPIO_AE_VM_01_0
Offset: 0x30c8

GPIO_AF_VM_00_0
Offset: 0x3100

GPIO_AF_VM_01_0
Offset: 0x3108

GPIO_AF_VM_02_0
Offset: 0x3110

GPIO_AF_VM_03_0
Offset: 0x3118

GPIO_G_VM_00_0
Offset: 0x4000

GPIO_G_VM_01_0
Offset: 0x4008

GPIO_G_VM_02_0
Offset: 0x4010

GPIO_G_VM_03_0
Offset: 0x4018

GPIO_G_VM_04_0
Offset: 0x4020

GPIO_G_VM_05_0
Offset: 0x4028

GPIO_G_VM_06_0
Offset: 0x4030

GPIO_G_VM_07_0
Offset: 0x4038

GPIO_H_VM_00_0
Offset: 0x4040

GPIO_H_VM_01_0
Offset: 0x4048

GPIO_H_VM_02_0
Offset: 0x4050

GPIO_H_VM_03_0
Offset: 0x4058

GPIO_H_VM_04_0
Offset: 0x4060

GPIO_H_VM_05_0
Offset: 0x4068

GPIO_H_VM_06_0
Offset: 0x4070

GPIO_H_VM_07_0
Offset: 0x4078

GPIO_I_VM_00_0
Offset: 0x4080

GPIO_I_VM_01_0
Offset: 0x4088

GPIO_I_VM_02_0
Offset: 0x4090

GPIO_I_VM_03_0
Offset: 0x4098

GPIO_I_VM_04_0
Offset: 0x40a0

GPIO_I_VM_05_0
Offset: 0x40a8

GPIO_I_VM_06_0
Offset: 0x40b0

GPIO_J_VM_00_0
Offset: 0x5000

GPIO_J_VM_01_0
Offset: 0x5008

GPIO_J_VM_02_0
Offset: 0x5010

GPIO_J_VM_03_0
Offset: 0x5018

GPIO_J_VM_04_0
Offset: 0x5020

GPIO_J_VM_05_0
Offset: 0x5028

GPIO_C_VM_00_0
Offset: 0x5040

GPIO_C_VM_01_0
Offset: 0x5048

GPIO_C_VM_02_0
Offset: 0x5050

GPIO_C_VM_03_0
Offset: 0x5058

GPIO_C_VM_04_0
Offset: 0x5060

GPIO_C_VM_05_0
Offset: 0x5068

GPIO_C_VM_06_0
Offset: 0x5070

GPIO_C_VM_07_0
Offset: 0x5078

GPIO_D_VM_00_0
Offset: 0x5080

GPIO_D_VM_01_0
Offset: 0x5088

GPIO_D_VM_02_0
Offset: 0x5090

GPIO_D_VM_03_0
Offset: 0x5098

GPIO_E_VM_00_0
Offset: 0x50c0

GPIO_E_VM_01_0
Offset: 0x50c8

GPIO_E_VM_02_0
Offset: 0x50d0

GPIO_E_VM_03_0
Offset: 0x50d8

GPIO_E_VM_04_0
Offset: 0x50e0

GPIO_E_VM_05_0
Offset: 0x50e8

GPIO_E_VM_06_0
Offset: 0x50f0

GPIO_E_VM_07_0
Offset: 0x50f8

GPIO_E_VM_07_0
Offset: 0x50f8

GPIO_F_VM_00_0
Offset: 0x5100

GPIO_F_VM_01_0
Offset: 0x5108

GPIO_F_VM_02_0
Offset: 0x5110

GPIO_F_VM_03_0
Offset: 0x5118

GPIO_F_VM_04_0
Offset: 0x5120

GPIO_F_VM_05_0
Offset: 0x5128

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: GPIO_CTL<*kk*>_VM_SCR_0
Reset: 0x0000ffff (0bxxxx,xxxx,xxxx,xxxx,1111,1111,1111,1111)

Bit	Reset	Description
15:14	0x3	**VM8**: This field specifies the access permission for VM8 (Guest OS7). **00** = No Access for VM8 (Guest OS7). **01** = Read Only Access for VM8 (Guest OS7). **10** = Write Only Access for VM8 (Guest OS7). **11** = FULL access for VM8 (Guest OS7).
13:12	0x3	**VM7**: This field specifies the access permission for VM7 (Guest OS6). **00** = No Access for VM7 (Guest OS6). **01** = Read Only Access for VM7 (Guest OS6). **10** = Write Only Access for VM7 (Guest OS6). **11** = FULL access for VM7 (Guest OS6).
11:10	0x3	**VM6**: This field specifies the access permission for VM6 (Guest OS5). **00** = No Access for VM6 (Guest OS5). **01** = Read Only Access for VM6 (Guest OS5). **10** = Write Only Access for VM6 (Guest OS5). **11** = FULL access for VM6 (Guest OS5).
9:8	0x3	**VM5**: This field specifies the access permission for VM5 (Guest OS4). **00** = No Access for VM5 (Guest OS4). **01** = Read Only Access for VM5 (Guest OS4). **10** = Write Only Access for VM5 (Guest OS4). **11** = FULL access for VM5 (Guest OS4).
7:6	0x3	**VM4**: This field specifies the access permission for VM4 (Guest OS3). **00** = No Access for VM4 (Guest OS3). **01** = Read Only Access for VM4 (Guest OS3). **10** = Write Only Access for VM4 (Guest OS3). **11** = FULL access for VM4 (Guest OS3).
5:4	0x3	**VM3**: This field specifies the access permission for VM3 (Guest OS2). **00** = No Access for VM3 (Guest OS2). **01** = Read Only Access for VM3 (Guest OS2). **10** = Write Only Access for VM3 (Guest OS2). **11** = FULL access for VM3 (Guest OS2).
3:2	0x3	**VM2**: This field specifies the access permission for VM2 (Guest OS1). **00** = No Access for VM2 (Guest OS1). **01** = Read Only Access for VM2 (Guest OS1). **10** = Write Only Access for VM2 (Guest OS1). **11** = FULL access for VM2 (Guest OS1).

Bit	Reset	Description
1:0	0x3	**VM1**: This field specifies the access permission for VM1 (Guest OS0). **00** = No Access for VM1 (Guest OS0). **01** = Read Only Access for VM1 (Guest OS0). **10** = Write Only Access for VM1 (Guest OS0). **11** = FULL access for VM1 (Guest OS0).

GPIO_CTL<*kk*>_INTERRUPT_STATUS_VM_0,

where <*kk*> = 0, 1, 2, 3, 4, 5.

GPIO_CTL0_INTERRUPT_STATUS_VM_0
Offset: 0x808

GPIO_CTL1_INTERRUPT_STATUS_VM_0
Offset: 0x1808

GPIO_CTL2_INTERRUPT_STATUS_VM_0
Offset: 0x2808

GPIO_CTL3_INTERRUPT_STATUS_VM_0
Offset: 0x3808

GPIO_CTL4_INTERRUPT_STATUS_VM_0
Offset: 0x4808

GPIO_CTL5_INTERRUPT_STATUS_VM_0
Offset: 0x5808

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: GPIO_CTL<*kk*>_INTERRUPT_STATUS_VM_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**GPIO_INTERRUPT_STATUS_VM**: Each bit of this field Indicates the Interrupt Status whether any GPIO owned by a specific Guest OS resulted an Interrupt. Bit 0 is for VM1 (Guest OS0), Bit 1 is for VM2 (Guest OS1), Bit 2 is for VM3 (Guest OS2), Bit 3 is for VM4 (Guest OS3), Bit 4 is for VM5 (Guest OS4), Bit 5 is for VM6 (Guest OS5), Bit 6 is for VM7 (Guest OS6), Bit 7 is for VM8 (Guest OS7). **0** = Interrupt not set. **1** = Interrupt set. The bit in this field is cleared based on GPIO_<*i*>/<*iii*>_INTERRUPT_CLEAR_0<*j*>_0.GPIO_INTERRUPT_CLEAR.

GPIO_<*k*>_ICG_EN_OVERRIDE_0,

where <*k*> = CTL0, CTL1, CTL2, CTL3, CTL4, CTL5, AON.

GPIO_CTL0_ICG_EN_OVERRIDE_0
Offset: 0x810

GPIO_CTL1_ICG_EN_OVERRIDE_0
Offset: 0x1810

GPIO_CTL2_ICG_EN_OVERRIDE_0
Offset: 0x2810

GPIO_CTL3_ICG_EN_OVERRIDE_0
Offset: 0x3810

GPIO_CTL4_ICG_EN_OVERRIDE_0
Offset: 0x4810

GPIO_CTL5_ICG_EN_OVERRIDE_0
Offset: 0x5810

GPIO_AON_ICG_EN_OVERRIDE_0
Offset: 0x0810

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GPIO_<*k*>_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**GPIO_ICG_EN_OVERRIDE**: This field enables ICG Override. **0** = ICG Override Disabled. **1** = ICG Override Enabled.

GPIO_<*i*>_INT<*jjj*>_ROUTE_MAPPING_0,

where <*i*> = A, B, C, D, E, F, G, H, I, J, K, L, M, N, P, Q, R, X, Y, Z, AA, AC, AD, AE, AF, AG, BB, CC, DD, EE, GG,

<*jj*> = 0, 1, 2, 3, 4, 5, 6, 7, (Security master)

<*jjj*> = 0, 1, 2, 3, 4, 5, 6, 7, (VM / Guest OS) and

<*k*> = CTL0, when <*i*> = A, B, AC, AD,
= CTL1, when <*i*> = X, Y, Z,
= CTL2, when <*i*> = M, N P, Q, R,
= CTL3, when <*i*> = K, , AE, AF, AG,
= CTL4, when <*i*> = G, H, I,
= CTL5, when <*i*> = C, D, E, F, J,
= AON, when <*i*>= AA, BB, CC, DD, EE, GG.

GPIO_A_INT0_ROUTE_MAPPING_0
Offset: 0x814

GPIO_A_INT1_ROUTE_MAPPING_0
Offset: 0x818

GPIO_A_INT2_ROUTE_MAPPING_0
Offset: 0x81c

GPIO_A_INT3_ROUTE_MAPPING_0
Offset: 0x820

GPIO_A_INT4_ROUTE_MAPPING_0
Offset: 0x824

GPIO_A_INT5_ROUTE_MAPPING_0
Offset: 0x828

GPIO_A_INT6_ROUTE_MAPPING_0
Offset: 0x82c

GPIO_A_INT7_ROUTE_MAPPING_0
Offset: 0x830

GPIO_AC_INT0_ROUTE_MAPPING_0
Offset: 0x834

GPIO_AC_INT1_ROUTE_MAPPING_0
Offset: 0x838

GPIO_AC_INT2_ROUTE_MAPPING_0
Offset: 0x83c

GPIO_AC_INT3_ROUTE_MAPPING_0
Offset: 0x840

GPIO_AC_INT4_ROUTE_MAPPING_0
Offset: 0x844

GPIO_AC_INT5_ROUTE_MAPPING_0
Offset: 0x848

GPIO_AC_INT6_ROUTE_MAPPING_0
Offset: 0x84c

GPIO_AC_INT7_ROUTE_MAPPING_0
Offset: 0x850

GPIO_X_INT0_ROUTE_MAPPING_0
Offset: 0x1814

GPIO_X_INT1_ROUTE_MAPPING_0
Offset: 0x1818

GPIO_X_INT2_ROUTE_MAPPING_0
Offset: 0x181c

GPIO_X_INT3_ROUTE_MAPPING_0
Offset: 0x1820

GPIO_X_INT4_ROUTE_MAPPING_0
Offset: 0x1824

GPIO_X_INT5_ROUTE_MAPPING_0
Offset: 0x1828

GPIO_X_INT6_ROUTE_MAPPING_0
Offset: 0x182c

GPIO_X_INT7_ROUTE_MAPPING_0
Offset: 0x1830

GPIO_Y_INT0_ROUTE_MAPPING_0
Offset: 0x1834

GPIO_Y_INT1_ROUTE_MAPPING_0
Offset: 0x1838

GPIO_Y_INT2_ROUTE_MAPPING_0
Offset: 0x183c

GPIO_Y_INT3_ROUTE_MAPPING_0
Offset: 0x1840

GPIO_Y_INT4_ROUTE_MAPPING_0
Offset: 0x1844

GPIO_Y_INT5_ROUTE_MAPPING_0
Offset: 0x1848

GPIO_Y_INT6_ROUTE_MAPPING_0
Offset: 0x184c

GPIO_Y_INT7_ROUTE_MAPPING_0
Offset: 0x1850

GPIO_Z_INT0_ROUTE_MAPPING_0
Offset: 0x1854

GPIO_Z_INT1_ROUTE_MAPPING_0
Offset: 0x1858

GPIO_Z_INT2_ROUTE_MAPPING_0
Offset: 0x185c

GPIO_Z_INT3_ROUTE_MAPPING_0
Offset: 0x1860

GPIO_Z_INT4_ROUTE_MAPPING_0
Offset: 0x1864

GPIO_Z_INT5_ROUTE_MAPPING_0
Offset: 0x1868

GPIO_Z_INT6_ROUTE_MAPPING_0
Offset: 0x186c

GPIO_Z_INT7_ROUTE_MAPPING_0
Offset: 0x1870

GPIO_M_INT0_ROUTE_MAPPING_0
Offset: 0x2814

GPIO_M_INT1_ROUTE_MAPPING_0
Offset: 0x2818

GPIO_M_INT2_ROUTE_MAPPING_0
Offset: 0x281c

GPIO_M_INT3_ROUTE_MAPPING_0
Offset: 0x2820

GPIO_M_INT4_ROUTE_MAPPING_0
Offset: 0x2824

GPIO_M_INT5_ROUTE_MAPPING_0
Offset: 0x2828

GPIO_M_INT6_ROUTE_MAPPING_0
Offset: 0x282c

GPIO_M_INT7_ROUTE_MAPPING_0
Offset: 0x2830

GPIO_N_INT0_ROUTE_MAPPING_0
Offset: 0x2834

GPIO_N_INT1_ROUTE_MAPPING_0
Offset: 0x2838

GPIO_N_INT2_ROUTE_MAPPING_0
Offset: 0x283c

GPIO_N_INT3_ROUTE_MAPPING_0
Offset: 0x2840

GPIO_N_INT4_ROUTE_MAPPING_0
Offset: 0x2844

GPIO_N_INT5_ROUTE_MAPPING_0
Offset: 0x2848

GPIO_N_INT6_ROUTE_MAPPING_0
Offset: 0x284c

GPIO_N_INT7_ROUTE_MAPPING_0
Offset: 0x2850

GPIO_P_INT0_ROUTE_MAPPING_0
Offset: 0x2854

GPIO_P_INT1_ROUTE_MAPPING_0
Offset: 0x2858

GPIO_P_INT2_ROUTE_MAPPING_0
Offset: 0x285c

GPIO_P_INT3_ROUTE_MAPPING_0
Offset: 0x2860

GPIO_P_INT4_ROUTE_MAPPING_0
Offset: 0x2864

GPIO_P_INT5_ROUTE_MAPPING_0
Offset: 0x2868

GPIO_P_INT6_ROUTE_MAPPING_0
Offset: 0x286c

GPIO_P_INT7_ROUTE_MAPPING_0
Offset: 0x2870

GPIO_Q_INT0_ROUTE_MAPPING_0
Offset: 0x2874

GPIO_Q_INT1_ROUTE_MAPPING_0
Offset: 0x2878

GPIO_Q_INT2_ROUTE_MAPPING_0
Offset: 0x287c

GPIO_Q_INT3_ROUTE_MAPPING_0
Offset: 0x2880

GPIO_Q_INT4_ROUTE_MAPPING_0
Offset: 0x2884

GPIO_Q_INT5_ROUTE_MAPPING_0
Offset: 0x2888

GPIO_Q_INT6_ROUTE_MAPPING_0
Offset: 0x288c

GPIO_Q_INT7_ROUTE_MAPPING_0
Offset: 0x2890

GPIO_K_INT0_ROUTE_MAPPING_0
Offset: 0x3814

GPIO_K_INT1_ROUTE_MAPPING_0
Offset: 0x3818

GPIO_K_INT2_ROUTE_MAPPING_0
Offset: 0x381c

GPIO_K_INT3_ROUTE_MAPPING_0
Offset: 0x3820

GPIO_K_INT4_ROUTE_MAPPING_0
Offset: 0x3824

GPIO_K_INT5_ROUTE_MAPPING_0
Offset: 0x3828

GPIO_K_INT6_ROUTE_MAPPING_0
Offset: 0x382c

GPIO_K_INT7_ROUTE_MAPPING_0
Offset: 0x3830

GPIO_AG_INT0_ROUTE_MAPPING_0
Offset: 0x3854

GPIO_AG_INT1_ROUTE_MAPPING_0
Offset: 0x3858

GPIO_AG_INT2_ROUTE_MAPPING_0
Offset: 0x385c

GPIO_AG_INT3_ROUTE_MAPPING_0
Offset: 0x3860

GPIO_AG_INT4_ROUTE_MAPPING_0
Offset: 0x3864

GPIO_AG_INT5_ROUTE_MAPPING_0
Offset: 0x3868

GPIO_AG_INT6_ROUTE_MAPPING_0
Offset: 0x386c

GPIO_AG_INT7_ROUTE_MAPPING_0
Offset: 0x3870

GPIO_G_INT0_ROUTE_MAPPING_0
Offset: 0x4814

GPIO_G_INT1_ROUTE_MAPPING_0
Offset: 0x4818

GPIO_G_INT2_ROUTE_MAPPING_0
Offset: 0x481c

GPIO_G_INT3_ROUTE_MAPPING_0
Offset: 0x4820

GPIO_G_INT4_ROUTE_MAPPING_0
Offset: 0x4824

GPIO_G_INT5_ROUTE_MAPPING_0
Offset: 0x4828

GPIO_G_INT6_ROUTE_MAPPING_0
Offset: 0x482c

GPIO_G_INT7_ROUTE_MAPPING_0
Offset: 0x4830

GPIO_H_INT0_ROUTE_MAPPING_0
Offset: 0x4834

GPIO_H_INT1_ROUTE_MAPPING_0
Offset: 0x4838

GPIO_H_INT2_ROUTE_MAPPING_0
Offset: 0x483c

GPIO_H_INT3_ROUTE_MAPPING_0
Offset: 0x4840

GPIO_H_INT4_ROUTE_MAPPING_0
Offset: 0x4844

GPIO_H_INT5_ROUTE_MAPPING_0
Offset: 0x4848

GPIO_H_INT6_ROUTE_MAPPING_0
Offset: 0x484c

GPIO_H_INT7_ROUTE_MAPPING_0
Offset: 0x4850

GPIO_C_INT0_ROUTE_MAPPING_0
Offset: 0x5834

GPIO_C_INT1_ROUTE_MAPPING_0
Offset: 0x5838

GPIO_C_INT2_ROUTE_MAPPING_0
Offset: 0x583c

GPIO_C_INT3_ROUTE_MAPPING_0
Offset: 0x5840

GPIO_C_INT4_ROUTE_MAPPING_0
Offset: 0x5844

GPIO_C_INT5_ROUTE_MAPPING_0
Offset: 0x5848

GPIO_C_INT6_ROUTE_MAPPING_0
Offset: 0x584c

GPIO_C_INT7_ROUTE_MAPPING_0
Offset: 0x5850

GPIO_E_INT0_ROUTE_MAPPING_0
Offset: 0x5874

GPIO_E_INT1_ROUTE_MAPPING_0
Offset: 0x5878

GPIO_E_INT2_ROUTE_MAPPING_0
Offset: 0x587c

GPIO_E_INT3_ROUTE_MAPPING_0
Offset: 0x5880

GPIO_E_INT4_ROUTE_MAPPING_0
Offset: 0x5884

GPIO_E_INT5_ROUTE_MAPPING_0
Offset: 0x5888

GPIO_E_INT6_ROUTE_MAPPING_0
Offset: 0x588c

GPIO_E_INT7_ROUTE_MAPPING_0
Offset: 0x5890

GPIO_EE_INT0_ROUTE_MAPPING_0
Offset: 0x0814

GPIO_EE_INT1_ROUTE_MAPPING_0
Offset: 0x0818

GPIO_EE_INT2_ROUTE_MAPPING_0
Offset: 0x81c

GPIO_EE_INT3_ROUTE_MAPPING_0
Offset: 0x0820

GPIO_EE_INT4_ROUTE_MAPPING_0
Offset: 0x0824

GPIO_EE_INT5_ROUTE_MAPPING_0
Offset: 0x0828

GPIO_EE_INT6_ROUTE_MAPPING_0
Offset: 0x082c

GPIO_EE_INT7_ROUTE_MAPPING_0
Offset: 0x0830

GPIO_CC_INT0_ROUTE_MAPPING_0
Offset: 0x0854

GPIO_CC_INT1_ROUTE_MAPPING_0
Offset: 0x0858

GPIO_CC_INT2_ROUTE_MAPPING_0
Offset: 0x085c

GPIO_CC_INT3_ROUTE_MAPPING_0
Offset: 0x0860

GPIO_CC_INT4_ROUTE_MAPPING_0
Offset: 0x0864

GPIO_CC_INT5_ROUTE_MAPPING_0
Offset: 0x0868

GPIO_CC_INT6_ROUTE_MAPPING_0
Offset: 0x086c

GPIO_CC_INT7_ROUTE_MAPPING_0
Offset: 0x0870

GPIO_AA_INT0_ROUTE_MAPPING_0
Offset: 0x0894

GPIO_AA_INT1_ROUTE_MAPPING_0
Offset: 0x0898

GPIO_AA_INT2_ROUTE_MAPPING_0
Offset: 0x089c

GPIO_AA_INT3_ROUTE_MAPPING_0
Offset: 0x08a0

GPIO_AA_INT4_ROUTE_MAPPING_0
Offset: 0x08a4

GPIO_AA_INT5_ROUTE_MAPPING_0
Offset: 0x08a8

GPIO_AA_INT6_ROUTE_MAPPING_0
Offset: 0x08ac

GPIO_AA_INT7_ROUTE_MAPPING_0
Offset: 0x08b0

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GPIO_<*k*>_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**VAL**: This field has as many bits as the number of GPIO Pins in the given GPIO Port, with each such bit indicating whether the corresponding GPIO Pin has caused the assertion of the given Interrupt (<*jj*>). **0** = No. **1** = Yes.

GPIO_I_INT0_ROUTE_MAPPING_0
Offset: 0x4854

GPIO_I_INT1_ROUTE_MAPPING_0
Offset: 0x4858

GPIO_I_INT2_ROUTE_MAPPING_0
Offset: 0x485c

GPIO_I_INT3_ROUTE_MAPPING_0
Offset: 0x4860

GPIO_I_INT4_ROUTE_MAPPING_0
Offset: 0x4864

GPIO_I_INT5_ROUTE_MAPPING_0
Offset: 0x4868

GPIO_I_INT6_ROUTE_MAPPING_0
Offset: 0x486c

GPIO_I_INT7_ROUTE_MAPPING_0
Offset: 0x4870

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GPIO_<*k*>_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000,0000)

Bit	Reset	Description
6:0	0x0	**VAL**: This field has as many bits as the number of GPIO Pins in the given GPIO Port, with each such bit indicating whether the corresponding GPIO Pin has caused the assertion of the given Interrupt (<*jj*>). **0** = No. **1** = Yes.

GPIO_R_INT0_ROUTE_MAPPING_0
Offset: 0x2894

GPIO_R_INT1_ROUTE_MAPPING_0
Offset: 0x2898

GPIO_R_INT2_ROUTE_MAPPING_0
Offset: 0x289c

GPIO_R_INT3_ROUTE_MAPPING_0
Offset: 0x28a0

GPIO_R_INT4_ROUTE_MAPPING_0
Offset: 0x28a4

GPIO_R_INT5_ROUTE_MAPPING_0
Offset: 0x28a8

GPIO_R_INT6_ROUTE_MAPPING_0
Offset: 0x28ac

GPIO_R_INT7_ROUTE_MAPPING_0
Offset: 0x28b0

GPIO_J_INT0_ROUTE_MAPPING_0
Offset: 0x5814

GPIO_J_INT1_ROUTE_MAPPING_0
Offset: 0x5818

GPIO_J_INT2_ROUTE_MAPPING_0
Offset: 0x581c

GPIO_J_INT3_ROUTE_MAPPING_0
Offset: 0x5820

GPIO_J_INT4_ROUTE_MAPPING_0
Offset: 0x5824

GPIO_J_INT5_ROUTE_MAPPING_0
Offset: 0x5828

GPIO_J_INT6_ROUTE_MAPPING_0
Offset: 0x582c

GPIO_J_INT7_ROUTE_MAPPING_0
Offset: 0x5830

GPIO_F_INT0_ROUTE_MAPPING_0
Offset: 0x5894

GPIO_F_INT1_ROUTE_MAPPING_0
Offset: 0x5898

GPIO_F_INT2_ROUTE_MAPPING_0
Offset: 0x589c

GPIO_F_INT3_ROUTE_MAPPING_0
Offset: 0x58a0

GPIO_F_INT4_ROUTE_MAPPING_0
Offset: 0x58a4

GPIO_F_INT5_ROUTE_MAPPING_0
Offset: 0x58a8

GPIO_F_INT6_ROUTE_MAPPING_0
Offset: 0x58ac

GPIO_F_INT7_ROUTE_MAPPING_0
Offset: 0x58b0

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GPIO_<*k*>_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**VAL**: This field has as many bits as the number of GPIO Pins in the given GPIO Port, with each such bit indicating whether the corresponding GPIO Pin has caused the assertion of the given Interrupt (<*jj*>). **0** = No. **1** = Yes.

GPIO_AD_INT0_ROUTE_MAPPING_0
Offset: 0x854

GPIO_AD_INT1_ROUTE_MAPPING_0
Offset: 0x858

GPIO_AD_INT2_ROUTE_MAPPING_0
Offset: 0x85c

GPIO_AD_INT3_ROUTE_MAPPING_0
Offset: 0x860

GPIO_AD_INT4_ROUTE_MAPPING_0
Offset: 0x864

GPIO_AD_INT5_ROUTE_MAPPING_0
Offset: 0x868

GPIO_AD_INT6_ROUTE_MAPPING_0
Offset: 0x86c

GPIO_AD_INT7_ROUTE_MAPPING_0
Offset: 0x870

GPIO_L_INT0_ROUTE_MAPPING_0
Offset: 0x3834

GPIO_L_INT1_ROUTE_MAPPING_0
Offset: 0x3838

GPIO_L_INT2_ROUTE_MAPPING_0
Offset: 0x383c

GPIO_L_INT3_ROUTE_MAPPING_0
Offset: 0x3840

GPIO_L_INT4_ROUTE_MAPPING_0
Offset: 0x3844

GPIO_L_INT5_ROUTE_MAPPING_0
Offset: 0x3848

GPIO_L_INT6_ROUTE_MAPPING_0
Offset: 0x384c

GPIO_L_INT7_ROUTE_MAPPING_0
Offset: 0x3850

GPIO_AF_INT0_ROUTE_MAPPING_0
Offset: 0x3894

GPIO_AF_INT1_ROUTE_MAPPING_0
Offset: 0x3898

GPIO_AF_INT2_ROUTE_MAPPING_0
Offset: 0x389c

GPIO_AF_INT3_ROUTE_MAPPING_0
Offset: 0x38a0

GPIO_AF_INT4_ROUTE_MAPPING_0
Offset: 0x38a4

GPIO_AF_INT5_ROUTE_MAPPING_0
Offset: 0x38a8

GPIO_AF_INT6_ROUTE_MAPPING_0
Offset: 0x38ac

GPIO_AF_INT7_ROUTE_MAPPING_0
Offset: 0x38b0

GPIO_D_INT0_ROUTE_MAPPING_0
Offset: 0x5854

GPIO_D_INT1_ROUTE_MAPPING_0
Offset: 0x5858

GPIO_D_INT2_ROUTE_MAPPING_0
Offset: 0x585c

GPIO_D_INT3_ROUTE_MAPPING_0
Offset: 0x5860

GPIO_D_INT4_ROUTE_MAPPING_0
Offset: 0x5864

GPIO_D_INT5_ROUTE_MAPPING_0
Offset: 0x5868

GPIO_D_INT6_ROUTE_MAPPING_0
Offset: 0x586c

GPIO_D_INT7_ROUTE_MAPPING_0
Offset: 0x5870

GPIO_BB_INT0_ROUTE_MAPPING_0
Offset: 0x08b4

GPIO_BB_INT1_ROUTE_MAPPING_0
Offset: 0x08b8

GPIO_BB_INT2_ROUTE_MAPPING_0
Offset: 0x08bc

GPIO_BB_INT3_ROUTE_MAPPING_0
Offset: 0x08c0

GPIO_BB_INT4_ROUTE_MAPPING_0
Offset: 0x08c4

GPIO_BB_INT5_ROUTE_MAPPING_0
Offset: 0x08c8

GPIO_BB_INT6_ROUTE_MAPPING_0
Offset: 0x08cc

GPIO_BB_INT7_ROUTE_MAPPING_0
Offset: 0x08d0

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GPIO_<*k*>_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	0x0	**VAL**: This field has as many bits as the number of GPIO Pins in the given GPIO Port, with each such bit indicating whether the corresponding GPIO Pin has caused the assertion of the given Interrupt (<*jj*>). **0** = No. **1** = Yes.

GPIO_DD_INT0_ROUTE_MAPPING_0
Offset: 0x0874

GPIO_DD_INT1_ROUTE_MAPPING_0
Offset: 0x0878

GPIO_DD_INT2_ROUTE_MAPPING_0
Offset: 0x087c

GPIO_DD_INT3_ROUTE_MAPPING_0
Offset: 0x0880

GPIO_DD_INT4_ROUTE_MAPPING_0
Offset: 0x0884

GPIO_DD_INT5_ROUTE_MAPPING_0
Offset: 0x0888

GPIO_DD_INT6_ROUTE_MAPPING_0
Offset: 0x088c

GPIO_DD_INT7_ROUTE_MAPPING_0
Offset: 0x0890

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GPIO_<*k*>_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	0x0	**VAL**: This field has as many bits as the number of GPIO Pins in the given GPIO Port, with each such bit indicating whether the corresponding GPIO Pin has caused the assertion of the given Interrupt (<*jj*>). **0** = No. **1** = Yes.

GPIO_AE_INT0_ROUTE_MAPPING_0
Offset: 0x3874

GPIO_AE_INT1_ROUTE_MAPPING_0
Offset: 0x3878

GPIO_AE_INT2_ROUTE_MAPPING_0
Offset: 0x387c

GPIO_AE_INT3_ROUTE_MAPPING_0
Offset: 0x3880

GPIO_AE_INT4_ROUTE_MAPPING_0
Offset: 0x3884

GPIO_AE_INT5_ROUTE_MAPPING_0
Offset: 0x3888

GPIO_AE_INT6_ROUTE_MAPPING_0
Offset: 0x388c

GPIO_AE_INT7_ROUTE_MAPPING_0
Offset: 0x3890

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GPIO_<*k*>_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**VAL**: This field has as many bits as the number of GPIO Pins in the given GPIO Port, with each such bit indicating whether the corresponding GPIO Pin has caused the assertion of the given Interrupt (<*jj*>). **0** = No. **1** = Yes.

GPIO_B_INT0_ROUTE_MAPPING_0
Offset: 0x874

GPIO_B_INT1_ROUTE_MAPPING_0
Offset: 0x878

GPIO_B_INT2_ROUTE_MAPPING_0
Offset: 0x87c

GPIO_B_INT3_ROUTE_MAPPING_0
Offset: 0x880

GPIO_B_INT4_ROUTE_MAPPING_0
Offset: 0x884

GPIO_B_INT5_ROUTE_MAPPING_0
Offset: 0x888

GPIO_B_INT6_ROUTE_MAPPING_0
Offset: 0x88c

GPIO_B_INT7_ROUTE_MAPPING_0
Offset: 0x890

GPIO_GG_INT0_ROUTE_MAPPING_0
Offset: 0x0834

GPIO_GG_INT1_ROUTE_MAPPING_0
Offset: 0x0838

GPIO_GG_INT2_ROUTE_MAPPING_0
Offset: 0x083c

GPIO_GG_INT3_ROUTE_MAPPING_0
Offset: 0x0840

GPIO_GG_INT4_ROUTE_MAPPING_0
Offset: 0x0844

GPIO_GG_INT5_ROUTE_MAPPING_0
Offset: 0x0848

GPIO_GG_INT6_ROUTE_MAPPING_0
Offset: 0x084c

GPIO_GG_INT7_ROUTE_MAPPING_0
Offset: 0x0850

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GPIO_<*k*>_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**VAL**: This field has as many bits as the number of GPIO Pins in the given GPIO Port, with each such bit indicating whether the corresponding GPIO Pin has caused the assertion of the given Interrupt (<*jj*>). **0** = No. **1** = Yes.

GPIO_<*k*>_SPARE_0,

where <*k*> = CTL0, CTL1, CTL2, CTL3, CTL4, CTL5, AON.

GPIO_CTL0_SPARE_0
Offset: 0x894

GPIO_CTL1_SPARE_0
Offset: 0x1874

GPIO_CTL2_SPARE_0
Offset: 0x28b4

GPIO_CTL3_SPARE_0
Offset: 0x38b4

GPIO_CTL4_SPARE_0
Offset: 0x4874

GPIO_CTL5_SPARE_0
Offset: 0x58b4

GPIO_AON_SPARE_0
Offset: 0x08d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**SPARE**

8.5.8 GPIO FSI Control Registers

NOTE:

All GPIO_S, GPIO_T, and GPIO_W Control Registers share the same base address under the name FSI_GPIO_CLT0.
All GPIO_U and GPIO_V Control Registers share the same base address under the name FSI_GPIO_CLT1.
For the base address of FSI GPIO Controller 0 Control Registers (FSI_GPIO_CLT0) and FSI GPIO Controller 1 Control Registers (FSI_GPIO_CLT1), please refer to the System Address Map in this TRM.

NOTE:

The binary “x” is turned to “0” for the Reset and PROD hex value. Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as “x” or “Undefined” actually means that the bit could be either “0” or “1” after Reset, whereas a bit PROD value as “x” or “Don’t-care” means that either “0” or “1” can be written to the bit by software Initialization.

GPIO_<*iii*>_ENABLE_CONFIG_0<*j*>_0,

where <*iii*> = S, T, W, U, V and

<*j*> ∈ {0, 1, 2, 3, 4, 5, 6, 7}.

GPIO_S_ENABLE_CONFIG_00_0

Offset: 0x1000

GPIO_S_ENABLE_CONFIG_01_0
Offset: 0x1020

GPIO_S_ENABLE_CONFIG_02_0
Offset: 0x1040

GPIO_S_ENABLE_CONFIG_03_0
Offset: 0x1060

GPIO_S_ENABLE_CONFIG_04_0
Offset: 0x1080

GPIO_S_ENABLE_CONFIG_05_0
Offset: 0x10a0

GPIO_S_ENABLE_CONFIG_06_0
Offset: 0x10c0

GPIO_S_ENABLE_CONFIG_07_0
Offset: 0x10e0

GPIO_T_ENABLE_CONFIG_00_0
Offset: 0x1200

GPIO_T_ENABLE_CONFIG_01_0
Offset: 0x1220

GPIO_W_ENABLE_CONFIG_00_0
Offset: 0x1400

GPIO_W_ENABLE_CONFIG_01_0
Offset: 0x1420

GPIO_U_ENABLE_CONFIG_00_0
Offset: 0x1000

GPIO_U_ENABLE_CONFIG_01_0
Offset: 0x1020

GPIO_U_ENABLE_CONFIG_02_0
Offset: 0x1040

GPIO_U_ENABLE_CONFIG_03_0
Offset: 0x1060

GPIO_U_ENABLE_CONFIG_04_0
Offset: 0x1080

GPIO_U_ENABLE_CONFIG_05_0
Offset: 0x10a0

GPIO_U_ENABLE_CONFIG_06_0
Offset: 0x10c0

GPIO_U_ENABLE_CONFIG_07_0
Offset: 0x10e0

GPIO_V_ENABLE_CONFIG_00_0
Offset: 0x1200

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GPIO_<*iii*>_SCR_0<*j*>_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7	0x0	**TIMESTAMPING_FUNCTION**: This field allows the GPIO input change event to create a timestamp event. When enabled based on the settings in TRIGGER_TYPE & TRIGGER LEVEL time-stamp trigger is sent out to TSC. It is effective only when (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE) && (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.IN_OUT == IN). **0** = DISABLE. **1** = ENABLE.
6	0x0	**INTERRUPT_FUNCTION**: This field provides the Interrupt selection for the GPIO Input changes based on trigger type (TRIGGER_TYPE) trigger polarity (TRIGGER LEVEL) It is effective only when (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE). **0** = DISABLE. **1** = ENABLE. When ((GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE) && (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.INTERRUPT_FUNCTION == 1)), the Interrupt Status is updated in the Interrupt Status Register (GPIO_<*i*>_INTERRUPT_STATUS_G<*jj*>_0) for all the VM's, based on the settings of GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.TRIGGER_TYPE & GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.TRIGGER LEVEL.
5	0x0	**DEBOUNCE_FUNCTION**: This field provides the Debounce filter selection on the GPIO Input. It is effective only when (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE). **0** = DISABLE (no Debounce filter). **1** = ENABLE. When (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.DEBOUNCE_FUNCTION == ENABLE), the debouncing interval value is defined by (GPIO_<*i*>_DEBOUNCE_THRESHOLD_0<*j*>_0.DEBOUNCE_THRESHOLD).

Bit	Reset	Description
4	0x0	**TRIGGER_LEVEL**: This field selects the polarity of the GPIO Input trigger for the given trigger type. It is effective only when (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE). When (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.TRIGGER_TYPE == NO TRIGGER), this field is irrelevant. When (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.TRIGGER_TYPE == LEVEL), **0** = Trigger on level Low. **1** = Trigger on Level High. When (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.TRIGGER_TYPE == SINGLE EDGE), **0** = Trigger on Falling Edge. **1** = Trigger on Raising Edge. When (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.TRIGGER_TYPE == BOTH EDGES), this field is irrelevant.
3:2	0x0	**TRIGGER_TYPE**: This field selects the GPIO Input trigger type (condition). It is effective only when (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE). **0** = NO_TRIGGER. **1** = LEVEL. **2** = SINGLE EDGE. **3** = DOUBLE_EDGE (i.e. both Low-High, High-Low edges are detected as event).
1	0x0	**IN_OUT**: This field configures the GPIO for Input or Output. It is effective only when (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE). **0** = IN. **1** = OUT.
0	0x0	**GPIO_ENABLE**: This field is the GPIO functionality selection control required for output/input selection. It acts as a global qualifier for all the functions of the specific GPIO. It is also used for SLCG (Second Level Clock Gating) per pin-based GPIO logic. **0** = DISABLE. **1** = ENABLE. When Disabled, the GPIO is floated, i.e. both input and output are disabled. At this point, If PinMux selects the GPIO, the I/O is in High-Z state.

GPIO_<*iii*>_DEBOUNCE_THRESHOLD_0<*j*>_0,

where <*iii*> = S, T, W, U, V and

<*j*> ∈ {0, 1, 2, 3, 4, 5, 6, 7}.

GPIO_S_DEBOUNCE_THRESHOLD_00_0
Offset: 0x1004

GPIO_S_DEBOUNCE_THRESHOLD_01_0
Offset: 0x1024

GPIO_S_DEBOUNCE_THRESHOLD_02_0
Offset: 0x1044

GPIO_S_DEBOUNCE_THRESHOLD_03_0
Offset: 0x1064

GPIO_S_DEBOUNCE_THRESHOLD_04_0
Offset: 0x1084

GPIO_S_DEBOUNCE_THRESHOLD_05_0
Offset: 0x10a4

GPIO_S_DEBOUNCE_THRESHOLD_06_0
Offset: 0x10c4

GPIO_S_DEBOUNCE_THRESHOLD_07_0
Offset: 0x10e4

GPIO_T_DEBOUNCE_THRESHOLD_00_0
Offset: 0x1204

GPIO_T_DEBOUNCE_THRESHOLD_01_0
Offset: 0x1224

GPIO_W_DEBOUNCE_THRESHOLD_00_0
Offset: 0x1404

GPIO_W_DEBOUNCE_THRESHOLD_01_0
Offset: 0x1424

GPIO_U_DEBOUNCE_THRESHOLD_00_0
Offset: 0x1004

GPIO_U_DEBOUNCE_THRESHOLD_01_0
Offset: 0x1024

GPIO_U_DEBOUNCE_THRESHOLD_02_0
Offset: 0x1044

GPIO_U_DEBOUNCE_THRESHOLD_03_0
Offset: 0x1064

GPIO_U_DEBOUNCE_THRESHOLD_04_0
Offset: 0x1084

GPIO_U_DEBOUNCE_THRESHOLD_05_0
Offset: 0x10a4

GPIO_U_DEBOUNCE_THRESHOLD_06_0
Offset: 0x10c4

GPIO_U_DEBOUNCE_THRESHOLD_07_0
Offset: 0x10e4

GPIO_V_DEBOUNCE_THRESHOLD_00_0
Offset: 0x1204
Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GPIO_<*iii*>_SCR_0<*j*>_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**DEBOUNCE_THRESHOLD**: This field holds the Debounce threshold in msec to specify the debounce interval. **0** = No Debounce. **1** ~ N-1: Debouncing interval of 1 msec to (N-1) msec. **NOTE:** The actual Debounce interval has an accuracy granularity of 1 msec. Hence, when programmed to "1", the actual Debouncing interval is between 1 msec and 2 msec.

GPIO_<*iii*>_INPUT_0<*j*>_0,

where <*iii*> = S, T, W, U, V and

<*j*> ∈ {0, 1, 2, 3, 4, 5, 6, 7}.

GPIO_S_INPUT_00_0
Offset: 0x1008

GPIO_S_INPUT_01_0
Offset: 0x1028

GPIO_S_INPUT_02_0
Offset: 0x1048

GPIO_S_INPUT_03_0
Offset: 0x1068

GPIO_S_INPUT_04_0
Offset: 0x1088

GPIO_S_INPUT_05_0
Offset: 0x10a8

GPIO_S_INPUT_06_0
Offset: 0x10c8

GPIO_S_INPUT_07_0
Offset: 0x10e8

GPIO_T_INPUT_00_0
Offset: 0x1208

GPIO_T_INPUT_01_0
Offset: 0x1228

GPIO_W_INPUT_00_0
Offset: 0x1408

GPIO_W_INPUT_01_0
Offset: 0x1428

GPIO_U_INPUT_00_0
Offset: 0x1008

GPIO_U_INPUT_01_0
Offset: 0x1028

GPIO_U_INPUT_02_0
Offset: 0x1048

GPIO_U_INPUT_03_0
Offset: 0x1068

GPIO_U_INPUT_04_0
Offset: 0x1088

GPIO_U_INPUT_05_0
Offset: 0x10a8

GPIO_U_INPUT_06_0
Offset: 0x10c8

GPIO_U_INPUT_07_0
Offset: 0x10e8

GPIO_V_INPUT_00_0
Offset: 0x1208

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: GPIO_<*iii*>_SCR_0<*j*>_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
0	X	**GPIO_IN**: This field holds the GPIO Input value sampled after the specified Debouncing interval. It is effective only when (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE). It reflects the external value regardless of (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.IN_OUT) setting, so that the externally driven value is present in this field to aid debugging.

GPIO_<*iii*>_OUTPUT_CONTROL_0<*j*>_0,

where <*iii*> = S, T, W, U, V and

<*j*> ∈ {0, 1, 2, 3, 4, 5, 6, 7}.

GPIO_S_OUTPUT_CONTROL_00_0
Offset: 0x100c

GPIO_S_OUTPUT_CONTROL_01_0
Offset: 0x102c

GPIO_S_OUTPUT_CONTROL_02_0
Offset: 0x104c

GPIO_S_OUTPUT_CONTROL_03_0
Offset: 0x106c

GPIO_S_OUTPUT_CONTROL_04_0
Offset: 0x108c

GPIO_S_OUTPUT_CONTROL_05_0
Offset: 0x10ac

GPIO_S_OUTPUT_CONTROL_06_0
Offset: 0x10cc

GPIO_S_OUTPUT_CONTROL_07_0
Offset: 0x10ec

GPIO_T_OUTPUT_CONTROL_00_0
Offset: 0x120c

GPIO_T_OUTPUT_CONTROL_01_0
Offset: 0x122c

GPIO_W_OUTPUT_CONTROL_00_0
Offset: 0x140c

GPIO_W_OUTPUT_CONTROL_01_0
Offset: 0x142c

GPIO_U_OUTPUT_CONTROL_00_0
Offset: 0x100c

GPIO_U_OUTPUT_CONTROL_01_0
Offset: 0x102c

GPIO_U_OUTPUT_CONTROL_02_0
Offset: 0x104c

GPIO_U_OUTPUT_CONTROL_03_0
Offset: 0x106c

GPIO_U_OUTPUT_CONTROL_04_0
Offset: 0x108c

GPIO_U_OUTPUT_CONTROL_05_0
Offset: 0x10ac

GPIO_U_OUTPUT_CONTROL_06_0
Offset: 0x10cc

GPIO_U_OUTPUT_CONTROL_07_0
Offset: 0x10ec

GPIO_V_OUTPUT_CONTROL_00_0
Offset: 0x120c

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GPIO_<*iii*>_SCR_0<*j*>_0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	0x1	**GPIO_OUT_CONTROL**: This field Indicates whether the GPIO as an output is Floated or Actively driven. It is effective only when ((GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE) && (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.IN_OUT == OUT)). It is floated regardless of the value specified here, When ((GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == DISABLE) || (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.IN_OUT == IN)). **0** = DRIVEN. **1** = FLOATED.

GPIO_<*iii*>_OUTPUT_VALUE_0<*j*>_0,

where <*iii*> = S, T, W, U, V and

<*j*> ∈ {0, 1, 2, 3, 4, 5, 6, 7}.

GPIO_S_OUTPUT_VALUE_00_0
Offset: 0x1010

GPIO_S_OUTPUT_VALUE_01_0
Offset: 0x1030

GPIO_S_OUTPUT_VALUE_02_0
Offset: 0x1050

GPIO_S_OUTPUT_VALUE_03_0
Offset: 0x1070

GPIO_S_OUTPUT_VALUE_04_0
Offset: 0x1090

GPIO_S_OUTPUT_VALUE_05_0
Offset: 0x10b0

GPIO_S_OUTPUT_VALUE_06_0
Offset: 0x10d0

GPIO_S_OUTPUT_VALUE_07_0
Offset: 0x10f0

GPIO_T_OUTPUT_VALUE_00_0
Offset: 0x1210

GPIO_T_OUTPUT_VALUE_01_0
Offset: 0x1230

GPIO_W_OUTPUT_VALUE_00_0
Offset: 0x1410

GPIO_W_OUTPUT_VALUE_01_0
Offset: 0x1430

GPIO_U_OUTPUT_VALUE_00_0
Offset: 0x1010

GPIO_U_OUTPUT_VALUE_01_0
Offset: 0x1030

GPIO_U_OUTPUT_VALUE_03_0
Offset: 0x1070

GPIO_U_OUTPUT_VALUE_04_0
Offset: 0x1090

GPIO_U_OUTPUT_VALUE_05_0
Offset: 0x10b0

GPIO_U_OUTPUT_VALUE_06_0
Offset: 0x10d0

GPIO_U_OUTPUT_VALUE_07_0
Offset: 0x10f0

GPIO_V_OUTPUT_VALUE_00_0
Offset: 0x1210

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GPIO_<*iii*>_SCR_0<*j*>_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**GPIO_OUT_VAL**: This field holds the GPIO output value to be driven out when (GPIO_<*i*>_OUTPUT_CONTROL_<*j*>_0.GPIO_OUT_CONTROL == DRIVEN). It is effective only when ((GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE) && (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.IN_OUT == OUT)).

GPIO_<*iii*>_INTERRUPT_CLEAR_0<*j*>_0,

where <*iii*> = S, T, W, U, V and

<*j*> ∈ {0, 1, 2, 3, 4, 5, 6, 7}.

GPIO_S_INTERRUPT_CLEAR_00_0
Offset: 0x1014

GPIO_S_INTERRUPT_CLEAR_01_0
Offset: 0x1034

GPIO_S_INTERRUPT_CLEAR_02_0
Offset: 0x1054

GPIO_S_INTERRUPT_CLEAR_03_0
Offset: 0x1074

GPIO_S_INTERRUPT_CLEAR_04_0
Offset: 0x1094

GPIO_S_INTERRUPT_CLEAR_05_0
Offset: 0x10b4

GPIO_S_INTERRUPT_CLEAR_06_0
Offset: 0x10d4

GPIO_S_INTERRUPT_CLEAR_07_0
Offset: 0x10f4

GPIO_T_INTERRUPT_CLEAR_00_0
Offset: 0x1214

GPIO_T_INTERRUPT_CLEAR_01_0
Offset: 0x1234

GPIO_W_INTERRUPT_CLEAR_00_0
Offset: 0x1414

GPIO_W_INTERRUPT_CLEAR_01_0
Offset: 0x1434

GPIO_U_INTERRUPT_CLEAR_00_0
Offset: 0x1014

GPIO_U_INTERRUPT_CLEAR_01_0
Offset: 0x1034

GPIO_U_INTERRUPT_CLEAR_02_0
Offset: 0x1054

GPIO_U_INTERRUPT_CLEAR_03_0
Offset: 0x1074

GPIO_U_INTERRUPT_CLEAR_04_0
Offset: 0x1094

GPIO_U_INTERRUPT_CLEAR_05_0
Offset: 0x10b4

GPIO_U_INTERRUPT_CLEAR_06_0
Offset: 0x10d4

GPIO_U_INTERRUPT_CLEAR_07_0
Offset: 0x10f4

GPIO_V_INTERRUPT_CLEAR_00_0
Offset: 0x1214

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GPIO_<*iii*>_SCR_0<*j*>_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**GPIO_INTERRUPT_CLEAR**: This field holds the Interrupt clear control bit for clearing the Interrupt at GPIO Pin <*j*> in GPIO Port <*i*>/<*iii*>. It is effective only when ((GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0. (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.INTERRUPT_FUNCTION == ENABLE)). **0** = DON'T CLEAR. **1** = CLEAR.

GPIO_<*iii*>_INTERRUPT_STATUS_G<*jj*>_0,

where <*iii*> = S, T, W, U, V and

<*jj*> = 0, 1, 2, 3, 4, 5, 6, 7.

GPIO_S_INTERRUPT_STATUS_G0_0
Offset: 0x1100

GPIO_S_INTERRUPT_STATUS_G1_0
Offset: 0x1104

GPIO_S_INTERRUPT_STATUS_G2_0
Offset: 0x1108

GPIO_S_INTERRUPT_STATUS_G3_0
Offset: 0x110c

GPIO_S_INTERRUPT_STATUS_G4_0
Offset: 0x1110

GPIO_S_INTERRUPT_STATUS_G5_0
Offset: 0x1114

GPIO_S_INTERRUPT_STATUS_G6_0
Offset: 0x1118

GPIO_S_INTERRUPT_STATUS_G7_0
Offset: 0x111c

GPIO_U_INTERRUPT_STATUS_G0_0
Offset: 0x1100

GPIO_U_INTERRUPT_STATUS_G1_0
Offset: 0x1104

GPIO_U_INTERRUPT_STATUS_G2_0
Offset: 0x1108

GPIO_U_INTERRUPT_STATUS_G3_0
Offset: 0x110c

GPIO_U_INTERRUPT_STATUS_G4_0
Offset: 0x1110

GPIO_U_INTERRUPT_STATUS_G5_0
Offset: 0x1114

GPIO_U_INTERRUPT_STATUS_G6_0
Offset: 0x1118

GPIO_U_INTERRUPT_STATUS_G7_0
Offset: 0x111c

Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**GPIO_INTERRUPT_STATUS**: This field has as many bits as the number of GPIO Pins in the given GPIO Port, with each such bit indicating whether the corresponding GPIO Input change has resulted in Interrupt. Bit 0 is for Pin 0, Bit 1 is for Pin 1, ... Bit <*j*> is for Pin <*j*>. **0** = Interrupt not set. **1** = Interrupt set. It is effective only when ((GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE) && (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.INTERRUPT_FUNCTION == ENABLE)). The bit in this field is cleared based on GPIO_<*i*>/<*iii*>_INTERRUPT_CLEAR_0<*j*>_0.GPIO_INTERRUPT_CLEAR. In a Virtualized environment, the status reflects only the GPIOs belonging to VM. For the GPIO bit positions not mapped to VM, the corresponding bit in this field is written with "0" by hardware for software to read this register correctly.

GPIO_T_INTERRUPT_STATUS_G0_0
Offset: 0x1300

GPIO_T_INTERRUPT_STATUS_G1_0
Offset: 0x1304

GPIO_T_INTERRUPT_STATUS_G2_0
Offset: 0x1308

GPIO_T_INTERRUPT_STATUS_G3_0
Offset: 0x130c

GPIO_T_INTERRUPT_STATUS_G4_0
Offset: 0x1310

GPIO_T_INTERRUPT_STATUS_G5_0
Offset: 0x1314

GPIO_T_INTERRUPT_STATUS_G6_0
Offset: 0x1318

GPIO_T_INTERRUPT_STATUS_G7_0
Offset: 0x131c

GPIO_W_INTERRUPT_STATUS_G0_0
Offset: 0x1500

GPIO_W_INTERRUPT_STATUS_G1_0
Offset: 0x1504

GPIO_W_INTERRUPT_STATUS_G2_0
Offset: 0x1508

GPIO_W_INTERRUPT_STATUS_G3_0
Offset: 0x150c

GPIO_W_INTERRUPT_STATUS_G4_0
Offset: 0x1510

GPIO_W_INTERRUPT_STATUS_G5_0
Offset: 0x1514

GPIO_W_INTERRUPT_STATUS_G6_0
Offset: 0x1518

GPIO_W_INTERRUPT_STATUS_G7_0
Offset: 0x151c

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**GPIO_INTERRUPT_STATUS**: This field has as many bits as the number of GPIO Pins in the given GPIO Port, with each such bit indicating whether the corresponding GPIO Input change has resulted in Interrupt. Bit 0 is for Pin 0, Bit 1 is for Pin 1, ... Bit <*j*> is for Pin <*j*>. **0** = Interrupt not set. **1** = Interrupt set. It is effective only when ((GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE) && (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.INTERRUPT_FUNCTION == ENABLE)). The bit in this field is cleared based on GPIO_<*i*>/<*iii*>_INTERRUPT_CLEAR_0<*j*>_0.GPIO_INTERRUPT_CLEAR. In a Virtualized environment, the status reflects only the GPIOs belonging to VM. For the GPIO bit positions not mapped to VM, the corresponding bit in this field is written with "0" by hardware for software to read this register correctly.

GPIO_V_INTERRUPT_STATUS_G0_0
Offset: 0x1300

GPIO_V_INTERRUPT_STATUS_G1_0
Offset: 0x1304

GPIO_V_INTERRUPT_STATUS_G2_0
Offset: 0x1308

GPIO_V_INTERRUPT_STATUS_G3_0
Offset: 0x130c

GPIO_V_INTERRUPT_STATUS_G4_0
Offset: 0x1310

GPIO_V_INTERRUPT_STATUS_G5_0
Offset: 0x1314

GPIO_V_INTERRUPT_STATUS_G6_0
Offset: 0x1318

GPIO_V_INTERRUPT_STATUS_G7_0
Offset: 0x131c

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**GPIO_INTERRUPT_STATUS**: This field has as many bits as the number of GPIO Pins in the given GPIO Port, with each such bit indicating whether the corresponding GPIO Input change has resulted in Interrupt. Bit 0 is for Pin 0, Bit 1 is for Pin 1, ... Bit <*j*> is for Pin <*j*>. **0** = Interrupt not set. **1** = Interrupt set. It is effective only when ((GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.GPIO_ENABLE == ENABLE) && (GPIO_<*i*>_ENABLE_ CONFIG_0<*j*>_0.INTERRUPT_FUNCTION == ENABLE)). The bit in this field is cleared based on GPIO_<*i*>/<*iii*>_INTERRUPT_CLEAR_0<*j*>_0.GPIO_INTERRUPT_CLEAR. In a Virtualized environment, the status reflects only the GPIOs belonging to VM. For the GPIO bit positions not mapped to VM, the corresponding bit in this field is written with "0" by hardware for software to read this register correctly.

8.5.9 GPIO FSI Common Control Registers

NOTE:

All GPIO_S, GPIO_T, and GPIO_W Common Control Registers share the same base address under the name FSI_GPIO_CLT0.
All GPIO_U and GPIO_V Common Control Registers share the same base address under the name FSI_GPIO_CLT1.
For the base address of FSI GPIO Controller 0 Common Control Registers (FSI_GPIO_CLT0) and FSI GPIO Controller 1 Common Control Registers (FSI_GPIO_CLT1), please refer to the System Address Map in this TRM.

NOTE:

The binary "x" is turned to "0" for the Reset and PROD hex value. Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as "x" or "Undefined" actually means that the bit could be either "0" or "1" after Reset, whereas a bit PROD value as "x" or "Don't-care" means that either "0" or "1" can be written to the bit by software Initialization.

GPIO_FSI<*m*>_ICG_EN_OVERRIDE_0,

where <*m*> = 0, 1.

GPIO_FSI0_ICG_EN_OVERRIDE_0
Offset: 0x810

GPIO_FSI1_ICG_EN_OVERRIDE_0
Offset: 0x810

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GPIO_FSI<*m*>_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**GPIO_ICG_EN_OVERRIDE**: This field enables ICG Override. **0** = ICG Override Disabled. **1** = ICG Override Enabled.

GPIO_<*iii*>_INT<*jjj*>_ROUTE_MAPPING_0,

where <*iii*> = S, T, W, U, V and

<*jjj*> = 0, 1, 2, 3, 4, 5, 6, 7.

GPIO_S_INT0_ROUTE_MAPPING_0
Offset: 0x814

GPIO_S_INT1_ROUTE_MAPPING_0
Offset: 0x818

GPIO_S_INT2_ROUTE_MAPPING_0
Offset: 0x81c

GPIO_S_INT3_ROUTE_MAPPING_0
Offset: 0x820

GPIO_S_INT4_ROUTE_MAPPING_0
Offset: 0x824

GPIO_S_INT5_ROUTE_MAPPING_0
Offset: 0x828

GPIO_S_INT6_ROUTE_MAPPING_0
Offset: 0x82c

GPIO_S_INT7_ROUTE_MAPPING_0
Offset: 0x830

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GPIO_FSI0_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
7:0	0x0	**VAL**: This field has as many bits as the number of GPIO Pins in the given GPIO Port, with each such bit indicating whether the corresponding GPIO Pin has caused the assertion of the given Interrupt (<*jj*>). **0** = No. **1** = Yes.

GPIO_T_INT0_ROUTE_MAPPING_0
Offset: 0x834

GPIO_T_INT1_ROUTE_MAPPING_0
Offset: 0x838

GPIO_T_INT2_ROUTE_MAPPING_0
Offset: 0x83c

GPIO_T_INT3_ROUTE_MAPPING_0
Offset: 0x840

GPIO_T_INT4_ROUTE_MAPPING_0
Offset: 0x844

GPIO_T_INT5_ROUTE_MAPPING_0
Offset: 0x848

GPIO_T_INT6_ROUTE_MAPPING_0
Offset: 0x84c

GPIO_T_INT7_ROUTE_MAPPING_0
Offset: 0x850

GPIO_W_INT0_ROUTE_MAPPING_0
Offset: 0x854

GPIO_W_INT1_ROUTE_MAPPING_0
Offset: 0x858

GPIO_W_INT2_ROUTE_MAPPING_0
Offset: 0x85c

GPIO_W_INT3_ROUTE_MAPPING_0
Offset: 0x860

GPIO_W_INT4_ROUTE_MAPPING_0
Offset: 0x864

GPIO_W_INT5_ROUTE_MAPPING_0
Offset: 0x868

GPIO_W_INT6_ROUTE_MAPPING_0
Offset: 0x86c

GPIO_W_INT7_ROUTE_MAPPING_0
Offset: 0x870

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GPIO_FSI0_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1:0	0x0	**VAL**: This field has as many bits as the number of GPIO Pins in the given GPIO Port, with each such bit indicating whether the corresponding GPIO Pin has caused the assertion of the given Interrupt (<*jj*>). **0** = No. **1** = Yes.

GPIO_U_INT0_ROUTE_MAPPING_0
Offset: 0x814

GPIO_U_INT1_ROUTE_MAPPING_0
Offset: 0x818

GPIO_U_INT2_ROUTE_MAPPING_0
Offset: 0x81c

GPIO_U_INT3_ROUTE_MAPPING_0
Offset: 0x820

GPIO_U_INT4_ROUTE_MAPPING_0
Offset: 0x824

GPIO_U_INT5_ROUTE_MAPPING_0
Offset: 0x828

GPIO_U_INT6_ROUTE_MAPPING_0
Offset: 0x82c

GPIO_U_INT7_ROUTE_MAPPING_0
Offset: 0x830

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GPIO_FSI1_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
7:0	0x0	**VAL**: This field has as many bits as the number of GPIO Pins in the given GPIO Port, with each such bit indicating whether the corresponding GPIO Pin has caused the assertion of the given Interrupt (<*jj*>). **0** = No. **1** = Yes.

GPIO_V_INT0_ROUTE_MAPPING_0
Offset: 0x834

GPIO_V_INT1_ROUTE_MAPPING_0
Offset: 0x838

GPIO_V_INT2_ROUTE_MAPPING_0
Offset: 0x83c

GPIO_V_INT3_ROUTE_MAPPING_0
Offset: 0x840

GPIO_V_INT4_ROUTE_MAPPING_0
Offset: 0x844

GPIO_V_INT5_ROUTE_MAPPING_0
Offset: 0x848

GPIO_V_INT6_ROUTE_MAPPING_0
Offset: 0x84c

GPIO_V_INT7_ROUTE_MAPPING_0
Offset: 0x850

Read/Write: R/W
Parity Protection: Y
Shadow: N
SCR Protection: GPIO_FSI1_SCR_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**VAL**: This field has as many bits as the number of GPIO Pins in the given GPIO Port, with each such bit indicating whether the corresponding GPIO Pin has caused the assertion of the given Interrupt (<*jj*>). **0** = No. **1** = Yes.

GPIO_FSI<*m*>_SPARE_0,

where <*m*> = 0, 1.

GPIO_FSI0_SPARE_0
Offset: 0x874

GPIO_FSI1_SPARE_0
Offset: 0x854

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**SPARE**

8.6 Hardware Synchronization Primitives (HSP)

8.6.1 Overview

This chapter describes various modules used to allow multiple processors to share resources and communicate together. These synchronization elements are intended for software level interprocessor synchronization and messaging, especially between different Arm processors. This chapter does not discuss the synchronization elements used between a processor and a peripheral, or the mechanisms used between software threads in the same processor cluster.

8.6.1.1 Rationale

The Arm® instructions supporting exclusive access (LDREX/STREX, SWAP, etc.) can only be used inside a coherency domain and so cannot be used between different processor clusters in the SoC. This hardware therefore provides a set of hardware synchronization primitives for interprocessor synchronization.

There are also times when the cores inside the CCPLEX clusters do not operate in an SMP fashion. This is required particularly during some power management sequences with caches and/or MMU disabled. Hardware synchronization may again be required during these periods if exclusive access to a shared resource is required.

8.6.1.2 Use Cases

Interprocessor Communication (IPC) protocols use hardware synchronization mechanisms, especially the doorbells, when operating between two processors not in an SMP relationship.

8.6.1.3 Shared Mailboxes

The shared mailboxes support a single-writer and single-reader model.

8.6.1.4 Shared Semaphores

The shared semaphores are formed by three registers, each 32-bits wide:

- A register showing the current value of the semaphore
- Two write-only registers to set and clear individual semaphore bits

8.6.1.5 Arbitrated Semaphores

The arbitrated semaphores are true mutex binary semaphores that also include state information to avoid software use of a busy spin loop if so desired.

8.6.1.6 Doorbells

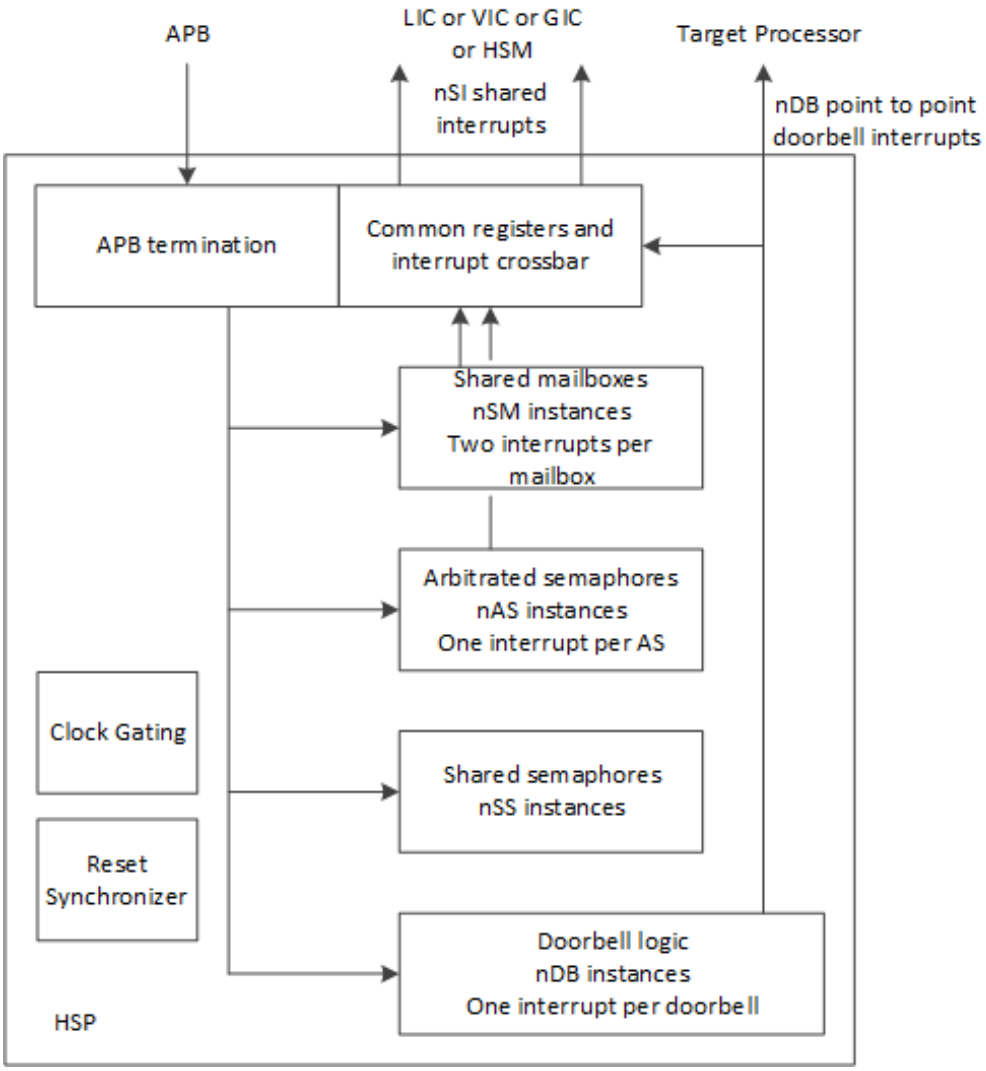
The doorbells implement a simple mechanism to support IPC protocols. Each doorbell is one-to-one associated with a given destination processor, and the assertion of the doorbell signal indicates to the destination processor that at least one other processor requires attention, typically because the source processor posted a message in memory.

The doorbell mechanism uses a source-tracking approach to aggregate multiple requests for attention into a single register, simplifying the software communication model: the target processor identifies the set of pending requests through a single register read.

8.6.2 Functional Description

The SoC implements an IP block that encapsulates the multiple hardware synchronization primitives, and is accessed via an APB bus.

Figure 8.55 Decomposition of the Hardware Synchronization Primitive



The HSP module is instantiated multiple times in Orin, together with the number of instances of the different primitives inside each instance.

In the following table:

- nSM is the number of Shared Mailboxes
- nAS is the number of Arbitrated Semaphores
- nSS is the number of Shared Semaphores
- nDB is the number of Doorbells
- nSI is the number of Shared Interrupts (see below)
- Red shows which HSPs are redundant
- Type shows the two possible HSP configurations

Table 8.60 HSP Module Instantiation

Instance	nSM	nAS	nSS	nDB	nSI	Red	Type	Description
TOP0	8	2	2	10	8	Yes	0	TOP0 instance additionally has Arbitrated Semaphores and Doorbells with each doorbell having an identified destination.
TOP1	8	0	4	0	8	Yes	1	Consists a total of eight bidirectional mailboxes at TOP, same as R5 HSP to limit verification effort.
TOP2	8	0	4	0	8	Yes	1	Same as TOP1 Flavor and is mainly used for FSI<->CCPLEX communication.
BPMP	8	0	4	0	8	Yes	1	All Cortex-R5 clusters, the A9 cluster, FSI Cluster, TOP1, and TOP2 instances use a similarly configured HSP.
AON	8	0	4	0	8			
SCE	8	0	4	0	8	Yes		
SCE_HSP1	8	0	4	0	8	Yes		
SCE_HSP2	8	0	4	0	8	Yes		
APE	8	0	4	0	8			
RCE	8	0	4	0	8	Yes		
RCE HSP1	8	0	4	0	8	Yes		
RCE HSP2	8	0	4	0	8	Yes		
PVA0	8	0	4	0	8			
DCE	8	0	4	0	8	Yes		
DCE HSP1	8	0	4	0	8	Yes		
DCE HSP2	8	0	4	0	8	Yes		
FSI	8	0	4	0	8	Yes		

The Shared Mailboxes, Arbitrated Semaphores, and Shared Semaphores are always replicated per security level, and each HSP supports two different security levels (TrustZone® or non-TrustZone), so that nSM, nAS, and nSS are always a multiple of two to allow differentiation between two security levels. The doorbell logic is only duplicated for the CPU where one doorbell can be used to ping the TrustZone secure software without interference from TrustZone non-secure software or exposing any information to the TrustZone non-secure software.

Note: RCE_HSP1, RCE_HSP2, DCE_HSP1, and DCE_HSP2 instances exist to maintain common design for SCE, RCE and DCE.

Doorbells are system resources, instantiated once into a top level HSP. Doorbell interrupts are sent to specific targets:

- Two doorbells connected to the LIC, and from there to the CCPLEX GIC, are normally mapped to TrustZone secure and non-secure. (The CCPLEX CPU doorbell signals could be made directly visible to the CCPMU, this is not done in Orin.)
- One doorbell directly connected to the CCPMU inside CCPLEX (the MCE used to communicate with BPMP).
- Five doorbells for AON, BPMP, SCE, DCE, and RCE R5 clusters. Each doorbell connected to VIC inside a Cortex-R5 cluster.
- One doorbell connected to the PSC interrupt controller.

8.6.2.1 Sub-Unit Descriptions

The HSP module has these main blocks:

- APB termination
- Common registers and interrupt crossbar
- Shared mailboxes
- Shared semaphores
- Arbitrated semaphores
- Doorbells

8.6.2.1.1 APB Termination

The APB termination logic is a slave APB that transforms incoming APB transactions into local register read and write operations.

8.6.2.1.2 Common Registers and Interrupt Crossbar

The common registers and interrupt crossbar block contains the interrupt routing logic that takes the set of local interrupts and generates a set of shared interrupts, connected to an interrupt controller associated with the instance:

- LIC for TOP0 and TOP1 HSP.
- LIC, FSI GIC, FSI AVIC, and FSI HSM for TOP2_HSP.
- One shared interrupt to the local AVIC and four shared interrupts to LIC for AON, SCE, RCE, DCE, and PVA HSP instances.
- Local AVIC, FSI GIC, FSI HSM for SCE_HSP1 and SCE_HSP2 instances.
- Local AVIC for RCE_HSP1, RCE_HSP2, DCE_HSP1, and DCE_HSP2 instances.
- Local GIC and AVIC for FSI HSP instance.
- The LIC and local VIC for HSP instantiated in BPMP.
- The A9 GIC for HSP instantiated inside the APE.

Note: Number of shared interrupts (nSI) is eight for all HSP instances but only five of them are used in AON_HSP, BPMP_HSP, SCE_HSP, RCE_HSP, DCE_HSP, PVA_HSP, and APE_HSP instances.

8.6.2.1.3 Shared Mailboxes

HSP supports two types of shared mailboxes, 32-bit and 128-bit. Each of the eight shared mailboxes in HSP can be used as either 32-bit shared mailbox or 128-bit mailbox. Usage of 32-bit Mailbox and 128-bit Mailbox is mutually exclusive as they share same Full and Empty interrupts.

32-bit Shared Mailbox

The 32-bit shared mailboxes are word width registers with a very simple interrupt mechanism controlled by the MSB. Their simplest operating model is single writer/single reader.

Each 32-bit shared mailbox has the following resources:

- A 32-bit register.
- Full and empty interrupts controlled by the MSB value of the register. The MSB is interpreted as a Valid bit, so the full interrupt is the value of the MSB and the empty interrupt is the inverted value of the MSB.
- Enable bits for full and empty interrupts.

128-bit Shared Mailbox

To add support for Mailbox data width>31-bit, 128-bit Mailbox support is added in Orin**:**

- Current Software is not impacted and can choose between 32-bit mailbox vs 128-bit mailbox as per the use-cases.
- Five new registers are added for 128-bit mailbox support: one register for TAG and four registers of each 32-bit for mailbox data.
- Mailbox full interrupt is asserted when the TAG field of the TAG register is one and mailbox empty bit is asserted when the TAG is set to 0 in TAG register.
- Mailbox data can be written into the four new mailbox data registers as per the required data size.

The number of shared mailboxes is extended to support more than one pair of communicating processors and virtualization requirements. There are 24 unidirectional mailboxes (eight for TOP0, eight for TOP1, eight for TOP2) at TOP_HSP and eight unidirectional mailboxes local to each Cortex-R5, FSI, and A9 cluster: APE, BPMP, AON, SCE, RCE, DCE, PVA0, and FSI. Additionally, SCE cluster has SCE_HSP1 and SCE_HSP2 instances each with eight unidirectional mailboxes, RCE cluster has RCE_HSP1 and RCE_HSP2 instances each with eight unidirectional mailboxes and DCE cluster has DCE_HSP1 and DCE_HSP2 instances each with eight unidirectional mailboxes.

The expected usage model of shared mailboxes is:

- Dedicated to a specific consumer, as it is generally easier to share the input port

- Dedicated to a bidirectional IPC method, in which case shared mailboxes are allocated as a pair, one per direction
- Preferably, the IPC is between one producer and one consumer to avoid resource conflicts This is not always possible in multithreaded or multiprocessor use cases, in which either:
  - A single threaded software agent is used as a resource manager per direction, arbitrating, and dispatching message. This assumes that all software producers are able to synchronize using software mechanisms, i.e., not shared mailboxes, and the same for all software consumers. This is a viable model for the BPMP ↔ CCPLEX IPC.
  - The input and output semaphores are used for resource sharing
- No easy way to operate a mailbox with agents at different levels of security on the same port. If the IPC requires a mix of security levels and QoS per level, the IPC must use as many mailboxes as required to achieve the required QoS.
- Targeting (very) short messages, not bulk transfer. The assumption is that the message can identify a descriptor for a larger structure when needed, possibly via an address offset against some known base or an index in a list or whatever is convenient for software. Shared mailboxes are not optimized for bandwidth.

8.6.2.1.4 Shared Semaphores

The shared semaphores are registers with associated SET/CLR addresses to allow easy manipulation of individual bits inside them, i.e., without the need for a RMW operation. There is no hardware arbitration, so software must statically allocate the ownership of individual semaphores for correct operation.

The shared semaphores are formed by three registers, each 32-bits wide:

- Read-write register showing the current value of the semaphore
- Two write-only registers to set and clear individual semaphore bits

These semaphores can be used for synchronizing processors acting in a producer/consumer relationship, i.e., a pair of them can be used to implement either a two- or four-way handshake.

Correct operation of the shared semaphores require to statically allocate an owner for each bit, because it is impossible to implement an atomic test and set (or similar operation) given the register interface. It also means that calling them "semaphores" is a bit misleading; they are atomic set and atomic clear registers.

8.6.2.1.5 Arbitrated Semaphores

The arbitrated semaphores are true binary semaphores that also include state information to avoid a busy spin loop if so wanted. The arbitrated semaphores support two sides, denoted 0 and 1, and arbitrates between requests coming from each side.

In essence, the arbitrated semaphores are built in the following way, all registers being 32-bits wide:

- A set of read only flags representing ownership (SMP_GNT_ST), showing ownership of the semaphore, one register per side (0 and 1).
- A set of write only trigger request flags (SMP_GET), a processor can only acquire a resource (represented by a bit in the status word) through setting a corresponding request flag. The request is blocked until the resource is not owned by a different processor. Currently, request flags are only cleared by acquiring the resource. Again, there is one register per side (0 and 1).
- A set of clear flags (SMP_PUT), a processor relinquishes ownership by writing a one bit corresponding to the shared resource (read-write).
- A set of request status flags (SMP_REQ_ST) showing the current pending requests (read only)

One companion set of registers is defined, with one register per side (0 and 1).

- Enable register (INT_EN), enabling the generation of an interrupt if the corresponding status bit toggles.

Storing the request information instead of using write triggers allows the following implementation to avoid the typical busy spin lock loop and the corresponding waste of bandwidth and power.

```
EnableOwnershipInterrupt (i);
RequestResource (i);
WaitForInterrupt();
```

8.6.2.1.6 Doorbell Logic

A doorbell allows a set of source agents in the chip to request the attention of a specified target agent.

The doorbell logic presents two logically separate interfaces:

1. The trigger interface can be used by any master in the system; the doorbell logic uses source ID tracking to recognize the exact source.
2. The control interface is normally configured for use by the target agent only.

Note that the doorbell SLICE1 registers support masters with masterId>15.

The trigger interface also distinguishes between TrustZone secure and not secure security levels. The doorbell includes one bit per possible source, per supported security level. Writing to the trigger interface sets a corresponding bit as shown in the pseudo code below (for one doorbell):

```
SourceID = f(pUSER) // extract source ID for APB USER field
If SourceID <= 15
      If pns == TZ_SECURE:     Slice0[Map_0][SourceID] = 1b
      If pns == TZ_NOT_SECURE: Slice0[Map_1][SourceID] = 1b
  Else if ((SourceId >=16) && (SourceId < 32))
      If pns == TZ_SECURE:     Slice1[Map_0][SourceID] = 1b
      If pns == TZ_NOT_SECURE: Slice1[Map_1][SourceID] = 1b
```

The control logic uses security aware registers to control whose agent has access to it, with the following set of registers:

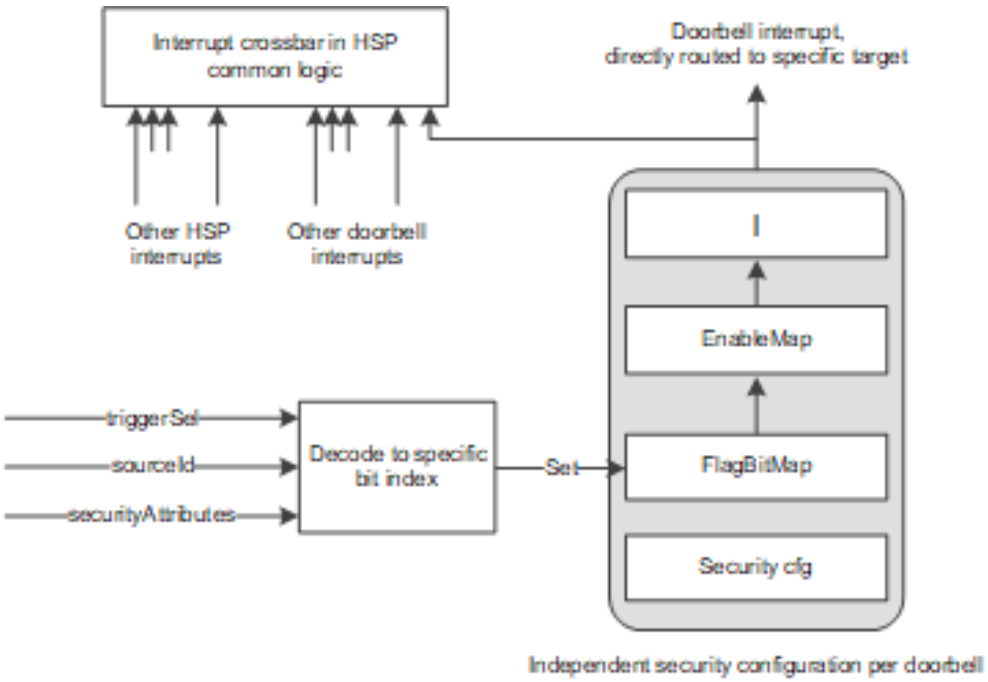
- Standard security configuration register to specify the exact security model
- Bitmap indicating which sources can raise the doorbell signal and at which security level
- Status/control register indicating the set of sources with pending flags (clear on write)
- Status/control register indicating the raw bitmap (clear on write)

The control logic then generates one doorbell signal per doorbell. The doorbell signals are:

- Exposed as a port with a direct connection to the target processor cluster interrupt controller
  - VIC inside a Cortex-R5 cluster
  - GIC inside the APE cluster
  - LIC for CCPLEX, and from there directly to the GIC, as the GIC is essentially transparent for the CCPLEX path
  - CCPLEX power management
  - CCPMU register for the CCPMU doorbell
- Exposed to the interrupt crossbar logic in the HSP, so that it can be mapped as a HSP shared interrupt, allowing the doorbell to be routed to a different target.

The operation of one doorbell is illustrated in the figure below:

Figure 8.56 Doorbell Logic



Doorbells are identified by numbers, with the following mapping between a doorbell number and its intended destination.

Table 8.61 Mapping between Doorbell Number and Usage

Doorbell	Target
0	CCPMU
1	CCPLEX TrustZone not secure
2	CCPLEX TrustZone secure
3	BPMP
4	AON
5	SCE
6	APE
7	RCE
8	DCE
9	PSC

8.6.2.2 Multiple Processors Access to Shared Resources

The SoC does not support full coherency mechanisms, so an IPC protocol must tackle ordering and visibility issues. The protocol uses a doorbell interrupt to get attention and a standard structure of circular buffers plus associated read/write pointers in shared memory (internal or external). With this model, the expected software sequence looks like the following:

8.6.2.2.1 Source Processor

1. Writes a set of messages with known format in shared memory circular buffer structure
2. If the shared memory is cacheable, flushes its cache to make the messages globally visible
3. Issues a DMB to ensure that pointer and message are ordered
4. Writes the value of the write pointer at known location in shared memory
5. If the shared memory is cacheable, flushes its cache
6. Issues a DMB to ensure ordering between pointer and interrupt
7. Writes to a doorbell register in MMIO space to trigger the interrupt to the destination

8.6.2.2.2 Destination Processor

1. Receives doorbell interrupt and enters Interrupt Service Routine
2. Reads the associated doorbell status register to identify the set of source processors with pending messages

3. Cache invalidates the address used by the write pointer for the source processor circular buffer
4. Issues a DMB
5. Read the write pointer
6. Cache invalidates the addresses used by the set of messages identified by the write pointer
7. Issues a DMB
8. Reads the set of messages

8.6.2.3 Using Doorbells for Handshake During Boot Sequence

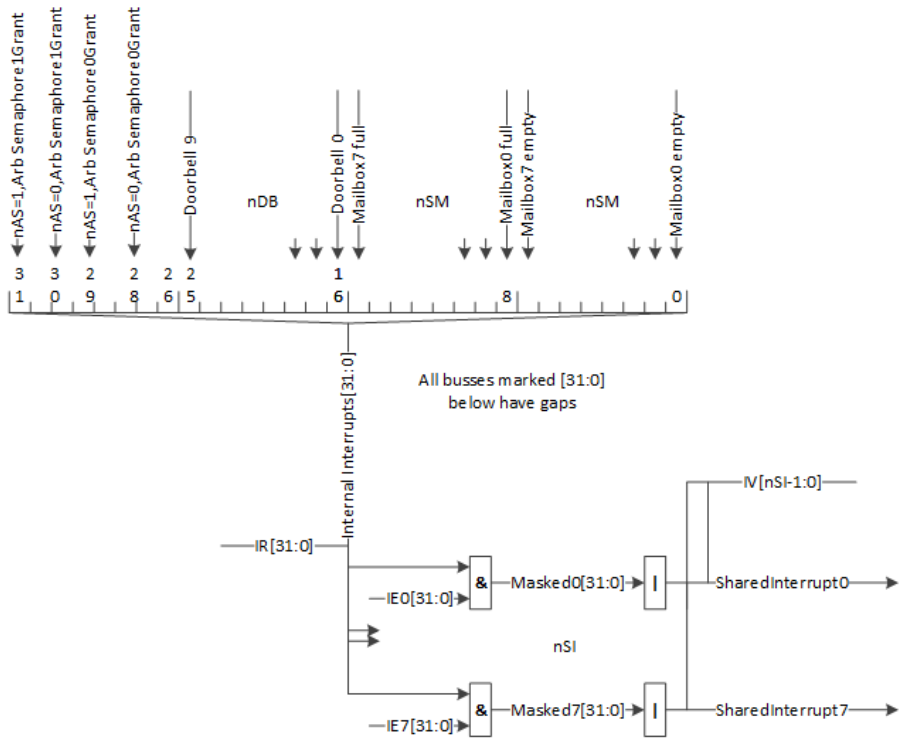
During the boot sequence, the doorbells are used to synchronize the boot sequence across multiple processors. Each doorbell allows for a full handshake between two processors, denoted as B (normally BPMP) and D (normally CCPLEX power management). The handshake makes sure that D executes a specific task A at a time controlled by B and works like this.

- B performs all tasks that must be done before task A can be done. When everything is ready, B signals to D that it can proceed by ringing D doorbell (i.e., writing to the doorbell trigger register). B can continue working on other tasks not dependent on task A completion. When it reaches a point that requires task A to be complete, B check the doorbell pending status. If the doorbell is still asserted, B waits for completion by polling on the doorbell status.
- D performs all tasks that must be done before task A can be done. When everything is ready D checks if its doorbell is asserted. If not, it polls on its doorbell until assertion.
- D performs task A, then signals completion to B by clearing the doorbell (i.e., by writing to the doorbell pending status). Clearing the doorbell is acting as an acknowledge by D.
- When B sees the doorbell pending status deasserted, it knows that task A is complete and can now execute tasks dependent on task A completion.

8.6.2.4 Shared Interrupts Configuration

Each shared interrupt is the aggregate of the selected bits; in other words, each shared interrupt is driven by the same structure of AND followed by OR used in LIC. The figure below shows the logic for the Top0 HSP instance.

Figure 8.57 Shared Interrupt Logic



Assume that software wants to map the shared interrupts as follows:

- Shared0 is a (aggregated) mailbox empty, so IE0 = 0xFF
- Shared1 is a (aggregated) mailbox full, so IE1 = 0xFF << 8
- Shared2 is Arbitrated Semaphore 0, and Arbitrated Semaphore 1 is not used, so IE2 = 0x1 << 28

8.6.2.5 Programming Model

The programming model is register based.

8.6.3 Programming Guidelines

8.6.3.1 Shared Mailboxes

A shared mailbox serves a unidirectional communication between a producer and a consumer. The two masters agree upon a mailbox to use, and which master is the producer and which master is the consumer. In the following sequences, it is assumed that shared mailbox {sm} is selected.

8.6.3.1.1 Producer Sending a Message

1. Make sure the selected shared mailbox is empty by checking the TAG field of register HSP_SHRD_MBOX_MBOX_{sm}_SHRD_MBOX_0 or HSP_SHRD_MBOX_MBOX_{sm} _SHRD_MBOX_TYPE1_TAG_0 register for 128-bit mailbox.
2. Write the message in the DATA field of register HSP_SHRD_MBOX_MBOX_{sm} _SHRD_MBOX_0 or HSP_SHRD_MBOX_MBOX_{sm}_SHRD_MBOX_TYPE1_DATA*_0 registers for 128-bit mailbox. Set the TAG field of the register to 1.

8.6.3.1.2 Consumer Receiving a Message

1. Read register HSP_SHRD_MBOX_MBOX_{sm} _SHRD_MBOX_0 or HSP_SHRD_MBOX_MBOX_{sm}_SHRD_MBOX_TYPE1_TAG_0 register for 128-bit mailbox. If TAG field is 1, then the DATA field carries the message.
2. After consuming the message in the mailbox, set the TAG field of the register to 0. The DATA field is don't care and can be written with any value.

8.6.3.2 Shared Semaphores

A producer/consumer pair uses shared semaphore to synchronize management of a shared resource. Each bit in a shared semaphore register represents a shared semaphore. The producer and consumer select a shared semaphore to use (a particular shared semaphore register and a particular bit) and agree on their roles.

In the following sequences, we assume bit {b} of shared semaphore register {ss} is selected.

8.6.3.2.1 Producer

1. Make sure the shared semaphore is empty by checking that bit {b} in HSP_SHRD_SEM_{ss} _SHRD_SMP_STA_0 is 0.
2. The Producer writes to the shared resource.
3. Indicate the data in the shared resource is ready for consumption by writing 1 to bit {b} in HSP_SHRD_SEM_{ss}_SHRD_SMP_STA_SET_0.

8.6.3.2.2 Consumer

1. The shared resource is ready for consumption when reading HSP_SHRD_SEM_{ss} _SHRD_SMP_STA_0 return value 1 in bit {b}.
2. The Consumer consumes the data in the shared resource.
3. Indicate that the shared resource is available for new data by writing 1 to bit {b} in HSP_SHRD_SEM_{ss}_SHRD_SMP_STA_CLR_0.

8.6.3.3 Arbitrated Semaphores

Each arbitrated semaphore is accessed through two register sets, indexed by 0 and 1; one set for each master sharing the arbitrated semaphore. Each bit in the arbitrated semaphore register corresponds to an arbitrated semaphore. The two masters agree on an arbitrated semaphore to use (a particular arbitrated semaphore register and a particular bit), and agree on which master takes register set 0 and which master takes register set 1.

In the following sequences, we assume bit {b} of the arbitrated semaphore register {as} is selected.

8.6.3.3.1 Requesting Semaphore with Interrupt

1. Enable the ownership interrupt by setting the corresponding bit in HSP_ARB_SEM{0,1}_{as}_INT_EN_0.
2. Set the corresponding bit in HSP_ARB_SEM{0,1}_{as}_SMP_GET_0 to 1.
3. Wait for the interrupt. When that happens, the semaphore is acquired.

8.6.3.3.2 Requesting Semaphore with Busy Spin Lock

1. Set bit {b} in HSP_ARB_SEM{0,1}..{as}_SMP_GET_0 to 1.
2. Poll register HSP_ARB_SEM{0,1}..{as}_SMP_GNT_ST_0. When the bit {b} is 1, the semaphore is acquired.

8.6.3.3.3 Relinquishing Semaphore

1. Set bit {b} in HSP_ARB_SEM{0,1}_{as}_SMP_PUT_0 to 1.

8.6.3.4 Doorbells

A triggering master can ask for attention from the receiving master by ringing the doorbell of the receiving master. In the following sequences, we assume {db} is the doorbell index of the receiving master. There is a dedicated doorbell for each master in TOP0_HSP.

8.6.3.4.1 Ringing a Doorbell

1. Check that the receiving master receives the doorbell by ensuring that the bit corresponds to the triggering master and its security level in HSP_DBELL_<db>_ENABLE_0 for masters with MasterID 0-15 and HSP_DBELL_<db>_ENABLE_SLICE1_0 for masters with MasterID 16-31.
2. Write to HSP_DBELL_{db}_TRIGGER_0. Any value can be written.

8.6.3.4.2 Answering a Doorbell

After servicing a doorbell, the receiving master clears the doorbell from a particular master and security level by writing 1 to the corresponding bit in HSP_DBELL_{db}_PENDING_0 registers for masters with MasterId 0-15 and HSP_DBELL_<db>_PENDING_SLICE1_0 for masters with MasterID 16-31.

8.6.4 HSP Registers

The set of registers is split across the different modules, each set share a common base. Each module has a number of 64-KiB pages to help in virtualization in Orin and all of the modules expose their registers via an APB interface. Each submodule in HSP has its own set of 64-KiB pages plus an extra 64-KiB page for the common registers. The different windows are placed one after the other in this order:

- One 64-KiB page for the common registers
- nSM/2 64-KiB pages for the shared mailboxes, grouped in pairs to support bidirectional communication. The number of mailboxes in a given HSP must be even.
- nSS 64-KiB pages for the shared semaphores
- nAS 64-KiB pages for the arbitrated semaphores
- One 64-KiB page for the doorbells.

All these are referred to a base address shown as HSP_{inst}_BASE in this document, where {inst} identifies the specific HSP instance.

All transactions sent to a given module must be terminated, if an address is not assigned or if the access violates some access restrictions, an error response is sent and the access does not affect any state outside of the APB interface itself.

8.6.4.1 Common Registers

Common registers are used for two independent reasons:

1. For truly common functionality
2. To increase the functionality while keeping the legacy architecture. This is done for security handling. It is recommended for some modules are treated as one security group, with the security handling performing in a wrapper that instantiates the legacy code.

The common registers offset is from HSP_{inst}_BASE. The common registers consist of Security Control Registers (SCRs) and HSP Interrupt registers:

- HSP_SCR_SS_{ss}_REG_0
- HSP_SCR_AS_{as}_REG_0
- HSP_SCR_SM_{sm}_REG_0

- HSP_SCR_DBELL_{db}_REG_0
- HSP_INT_SCR_SCR_C0_REG_0
- HSP_INT_IE{i}_0 (register array)
- HSP_INT_IV_0
- HSP_INT_IR_0

8.6.4.2 HSP Interrupt Registers

The mapping of the HSP internal interrupts is defined as follows:

- Bits[00 + [nSM-1:0]] are the empty interrupts of the nSM instantiated shared mailboxes
- Bits[08 + [nSM-1:0]] are the full interrupts of the nSM instantiated shared mailboxes
- Bits[16 + [nDB-1:0]] are the interrupts of the nDB instantiated doorbells
- Bits[28 + [nAS-1:0]] are the grant interrupts of the nAS instantiated arbitrated semaphores

Bits not defined by the mapping above are marked reserved and not physically present. This is true for both IE{i} and IR registers. Note that this mapping means that nSM, nDB cannot exceed eight and nAS cannot exceed four, for a given HSP instance.

These registers have the same bit mapping, shown in the following table.

Table 8.62 Format of the IE{i} and IR Interrupt Registers

Bit	Reset	Description
31:30	0x0	smp1_gnt_enable/valid/asserted
29:28	0x0	smp0_gnt_enable/valid/asserted
25:16	0x0	dbell_enable/valid/asserted
15:8	0x0	mbox_full_enable/valid/asserted
7:0	0x0	mbox_empty_enable/valid/asserted

8.6.4.2.1 HSP_INT{i}_IE_0

This is an array of eight identical register entries. This array controls the routing of interrupt {i}, a bit mask indicating which of the internal interrupts is propagated to external interrupt {i}, {i} in [0:nSI-1].

ExternalInterrupts[i] = |(IE{i} & InternalInterrupts}

See the Format of the IE{i} and IR Interrupt Registers table for the organization of the register fields.

Offset: 0x100..0x11f | Read/Write: R/W | SCR Protection: SCR_C0_REG_0 | Reset: 0x00000000

8.6.4.2.2 HSP_INT_IV_0

This register contains the currently asserted shared interrupts, ExternalInterrupts.

Each bit of IV is mapped to the corresponding shared interrupt.

Offset: 0x300 | Read/Write: RO | SCR Protection: SCR_C0_REG_0 | Reset: 0x00000000

8.6.4.2.3 HSP_INT_IR_0

This register indicates the currently asserted internal interrupts, InternalInterrupts.

See the Format of the IE{i} and IR Interrupt Registers table for the organization of the register fields.

Offset: 0x304 | Read/Write: RO | SCR Protection: SCR_C0_REG_0 | Reset: 0x000000ff

8.6.4.2.4 HSP_INT_HSP_CLK_OVR_0

Offset: 0x30c | Read/Write: R/W | Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	hsp_clk_ovr_on

8.6.4.2.5 HSP_INT_DIMENSIONING_0

Offset: 0x380 | Read/Write: RO | Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
19:16	X	nSI: Number of shared interrupts in this instance
15:12	X	nDB: Number of doorbells in this instance
11:8	X	nAS: Number of arbitrated semaphores in this instance
7:4	X	nSS: Number of shared semaphores in this instance
3:0	X	nSM: Number of shared mailboxes in this instance

Shared Mailbox Registers

Shared mailboxes are placed 32 KiB apart, so that a pair is mapped in a 64 KiB page. The base for a given mailbox is SM{sm}*BASE{sm} = HSP*{inst}_BASE + 64 KiB + {sm} * 32 KiB.

Note: Offset mentioned below is from SM{0}_BASE

32-bit Shared Mailbox Registers

8.6.4.2.6 HSP_SHRD_MBOX_MBOX_{sm}_SHRD_MBOX_0

Software-controlled mailbox registers. There are eight shared mailbox registers, where {sm} = 0 through 7.

Offset: 0x0 + (sm * 0x8000) | Read/Write: R/W | SCR Protection: SM_{sm}_REG_0 | Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0	TAG: This bit acts as a Valid bit for the mailbox. The mailbox is deemed full when TAG = 1b and empty when TAG = 0b, with the status also reflected in corresponding interrupts. The producer writes TAG to 1b to signal the availability of a message; the consumer writes TAG to 0b to signal that the message has been fully handled.
30:0	0	DATA: Value exchanged between producer and consumer. No hardware semantic.

8.6.4.2.7 HSP_SHRD_MBOX_MBOX_{sm}_SHRD_MBOX_FULL_INT_IE_0

Offset: 0x4 + (sm * 0x8000) | Read/Write: R/W | Parity Protection: N | SCR Protection: SM_{sm}_REG_0 | Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	0x1	ENABLE: Mailbox full interrupt enable

8.6.4.2.8 HSP_SHRD_MBOX_MBOX_{sm}_SHRD_MBOX_EMPTY_INT_IE_0

Offset: 0x8 + (sm * 0x8000) | Read/Write: R/W | Parity Protection: N | SCR Protection: SM_{sm}_REG_0 | Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	0x1	ENABLE: Mailbox empty interrupt enable

128-bit Shared Mailbox Registers

8.6.4.2.9 HSP_SHRD_MBOX_MBOX_{sm}_SHRD_MBOX_TYPE1_TAG

Offset: 0x40 + (sm * 0x8000) | Read/Write: R/W | Parity Protection: N | SCR Protection: SM_{sm}_REG_0 | Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bits	Reset	Description
31	0	TAG: The bit acts as a Valid bit for the 128-bit mailbox. The mailbox is deemed full when TAG = 1b and empty when TAG = 0b, with the status also reflected in corresponding interrupts. The producer writes TAG to 1b to signal the availability of a message, the consumer write TAG to 0b to signal that the message has been fully handled.

8.6.4.2.10 HSP_SHRD_MBOX_MBOX_{sm}_SHRD_MBOX_TYPE1_DATA0

Offset: 0x48 + (sm * 0x8000) | Read/Write: R/W | Parity Protection: N | SCR Protection: SM_{sm}_REG_0 | Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bits	Reset	Description
31:0	0	DATA: First 4-byte chunk out of maximum 16B data to be exchanged between producer and consumer, no Hardware semantic.

8.6.4.2.11 HSP_SHRD_MBOX_MBOX_{sm}_SHRD_MBOX_TYPE1_DATA1

Offset: 0x4c + (sm * 0x8000) | Read/Write: R/W | Parity Protection: N | SCR Protection: SM_{sm}_REG_0 | Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bits	Reset	Description
31:0	0	DATA: Second 4-byte chunk out of maximum 16B data to be exchanged between producer and consumer, no Hardware semantic.

8.6.4.2.12 HSP_SHRD_MBOX_MBOX_{sm}_SHRD_MBOX_TYPE1_DATA2

Offset: 0x50 + (sm * 0x8000) | Read/Write: R/W | Parity Protection: N | SCR Protection: SM_{sm}_REG_0 | Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bits	Reset	Description
31:0	0	DATA: Third 4-byte chunk out of maximum 16B data to be exchanged between producer and consumer, no Hardware semantic.

8.6.4.2.13 HSP_SHRD_MBOX_MBOX_{sm}_SHRD_MBOX_TYPE1_DATA3

Offset: 0x54 + (sm * 0x8000) | Read/Write: R/W | Parity Protection: N | SCR Protection: SM_{sm}_REG_0 | Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bits	Reset	Description
31:0	0	DATA: Fourth 4-byte chunk out of maximum 16B data to be exchanged between producer and consumer, no Hardware semantic.

8.6.4.3 Shared Semaphore Registers

Each shared semaphore instance has an associated page starting at SS{ss}*BASE = HSP*{inst}_BASE + (1 + nSM/2) * 64 KiB + {ss} * 64 KiB.

Note: Offset mentioned below is from SS{0}_BASE

8.6.4.3.1 HSP_SHRD_SEM_{ss}_SHRD_SMP_STA_0

There are eight shared semaphore status registers, where {ss} is 0 through 7. Even though we have eight shared semaphore registers, only 0-3 are valid as maximum number of SS is four for Orin. This register gives the current status for the shared semaphore instance.

Offset: 0x000 + (ss * 0x10000) | Read/Write: RO | SCR Protection: SS_{ss}_REG_0 | Reset: 0x00000000

Bit	Reset	Description
31:0	0	SMP: Current value. Each bit is an individual semaphore.

8.6.4.3.2 HSP_SHRD_SEM_{ss}_SHRD_SMP_STA_SET_0

There are eight shared semaphore set registers, where {ss} is 0 through 7. Set the bits in this register to update the current value.

Offset: 0x004 + (ss * 0x10000) | Read/Write: WO | SCR Protection: SS_{ss}_REG_0 | Reset: NA

Bit	Reset	Description
31:0	NA	SMP_SET: Writing bit i to 1b sets semaphore bit i. Undefined for reads.

8.6.4.3.3 HSP_SHRD_SEM_{ss}_SHRD_SMP_STA_CLR_0

There are eight shared semaphore clear registers, where {ss} is 0 through 7. Clear the bits in this register to update the current value.

Offset: 0x008 + (ss * 0x10000) | Read/Write: WO | SCR Protection: SS_{ss}_REG_0 | Reset: NA

Bit	Reset	Description
31:0	NA	SMP_CLR: Writing bit i to 1b clears semaphore bit i. Undefined for reads.

8.6.4.4 Arbitrated Semaphore Registers

Arbitrated semaphore instance {as} has a register range starting at AS{as}*BASE = HSP*{inst}_BASE + (1 + nSM/2 + nSS) * 64 KiB + {as} * 64 KB. Each arbitrated semaphore instance contains two apertures, identified as {a} in {0,1}, corresponding to two different owners. The owner associated with an aperture is defined by software convention. The following table shows the base offset according to arbitrated semaphore instance and aperture.

Note: Offset mentioned below is from AS{0}_BASE

Table 8.63 Base Offsets of Arbitrated Semaphore Registers

Semaphore Instance {as}	Base Offset	
	Aperture 0 {a} = 0	Aperture 1 {a} = 1
0	0x00	0x20
1	0x10000	0x10020
2	0x20000	0x20020
3	0x30000	0x30020

8.6.4.4.1 HSP_ARB_SEM{a}_{as}_SMP_GNT_ST_0

This register indicates current ownership; each bit is an individual semaphore. There are four Arbitrated Semaphore Grant Status registers, where {as} = semaphore 0 through 3 and {a} = aperture 0 or 1.

Offset: Base | Read/Write: RO | SCR Protection: AS_{as}_REG_0 | Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0	STATUS: Bit i at 1 indicates aperture{a} is the current owner of resource i

8.6.4.4.2 HSP_ARB_SEM{a}_{as}_SMP_GET_0

There are four Arbitrated Semaphore Get registers, where {as} = s emaphore 0 through 3 and {a} = aperture 0 or 1. Even though we have four Arbitrated semaphore registers, only 0-1 are valid as maximum number of AS is two for Orin. Use this register to make a request for ownership.

Offset: Base + 0x4 | Read/Write: WO | SCR Protection: AS_{as}_REG_0 | Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	NA	GET: Writing bit i to 1b is a request for ownership of resource i from aperture {a}. All pending requests are accumulated in an internal register visible as HSP_ARB_SEM{a}_{as}_REQ_ST_0. Requests cannot be canceled; a request bit gets cleared when ownership is acquired.

8.6.4.4.3 HSP_ARB_SEM{a}_{as}_SMP_PUT_0

There are four Arbitrated Semaphore Put registers, where {as} = semaphore 0 through 3 and {a} = aperture 0 or 1. Use this register to relinquish ownership.

Offset: Base + 0x8 | Read/Write: WO | SCR Protection: AS_{as}_REG_0 | Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	NA	PUT: Writing bit i to 1b relinquishes ownership of resource i from aperture {a}.

8.6.4.4.4 HSP_ARB_SEM{a}_{as}_SMP_REQ_ST_0

There are four Arbitrated Semaphore Request registers, where {as} = semaphore 0 through 3 and {a} = aperture 0 or 1. This register indicates if there are any pending requests.

Offset: Base + 0xc | Read/Write: RO | SCR Protection: AS_{as}_REG_0 | Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0	REQ: Bitmap of pending requests from aperture {a}.

8.6.4.4.5 HSP_ARB_SEM{a}_{as}_SMP_INT_EN_0

There are four Arbitrated Semaphore Interrupt Enable registers, where {as} = semaphore 0 through 3 and {a} = aperture 0 or 1.

Offset: Base + 0x10 | Read/Write: R/W | SCR Protection: AS_{as}_REG_0 | Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

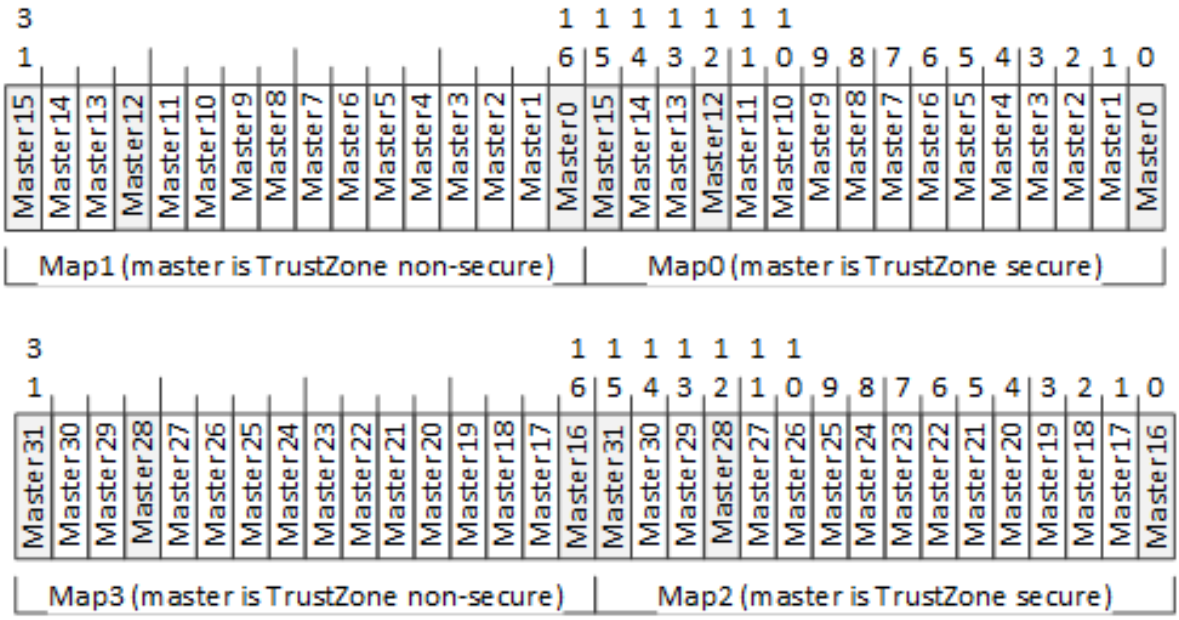
Bit	Reset	Description
31:0	0	EN: Interrupt enable. The interrupt is active when resource i is granted to aperture {a} and the corresponding enable bit is 1.

8.6.4.4.6 **Doorbell Registers**

All doorbell registers are in a single page, doorbell {db} has a register range starting at DB{db}*BASE = HSP*{inst}_BASE + (1 + nSM/2 + nSS + nAS) * 64 KiB + {db} * 0x100.

In the Enable, Raw, and Pending registers, each field is defined as a bitmap, where each bit is associated with a master ID. Only the bits associated with valid Master ID are guaranteed to be present. Two of these are packed per register to provide fast access; each map is denoted as MAP{sl}. The variable {sl} takes values 0 (TrustZone secure) and 1 (TrustZone non-secure).

Figure 8.58 Format for Doorbell Registers Using Bitmaps



Note: Offset mentioned below is from DB{0}_BASE

8.6.4.4.7 HSP_DBELL_{db}_TRIGGER_0

In this write-only register, set a pending flag based on the source ID and the security level of the transaction. There are 10 doorbell trigger registers, where {db} is a value from 0 to 9.

Offset: 0x0 + (db * 0x100) | Read/Write: WO | SCR Protection: None | Reset: NA

Bit	Reset	Description
31:0	NA	DUMMY: Any write to this register sets bit Raw_{reg_slice}{db}_{sl}[{src}], where {reg_slice} = 0 for master ID 0-15 and {reg_slice} = 1 for master ID 16-31 {src} = master ID {s1} = Map0, for master ID 0-15, TrustZone secure {s1} = Map1, for master ID 0-15, TrustZone non-secure {s1} = Map2, for master ID 16-31, TrustZone secure {s1} = Map3, for master ID 16-31, TrustZone non-secure

8.6.4.4.8 HSP_DBELL_{db}_ENABLE_0

This register indicates which pending flags can raise the doorbell signal. There are ten doorbell enable registers, where {db} is a value from 0 to 9. This register is used for MasterIDs <16(0 to 15).

Offset: 0x4 + (db * 0x100) | Read/Write: R/W | SCR Protection: DB_{db}_REG_0 | Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0	MAP1: Which bits in in Raw_{reg_slice}{db}_{sl} *can raise signal doorbell*{db}_{sl} where {reg_slice} = 0 for master ID 0-15 and {reg_slice} = 1 for master ID 16-31. This field is valid when {sl} = Map1, indicating the master is TrustZone non-secure).
15:0	0	MAP0: Which bits in Raw_{reg_slice}{db}_{sl} *can raise signal doorbell*{db}_{sl} where {reg_slice} = 0 for master ID 0-15 and {reg_slice} = 1 for master ID 16-31. This field is valid when {sl} = Map0, indicating the master is TrustZone secure).

8.6.4.4.9 HSP_DBELL_{db}_ENABLE_SLICE1_0

This register indicates which pending flags can raise the doorbell signal. There are ten doorbell enable registers, where {db} is a value from 0 to 9. This register is used for 15< MasterIDs <32(16 to 31).

Offset: 0x24 + (db * 0x100)
Read/Write: R/W
SCR Protection: HSP_DBELL_SCR_DB_0_REG_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	MAP1: Which bits in Raw_{reg_slice}{db}_{sl} *can raise signal doorbell*{db}_{sl} where {reg_slice} = 0 for master ID 0-15 and {reg_slice} = 1 for master ID 16-31. This field is valid when {sl} = Map3, indicating the master is TrustZone non-secure).
15:0	0x0	MAP0: Which bits in Raw_{reg_slice}{db}_{sl} *can raise signal doorbell*{db}_{sl} where {reg_slice} = 0 for master ID 0-15 and {reg_slice} = 1 for master ID 16-31. This field is valid when {sl} = Map2, indicating the master is TrustZone secure).

8.6.4.4.10 HSP_DBELL_{db}_RAW_0

This register indicates which flags have been set as a result of a write to the {db}_TRIGGER register. This register is used for MasterIDs <16(0 to 15).

Offset: 0x8 + (db * 0x100) | Read/Write: RWC | SCR Protection: DB_{db}_REG_0 | Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0	MAP1: Which flags are currently set; writing 1 to a bit clears it. This field is valid when {sl} = Map1, indicating the master is TrustZone non-secure).
15:0	0	MAP0: Which flags are currently set; writing 1 to a bit clears it. This field is valid when {sl} = Map0, indicating the master is TrustZone secure).

8.6.4.4.11 HSP_DBELL_{db}_RAW_SLICE1_0

This register indicates which flags have been set as a result of a write to the {db}_TRIGGER register. This register is used for 15 < MasterIDs <32(16 to 31).

Offset: 0x28+ (db * 0x100)
Read/Write: R/W
SCR Protection: HSP_DBELL_SCR_DB_0_REG_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	MAP1: Which flags are currently set; writing 1 to a bit clears it. This field is valid when {sl} = Map3, indicating the master is TrustZone non-secure).
15:0	0x0	MAP0: Which flags are currently set; writing 1 to a bit clears it. This field is valid when {sl} = Map2, indicating the master is TrustZone secure).

8.6.4.4.12 HSP_DBELL_{db}_PENDING_0

This register indicates which flags are set and enabled, equivalently the input to the OR reduce function that controls the signal doorbell_{db}. The read value for {i} is not guaranteed after execution of an operation on {i}. This register is used for MasterIDs <16(0 to 15).

Offset: 0xc + (db * 0x100) | Read/Write: RWC | SCR Protection: DB_{db}_REG_0 | Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0	MAP1: Which enabled flags are currently set. Writing 1 to a bit clears the equivalent bit in the HSP_DBELL_{db}_RAW_0 register, and thus implicitly clears the same bit in this register. This field is valid when {sl} = Map1, indicating the master is TrustZone non-secure).
15:0	0	MAP0: Which enabled flags are currently set. Writing 1 to a bit clears the equivalent bit in the HSP_DBELL_{db}_RAW_0 register, and thus implicitly clears the same bit in this register. This field is valid when {sl} = Map0, indicating the master is TrustZone secure).

8.6.4.4.13 HSP_DBELL_{db}_PENDING_SLICE1_0

This register indicates which flags are set and enabled, equivalently the input to the OR reduce function that controls the signal doorbell_{db}. The read value for {i} is not guaranteed after execution of an operation on {i}. This register is used for 15 < MasterIDs <32(16 to 31).

Offset: 0x2c+ (db * 0x100)
Read/Write: R/W
SCR Protection: HSP_DBELL_SCR_DB_0_REG_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	MAP1: Which enabled flags are currently set. Writing 1 to a bit clears the equivalent bit in the HSP_DBELL_{db}_RAW_SLICE1_0 register, and thus implicitly clears the same bit in this register. This field is valid when {sl} = Map3, indicating the master is TrustZone non-secure).
15:0	0x0	MAP0: Which enabled flags are currently set. Writing 1 to a bit clears the equivalent bit in the HSP_DBELL_{db}_RAW_SLICE1_0 register, and thus implicitly clears the same bit in this register. This field is valid when {sl} = Map2, indicating the master is TrustZone secure).

8.6.4.5 Invalid Registers

HSP, like other blocks, will normally create an APB slave error when invalid registers are accessed. However, certain addresses do not do this, as follows:

- Read and write accesses to the following invalid register addresses do not result in APB slave error response.
  0x03c5_0018, 0x03c5_001c, 0x03c5_0020, 0x03c5_0024, 0x03c5_0028, 0x03c5_002c, 0x03c7_0058, 0x03c7_005c, 0x03d5_0020, 0x03d5_0024, 0x03d5_0028, 0x03d5_002c, 0x0165_0020, 0x0165_0024, 0x0165_0028, 0x0165_002c

Note that ranges are mentioned only for TOP0, TOP1, and TOP2_HSP.

Writing and reading to the invalid register addresses above has no side effect on HSP functionality, but failure to report an error could mask programming errors so HSP users should be aware of this.

8.7 Design for Debugging (DFD)

8.7.1 Overview

System and software development may require the ability to study and debug the SOC and its operation, and so dedicated hardware units are provided to aid this debug ability. These debug capabilities are implemented throughout the chip to provide better visibility into its operation. The

additional visibility added via the debug components can also help in performance analysis by monitoring the state of the system in association with the timestamps at which the state was observed.

The Design for Debug (DFD) unit helps provide this additional visibility that helps debug functional and performance issues.

The following aspects are considered in DFD:

- Performance (e.g., software tracing)
- Functionality (e.g., ability to debug and trace hangs)
- Power policies (e.g., ability to debug and trace activity in low-power use cases)
- Control and Data Backbone changes that affect the routing of debug control and data traffic
- Clocking policies, especially default clocks.
- Security

The DFD capability is based on Arm CoreSight 2.0 Architecture. This chapter describes the SoC debug blocks and debugging strategies.

Information in the following documentation is useful to understand how to develop a debugging implementation using DFD. Most of these manuals are available from the public section of the Arm Infocenter website (http://infocenter.arm.com/help/index.jsp) or the Arm Developer website (https://developer.arm.com/documentation); others you may have to order from Arm. The versions shown are the most recent at the time of writing, however if a newer version is available, that should be used.

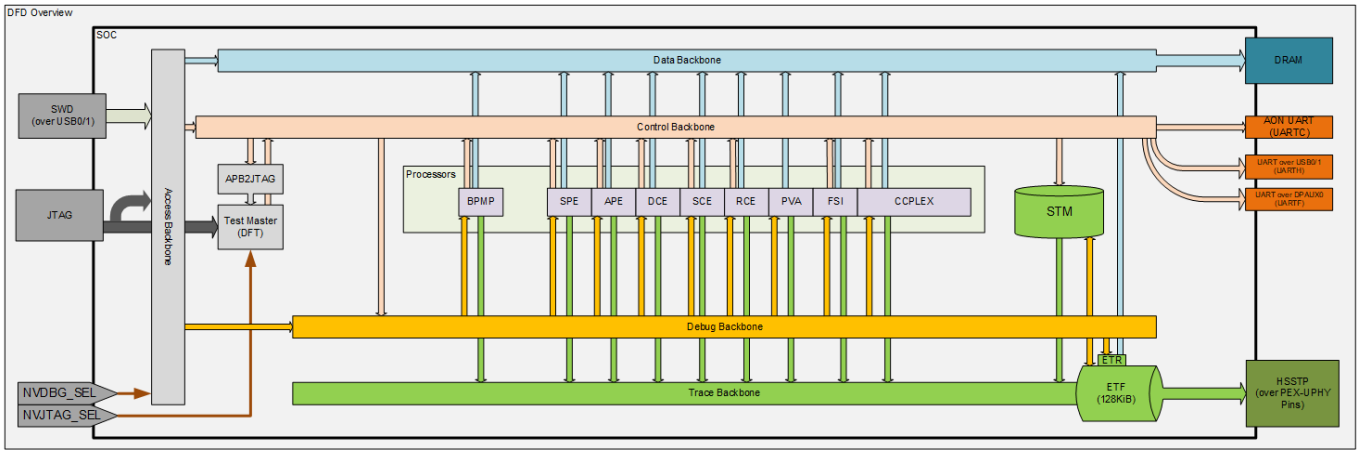
Document	Description
CoreSight SoC-400 Revision: r3p2 Technical Reference Manual (TRM)	Describes the Arm® CoreSight™ SoC-400 components
CoreSight Trace Memory Controller Revision: r0p1 TRM	Describes the Arm Trace Memory Controller
CoreSight System Trace Macrocell Revision: r0p1 TRM	Describes the Arm System Trace Macrocell (STM)
CoreSight Arch Specification v2.0	
Cortex-A9-PTM (r1p0) TRM	Describes the CoreSight PTM-A9 Program Trace Macrocell (PTM)
Cortex-R5-ETM (r0p0) TRM	Describes the CoreSight Embedded Trace Macrocell (ETM) for the Cortex®-R5 and Cortex-R5F processors, the CoreSight ETM-R5 macrocell
AMBA ATB Protocol Specification	
ARM v8.2 Architecture	Describes the Arm architecture v8.2, Armv8.2
Arm DynamIQ Shared Unit (DSU) TRM	
Arm Cortex-A78C (Hercules) TRM	

Document	Description
Arm Cortex-R52 TRM	
Arm Debug Interface v5	
Arm HSSTP Architecture Specification	
Arm DSTREAM and HSSTP (Probe)	
DSTREAM-HT Getting Started Guide	
DSTREAM-HT System Design Guide	
Xilinx® Aurora 8b10b	

8.7.1.1 System Connectivity Overview

The following figure provides an overview of the system connectivity of DFD in the SoC.

Figure 8.59 System Connectivity



DFD provides two major external debug interfaces into the chip: JTAG and SWD. The two Test Access Ports (TAPs), ArmJTAG (used for debug, also known as CoreSight) and NVJTAG (used for DFT purposes like BIST, ATE, etc.), are muxed over JTAG with NVJTAG_SEL acting as the mux select. When NVJTAG_SEL=1, the mux selects NVJTAG's TAP to communicate over the JTAG interface. SWD is like two-pin JTAG and is available via USB2 pins; it supports communication with DFD and acts as an alternative to ArmJTAG. Using SWD in lieu of ArmJTAG is referred to as SWD-DFD (note that it is not a replacement for NVJTAG). The Access Backbone primarily consists of the CoreSight Debug Access Port (DAP) housing two debug ports (JTAG-DP and SW-DP) and two access ports (AXI-AP and APB-AP).

Debug traffic from ArmJTAG or SWD-DFD is converted to debug-APB traffic using APB-AP, that can access external debug interfaces for Arm processors via the debug-APB network (also known as Debug Backbone) or can also issue direct transactions over Control Backbone (CBB)/Data

Backbone (DBB) using AXI-AP. The debug-APB network acts as a network that is independent of functional networks like CBB/DBB and helps access various debug resources in the chip without disturbing functional traffic. The Debug Backbone primarily consists of APB Interconnects (APBICs) and provides debug access to various CoreSight components and the external debug registers of the Arm processors. CBB is also an initiator on the Debug Backbone and allows on-chip software to perform self-hosted debug.

The Trace Backbone consists primarily of ATB bridges and funnels. The funnels can be configured via the debug backbone. The trace backbone allows traces to be routed to memory (DRAM) via Embedded Trace Router (ETR) and/or external trace port analyzer (TPA) like DSTREAM over the High Speed Serial Trace Port (HSSTP). Processor's execution traces are useful for debugging performance issues. Some processors, like the Cortex-R52, also support data traces. Traces are routed across the chip using the ATB network (also referred to as Trace Backbone), which is also independent of functional networks and provides a transparent way to move trace data across the chip. PTMs/ETMs support execution traces, which provide insight into program flow of a processor without requiring any active involvement of the processor for exporting this information. ETMs that support data tracing (e.g., R52 ETM) also provide insight into data transactions of a processor. STM supports instrumentation traces – these are basically like *printf* statements and, thus, require action by software to send useful debug information. STM allows hardware support for up to 64Ki different sources to stream their instrumentation traces over the ATB network and acts as a good alternative to software-based Unified-UART, which needs active software support and, hence, consumes a processor (SPE). STM also supports higher bandwidth (~100 MB/s) than UART (10-15 KB/s). Trace can be captured on-chip using a 128KiB Embedded Trace FIFO (ETF). It can also be streamed off-chip to DRAM via ETR or an external device like DSTREAM-HT via HSSTP. HSSTP allows capturing traces at rates up to 20Gbps without disrupting any functional traffic over control or data fabric. Thus, HSSTP is a transparent and efficient way to analyze real-time performance issues. HSSTP is available via USB3 pins.

Using SWD over USB2 pins for invasive debug traffic and HSSTP over USB3 pins for noninvasive trace traffic allows debug and high-speed trace in real-time over a USB Type-C connector. Adapters like HT-DAM additionally provide hot-plug support to dynamically switch the USB Type-C port from USB to SWD+HSSTP.

DFD-RAMDUMP helps flush CCPLEX-cache contents in DRAM and retain DRAM across Level-2 (L2cold) reset. The DRAM contents can be extracted post L2 reset, to analyze it for causes related to the hang that led to a system crash. DFD-RAMDUMP is useful in catastrophic events that lead to hard-hangs and system crashes when software is unable to service watchdog timers (WDTs). DFD-RAMDUMP should not be confused with DFT-RAMDUMP, which extracts on-chip RAM contents via NVJTAG.

The Cross-Trigger network helps convey debug state, events, and information between different components.

Features like ACCESS_BLOCK and ACCESS_TIMEOUT enable recovering the debug-APB interface from a transaction to a non-responsive target.

8.7.1.2 Summary of Major Features

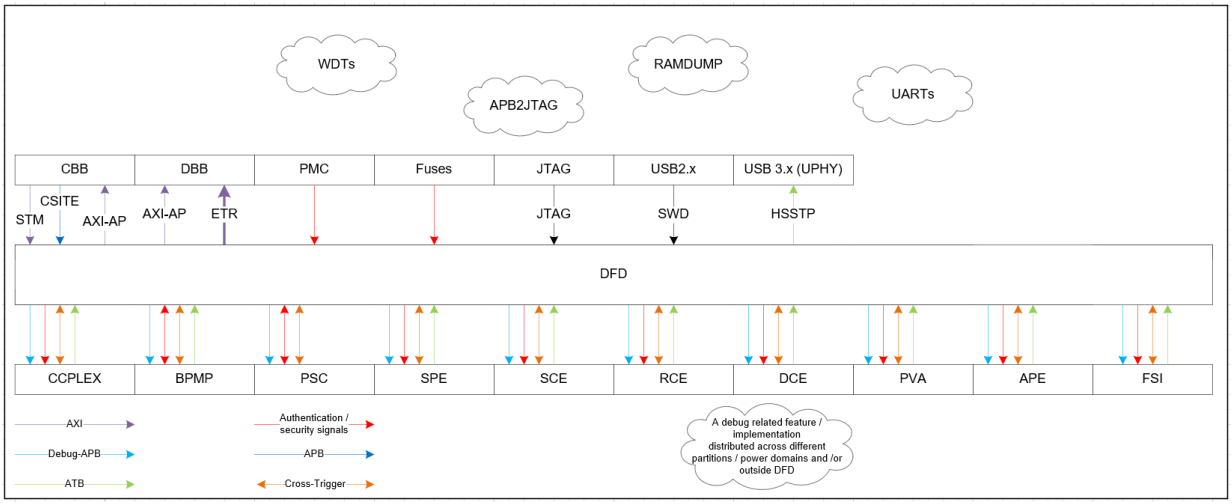
- Arm CoreSight SoC-400 based debug and trace support with STM and TMC
- External Debug interfaces supported – JTAG, SWD
- External Trace interfaces supported – HSSTP
- Closed Box Debug and Trace support – SWD and HSSTP via USB Type-C interface, UART via USB or DPAUX interface
- DFD-RAMDUMP to recover DRAM and CCPLEX cache contents after a system reset in the event of a system-hang/system-crash.
- Watchdog scheme to support recovery of on-chip contents (context/cache/RAMs) and memory (DRAM) in case of a hang.
- SC7 Debug mode
- AON UART for unified cluster logging

8.7.2 Functional Description

The following is an overview of the CoreSight based implementation for the SoC.

8.7.2.1 CoreSight

Figure 8.60 CoreSight Connectivity Overview



8.7.2.1.1 CoreSight Major and CoreSight Minor Topology

CoreSight is primarily divided into CoreSight Major and CoreSight Minor blocks. CoreSight Major contains most of the top-level components like the top APBIC and ROM table, and it aggregates the trace and routes it to various trace sinks (e.g., ETF, HSSTP, DRAM/memory). CoreSight Minor

serves the purpose of aggregating interfaces between various auxiliary clusters and CoreSight Major. The following figures give an overview of the CoreSight Major and CoreSight Minor blocks.

Figure 8.61 CoreSight Major

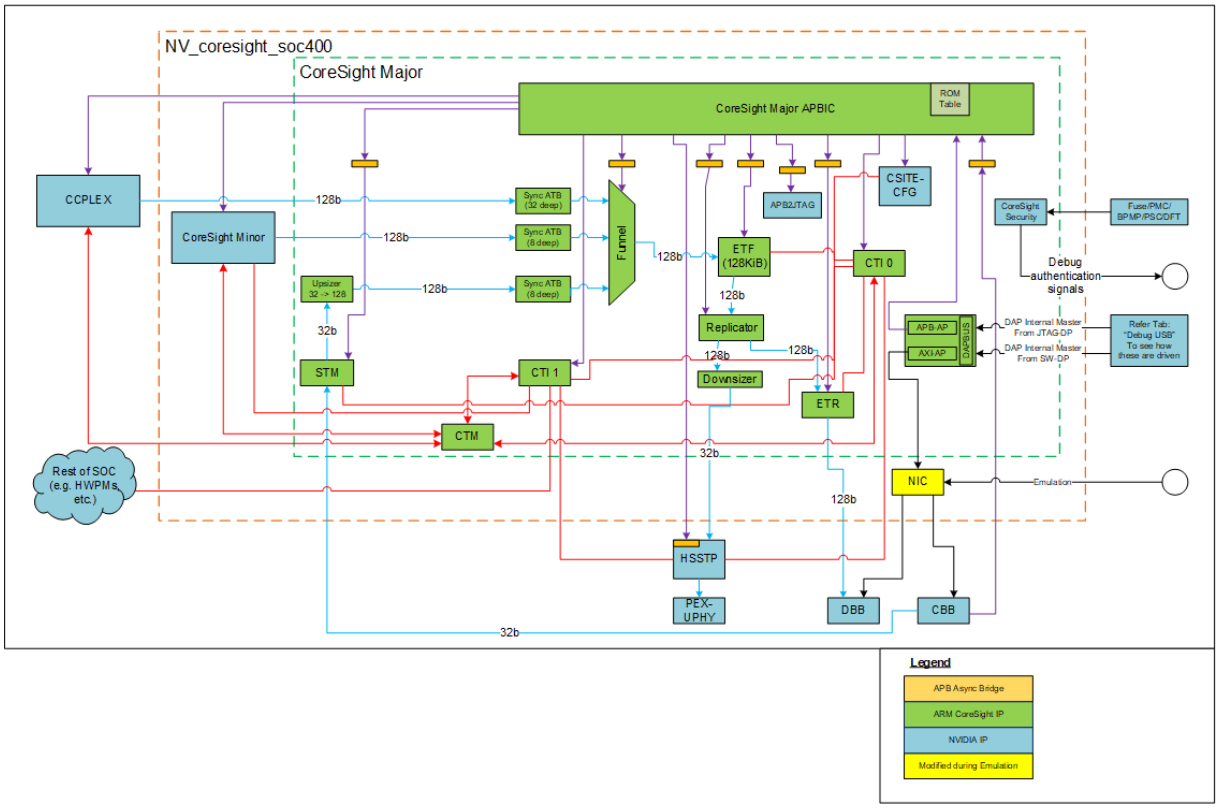
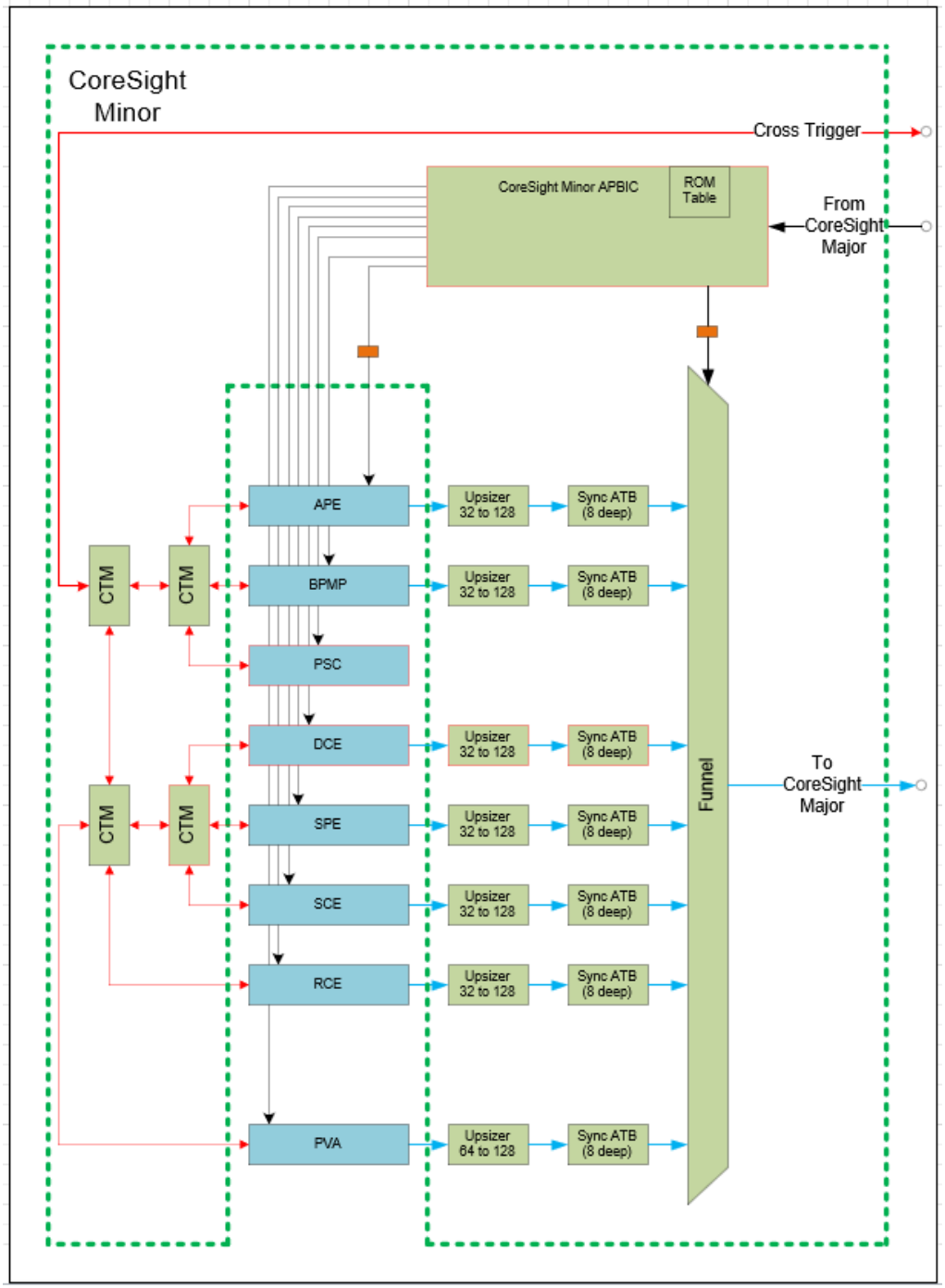


Figure 8.62 CoreSight Minor

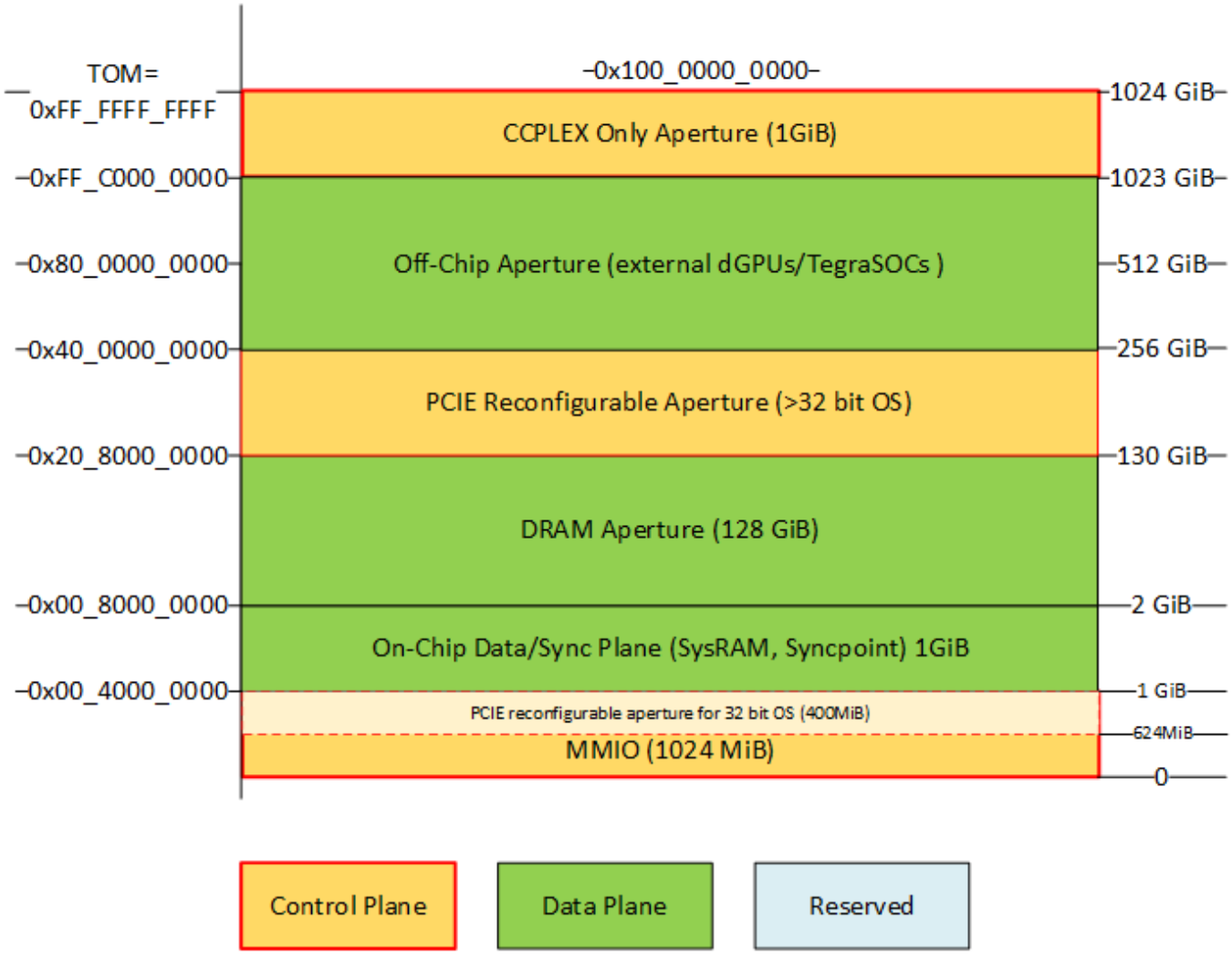


System AMAP and Routing within DFD NIC

Refer to the System AMAP for details. The following subsections emphasize the DFD related apertures in the System AMAP.

Below is a snippet of the System AMAP. Accesses from AXIAP in the address ranges in orange are routed by DFD's 1x2 NIC to CBB. The other accesses are routed by the DFD NIC to DBB. Also, DRAM locations (and other locations accessible to AXI-AP via DBB) with PA>=512G can only be accessed via SMMU.

Figure 8.63 System AMAP Diagram



Relocation and Overlay of DFD Apertures

The CoreSight Aperture is in physical address range 0x2400_0000 - 0x27FF_FFFF in the System AMAP.

STM's APB-target aperture lies inside the CoreSight Aperture. STM (AXI-Target aperture on CBB) is in the physical address range 0x2500_0000 - 0x25FF_FFFF. The STM-AXI aperture is in an AMAP region of CSITE. Though STM falls in the CSITE aperture range from on-chip software or CBB perspective, STM's AXI aperture is a target on the SOC-CBB, not on the debug-APB bus.

DFD's CoreSight DAPBUS Configuration

- APB-AP is connected to Access Port 0 of DFD's CoreSight DAPBUS
- AXI-AP is connected to Access Port 1 of DFD's CoreSight DAPBUS

8.7.2.1.2 Trace Support

Trace Network Overview

AMBA Trace Bus (ATB) constitutes the primary trace backbone in the system. Trace sources that can dump data over ATB include:

- Cortex-R5 (e.g., BPMP, SPE, SCE, RCE, DCE, PVA0-R5, PVA1-R5) ETMs
- Cortex-A9 (e.g., APE) PTM
- System Trace Macrocell (STM); implemented for software instrumentation purposes
- CCPLEX ETMs

Available trace sinks:

- ETF
  - 128KiB capacity
  - Can be configured as a buffer for on-chip trace storage
  - Typically used when
    - Connection to off-chip TPA is unavailable or has insufficient bandwidth
    - Traffic to DRAM needs to remain undisrupted
    - Trace can fit into 128KiB
    - ETF can be used to absorb bandwidth spikes in trace while streaming trace to DRAM or off-chip TPA
- DRAM
  - Capacity is determined by platform
  - ETR is used to to convert ATB traffic to AXI traffic and streamed via DBB to a GSC carveout. ETR has a dedicated GSC assigned to dump its trace in memory.
  - Typically used when
    - Connection to off-chip TPA is unavailable or has insufficient bandwidth
      - Traffic to DRAM can accommodate trace bandwidth
      - Trace does not fit in ETF
- TPA
  - Acts as an off-chip device that stores the trace sent over an interface
  - Arm's DSTREAM is an example of a TPA, which has a 4GiB storage capacity
  - Typically used when
    - Traffic to DRAM needs to remain undisrupted
    - ETF capacity is insufficient to store trace

- Off-chip trace streaming bandwidth is acceptable
- HSSTP is used to send traffic to TPA. Refer to the HSSTP section for details.

Trace Port Mappings

Table 8.64 Trace Port Mappings

Component	Port	Connected to	Comments
CoreSight Minor Funnel	M0	CoreSight Major Funnel	
	S0	Reserved	
	S1	APE	
	S2	BPMP	
	S3	SPE	
	S4	SCE	
	S5	RCE	
	S6	PVA	
	S7	DCE	
CoreSight Major Funnel	M0	ETF	
	S0	CoreSight Minor Funnel	
	S1	STM	
	S2	CCPLEX Top Funnel	
	S3	FSI Top Funnel	
PVA Collator Funnel	M0	CoreSight Minor Funnel	
	S0	PVA0	
FSI Top Funnel	M0	CoreSight Major Funnel	
	S0	FSI Cluster-0	FSI-R5
	S1	FSI Funnel-1	FSI-R52
FSI Funnel-1	M0	FSI Top Funnel	
	S0	Core-0 I-Trace	
	S1	Core-0 D-Trace	
	S2	Core-1 I-Trace	
	S3	Core-1 D-Trace	
	S4	Core-2 I-Trace	

Component	Port	Connected to	Comments
	S5	Core-2 D-Trace	
	S6	Core-3 I-Trace	
	S7	Core-3 D-Trace	
Replicator	S0	ETF	
	M0	ETR	
	M1	HSSTP	

Tracing to DRAM using ETR

ETR has a dedicated GSC (GSC#15). The size and location of the carveout is based on BCT and can be derived based on the GSC's BOM and SIZE registers.

ETR performance bandwidth requirements necessitate SMMU bypass. However, per security guidelines, accesses that bypass SMMU fall into a GSC. Hence, for performance purposes, ETR needs a GSC. Also, using GSC for ETR traces enables restricting accesses to trace information and avoids on-chip malicious code from snooping trace information.

ETR is categorized as a boot client by MSS. This has an implication that ETRW (ETR write client) PCFIFO interlock is enabled (MC_PCFIFO_CLIENT_CONFIG4_0[PCFIFO_ETRW_ORDERED_CLIENT]=ORDERED) by default out of reset. Software default for this is PCFIFO_ETRW_ORDERED_CLIENT=UNORDERED. PCFIFO_ETRW_ORDERED_CLIENT is programmed to UNORDERED for improved ETR DRAM bandwidth.

To ensure ETR maximizes data throughput, program ETR as follows:

- AXCTL[CacheCtrl0]=AXCTL[CacheCtrl1]=1 (enable buffering and modifiability attributes)
- AXCTL[CacheCtrl2]=AXCTL[CacheCtrl3]=0 (Write-allocate cache does not need to be enabled, as caching does not help ETR write pattern, which skims across the entire range of write-range, which exceeds maximum size of cache)
- WrBurstLen=0xF (maximum burst length)
- ScatterGatherMode=0 (avoid scatter-gather mode)
- RSZ: This is the ETR's target buffer size in DRAM. Recommend values greater than 0x8000. Do not use value less than 0x8000 (128KiB) because, in that case, ETF (128KiB) could be used as a circular buffer instead of streaming to DRAM.

ETR to DRAM throughput is 5.5GB/s.

Instrumentation Traces Using STM

Instrumentation traces refer to explicit software generated logs like printfs. STM is a CBB target. Various system initiators can write their instrumentation traces to STM. STM converts the traces into ATB format and routes them over the ATB fabric to different trace sinks like ETF, DRAM, or external TPA via HSSTP.

The observed CCPLEX to STM bandwidth is 250MB/s when there is no background traffic on the same target bridge and 49MB/s when there is background traffic on the same target bridge.

CoreSight STM Timestamp

The example below shows how to infer the time interval between two observed timestamp values.

In this example, the first timestamp is TS1 and the next timestamp is TS2.

TS_DIFF = TS2 – TS1

A timestamp packet occurs every 32ns and has granularity of 56 bits. The TS interpolator right extends this by 8 bits (2^8=256).

Hence, the timestamp value of the interpolator (output) is

= ((Upper 56 bits + lower 8 bits) / 256) * 32ns

= (64 bit Timestamp Interpolator Output / 256) * 32ns

Hence, time diff = (TS_DIFF / 256) * 32ns

= (TS_DIFF / 256) * 32ns

= (TS_DIFF / 8) ns

8.7.2.1.3 CoreSight Topology for Auxiliary Clusters

The Cortex-R5 based BPMP, SPE , SCE, DCE , and RCE have a Dual Core Lock Step (DCLS) implementation. The SPE consists of a single Cortex-R5 based subsystem, while APE consists of a single Cortex-A9 based subsystem. There is one PVA cluster, which consists of a Cortex-R5 based subsystem and two Vector Processing Units (VPUs). The topologies for these clusters are described in the figures below.

Figure 8.64 Cortex-R5 Cluster with DCLS Implementation

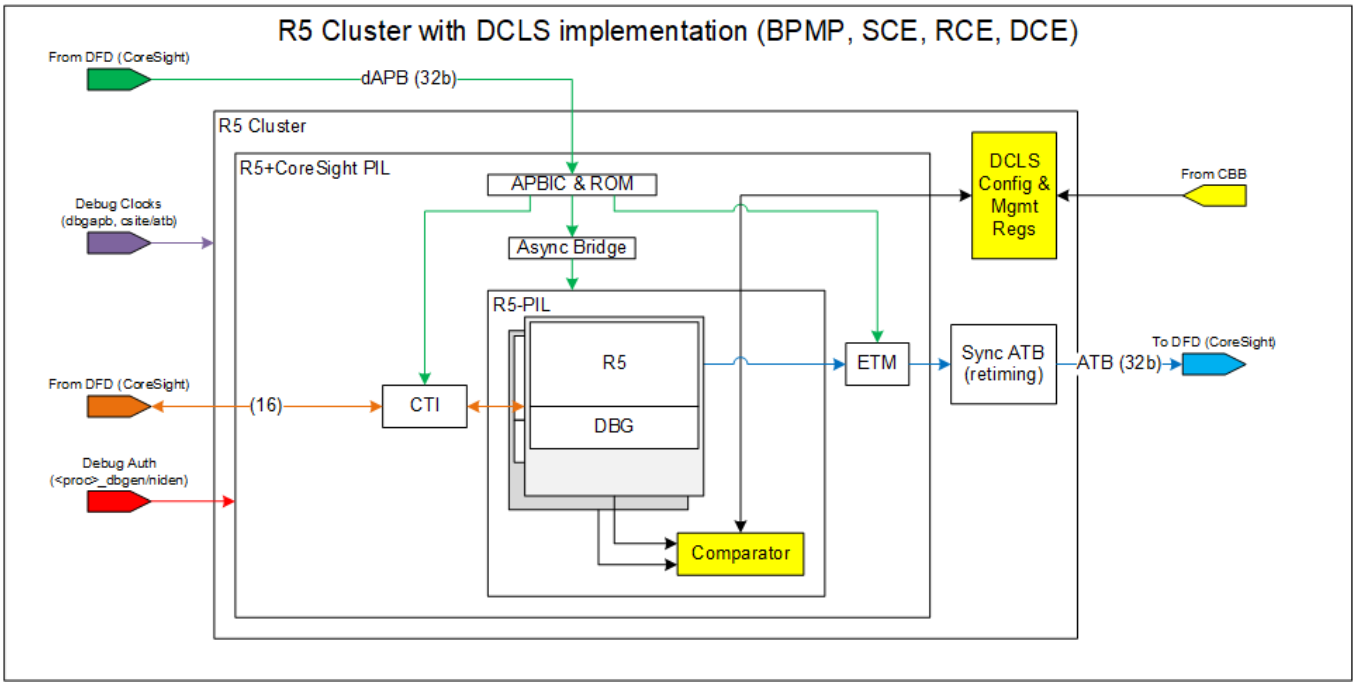


Figure 8.65 Cortex-R5 Cluster without DCLS Implementation

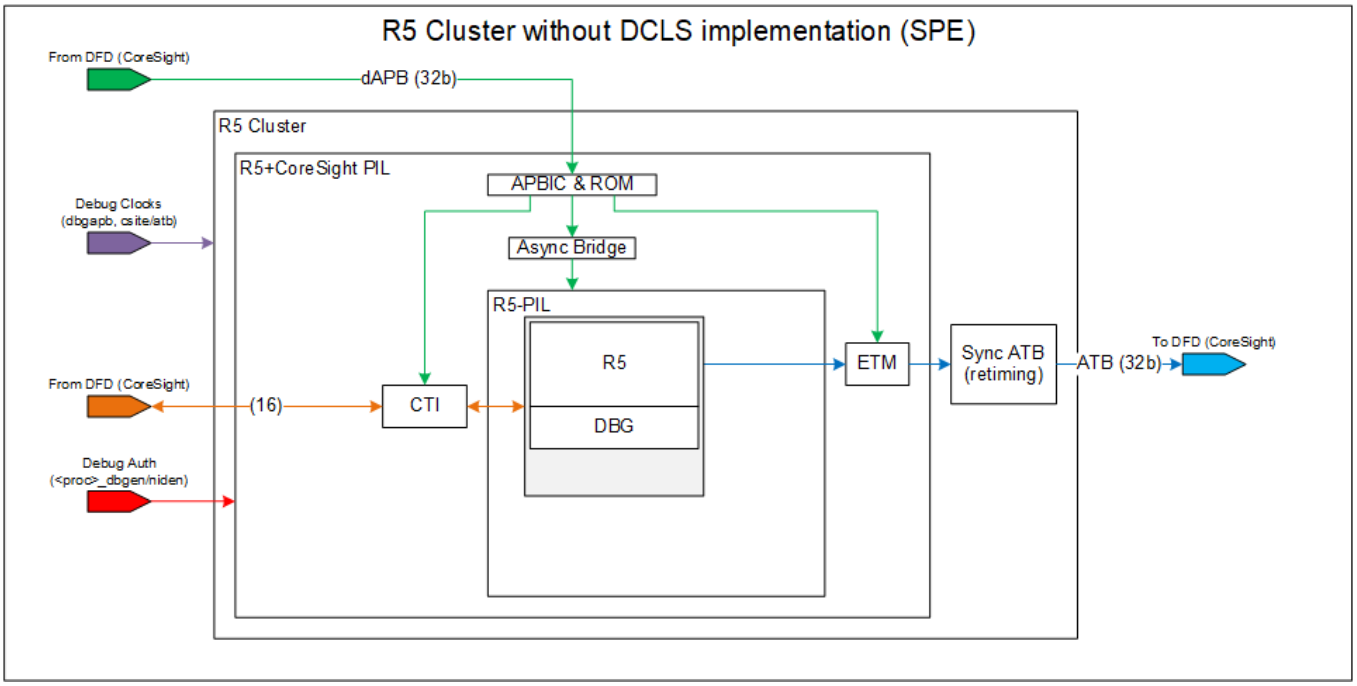


Figure 8.66 Cortex-A9 Cluster without DCLS Implementation

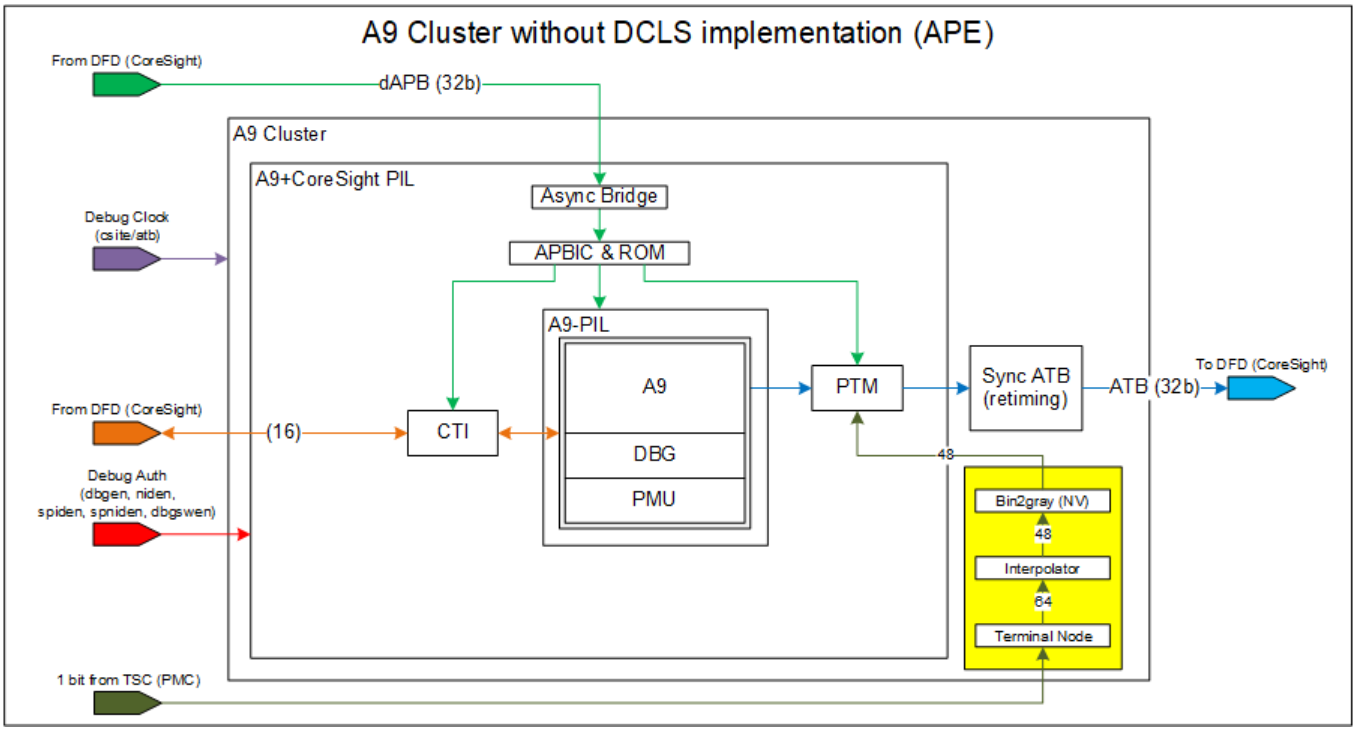
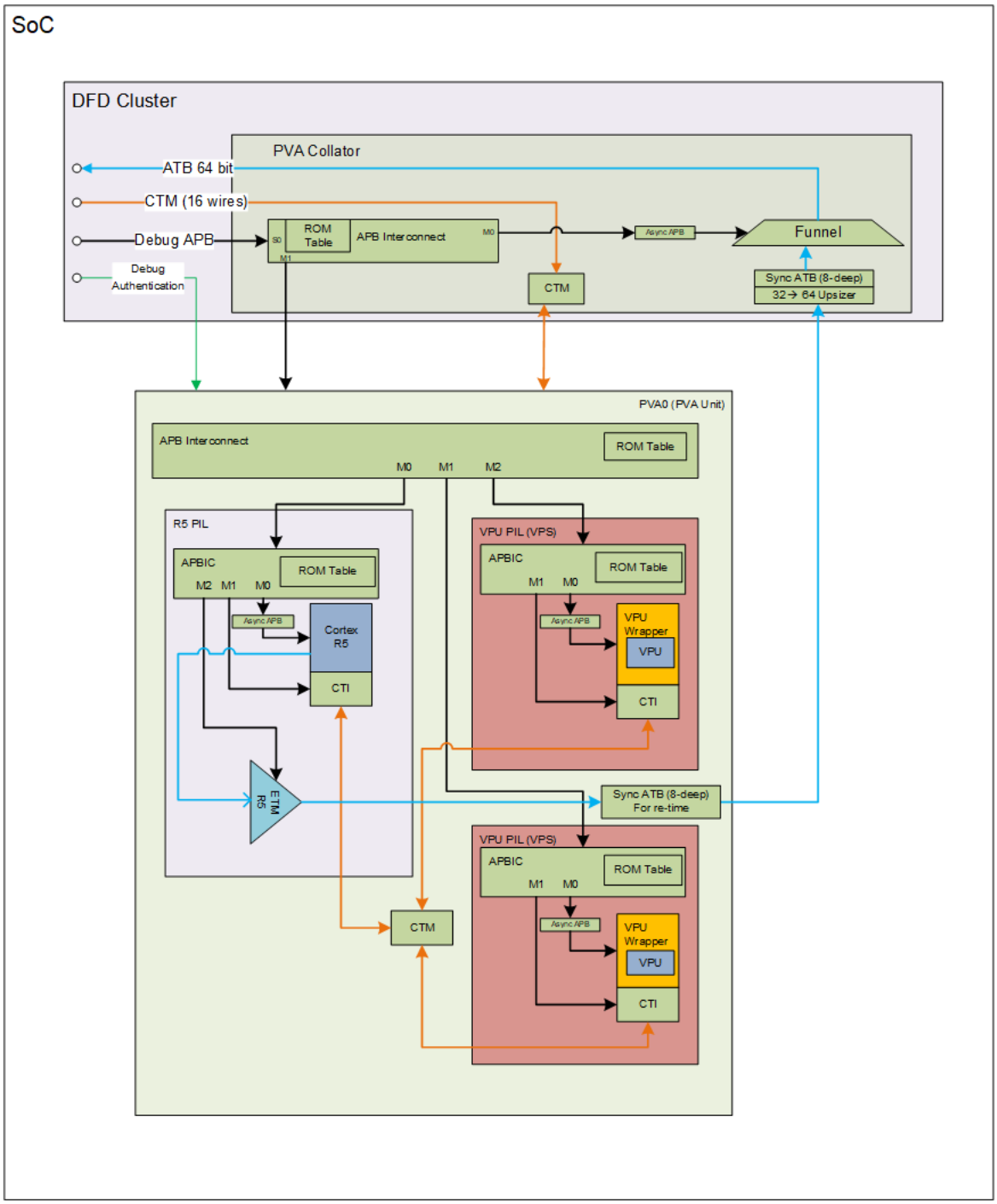


Figure 8.67 PVA Cluster



8.7.2.1.4 CoreSight Configuration Block

The CoreSight Configuration Block (CoreSight Cfg) is an SoC-specific aperture of registers useful for debug purposes and is accessible by the debugger over APBIC. Registers in this aperture are described along with the debug features in the subsequent sections.

8.7.2.1.5 CoreSight IDCODEs and CHIPIDs

JTAG-DP registers are not accessible over SWD, and SW-DP registers are not accessible over the JTAG interface. There are two instances of SWJ-DP component, one configured as SW-DP with JTAG-AP being accessible over the SWD interface, while the other is configured as JTAG-DP and accessible over the JTAG interface.

For details of accessing the CoreSight in the SoC environment, refer to the 'Arm CoreSight SOC-400 Technical Reference Manual, Revision: r3p2.'

The following table summarizes the IDCODEs of the different types of CoreSight instances in the SoC.

Table 8.65 CoreSight DAP Component Identification Values

Component	Register	Value	Comment
JTAG-AP	IDR	0x 3 4760010	
AXI-AP	IDR	0x44770004	
APB-AP	IDR	0x54770002	
JTAG-DP	IDCODE	0x6BA00477	IRLEN8=0
SW-DP	DPIDR	0x6BA02477	

The MMIO R/W register CORESIGHT_CFG_CHIPID_0 is used to check the connection to a particular CPU or Access Port. An MMIO write can be issued from the CPU or Access Port to this register and then the value can be read back. When the value read back matches the value written, the connection is considered successful. The shell performs this sequence and when the value read matches the value written, it prints out the matched value. Otherwise, it signals an error. To avoid false matches, different values are chosen for writing to and reading from different CPUs and Access Ports as summarized in the table below.

Table 8.66 CORESIGHT_CFG_CHIPID_0 R/W Values from Different Initiators using Debugger

CPU/Access Port	CHIPID_VAL	Comments
APB-AP	0xcdcdcdcd	
AXI-AP	0xcacacaca	

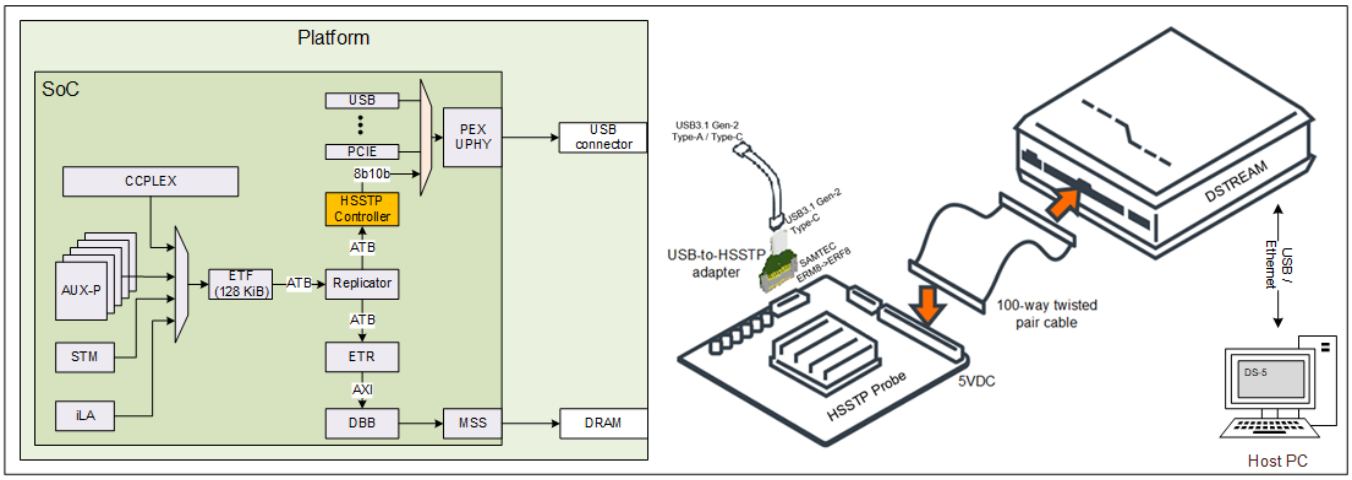
CPU/Access Port	CHIPID_VAL	Comments
APE	0xaaaaaaaa	
BPMP	0xbbbbbbbb	
SPE	0x5e5e5e5e	
SCE	0x5c5c5c5c	
RCE	0x04ce04ce	
DCE	0x0dce0dce	
PVA	-	PVA is not an initiator on CBB and hence PVA-R5 and VPUs cannot write/ read CORESIGHT_CFG_CHIPID_0. Instead, PVA-R5 can write to PVA_CFG_CCQ_STATUS3 a sample value (e.g., 0x095a095a) and read it back. The VPU cannot perform MMIO accesses. Hence, the test could have VPU write sample value (e.g., 0x05930593) to a location in VMEM and read it back.
-	-	
FSI	0x03510000 – cluster-0 core-0 0x03510010 – cluster-1 core-0 0x03510011 – cluster-1 core-1 0x03510012 – cluster-1 core-2 0x03510013 – cluster-1 core-3	R5: cluster-0 R52: cluster-1
CCPLEX Cores	bcbc0000 –cluster 0 cpu 0 bcbc0001 –cluster 0 cpu 1 bcbc0002 –cluster 0 cpu 2 bcbc0003 –cluster 0 cpu 3 bcbc0010 –cluster 1 cpu 0 bcbc0011 –cluster 1 cpu 1 bcbc0012 –cluster 1 cpu 2 bcbc0013 –cluster 1 cpu 3 bcbc0020 –cluster 2 cpu 0 bcbc0021 –cluster 2 cpu 1 bcbc0022 –cluster 2 cpu 2 bcbc0023 –cluster 2 cpu 3	

8.7.2.1.6 HSSTP

HSSTP Overview

HSSTP refers to High Speed Serial Trace Port. High Speed Serial Trace Port Probe (HSSTP Probe) refers to an Arm probe that converts the trace data output from HSSTP (Port) to a parallel format consumed by DSTREAM.

HSSTP provides the means to extract trace from various on-chip sources such as execution/ instruction traces from CCPLEX ETMs, Cortex-R5 ETMs, Cortex-A9 PTM, etc., and instrumentation traces from STM in real time to an off-chip device at high speeds. The maximum trace rate is capped at 10Gbps/lane with support for up to two lanes providing a combined bandwidth of 20Gbps.



Note: DSTREAM-HT and Arm-DS can be used instead of DSTREAM, HSSTP-Probe, and DS-5 mentioned in the diagram above.

The HSSTP Controller receives trace in ATB protocol from an upstream ATB component. It formats the trace to embed the source IDs (also known as, ATIDs) thus generating trace in a parallel format. The formatted trace is converted into a Xilinx Aurora Protocol and sent as pairs of octets (8b10b-encoded symbols) to the supported lanes of UPHY brick on which HSSTP trace is enabled for the particular platform configuration. The UPHY brick used for HSSTP is also referred to as ‘PEX-UPHY’ or ‘HSIO-UPHY’ to differentiate it from other UPHY bricks in the chip. The parallel 8b10b encoded octets are serialized within the UPHY and transmitted over a pair of differential signals, which constitute as one ‘lane’.

Depending on the non-debug/functional use case for the particular port on the platform, a connector is available. For example, in platforms where UPHY Lane-0 is connected to a USB connector, HSSTP is available over the USB connector. Arm provides a USB-to-HSSTP adapter that converts the USB form factor to the SAMTEC-ERM8 form factor. The Arm HSSTP probe and DSTREAM(-ST) connected downstream eventually make the trace available to the Host (PC).

HSSTP Target Use Case

In automotive, for safety reasons, all known issues that adversely affect the system must be analyzed and the cause determined so they can be fixed. If issues are found, e.g., a function takes 500ms more to respond than expected, causing it to miss its deadline, then it is necessary to understand why, especially if it is a critical function like applying brakes or a steering reaction. If the event occurs only under a specific set of circumstances, to reproduce the issue, it is necessary to know all the circumstances that led to it. This can be difficult to determine.

Tracing to DRAM may disrupt memory bandwidth and thus, affect system behavior. This may make it difficult to test system behavior, as the DRAM patterns while streaming to DRAM are different than in mission mode, when tracing is not enabled. For performance issues where such functional logic (e.g., DRAM bandwidth) is already stressed, it is helpful to be able to perform tracing without affecting memory bandwidth or access patterns.

HSSTP provides a real-time high bandwidth (20Gbps) trace path to off-chip targets like DSTREAM-HT in a transparent manner without affecting system behavior, for example, DRAM bandwidth or patterns by streaming trace off-chip rather than to DRAM.

The trace can either be stored in a circular buffer (8GiB for DSTREAM-HT) in the HSSTP-Probe or streamed to the host machine (bandwidth limited by USB interface between host machine and debugger like DSTREAM-HT). When used in circular buffer configuration, software can program the trigger to stop tracing when the event of interest occurs (in the above case, the increased response latency). This allows the circular buffer to store the execution details for last ‘n’-seconds helping developers understand what was happening in the system before the failure occurred.

HSSTP Configurations

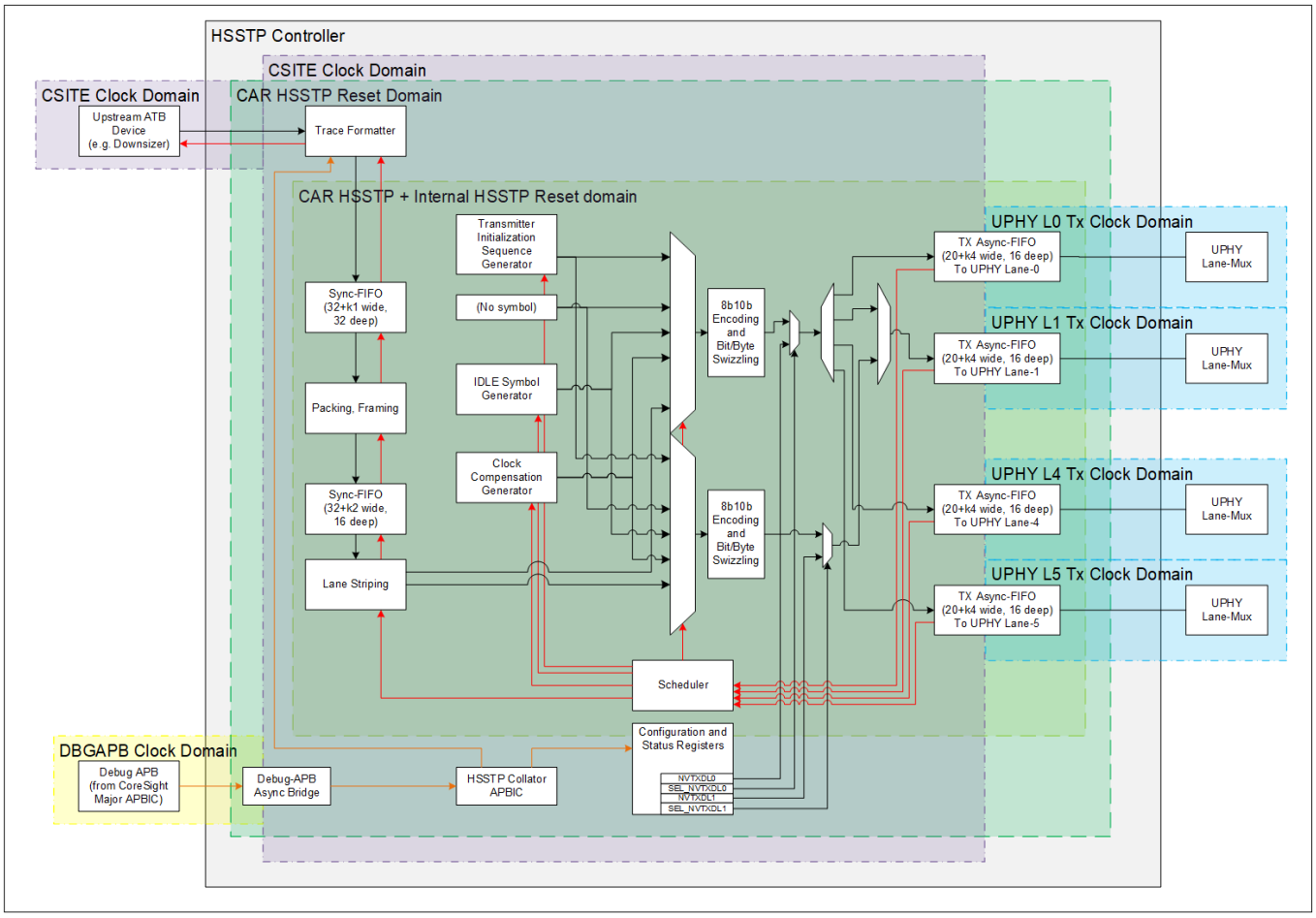
The following HSSTP configurations are supported.

Table 8.67 HSSTP Configurations

Cfg#	Lane Mapping	Maximum Rate Per Lane	Maximum Combined Rate	Comments
A	HSSTP_L0 > UPHY_L0_TX HSSTP_L1 > UPHY_L1_TX	10Gbps	2 x 10Gbps = 20Gbps	
B	HSSTP_L0 > UPHY_L0_TX	10Gbps	10Gbps	
C	HSSTP_L0 > UPHY_L1_TX	10Gbps	10Gbps	
D	HSSTP_L0 > UPHY_L4_TX	10Gbps	10Gbps	
E	HSSTP_L0 > UPHY_L5_TX	10Gbps	10Gbps	

Note: To support 20Gbps, the system has a USB3.1 Gen-2 USB Type-C connector.

Figure 8.68 HSSTP Controller Block Overview



HSSTP Clocks and Resets

PHY Bit Rates

HSSTP debuggers support the following bit rates per lane: 2.5, 3.0, 3.125, 4.25, 5.0, 6.0, 6.25, 8.0, 8.5, 10.0, 10.3125, 12.0, and 12.5Gbps.

Due to clocking limitations in the UPHY brick, bit rates supported are: 2.5 (=10.0/4), 5.0 (=10.0/2) and 10.0Gbps.

Note: Arm's HSSTP probe does not support 8.5Gbps rate.

Overview of Clock Domains

Below are interactions between different clock domains inside the HSSTP controller:

- **DBGAPB_CLK and CSITE_CLK**: This occurs within the Debug-APB Async bridge that converts debug-APB transactions from CoreSight Major APBIC on DBGAPB_CLK to HSSTP Collator APBIC on CSITE_CLK.
- **CSITE_CLK (ungated) and CSITE_CLK (gated)**: Inside TPIU, traceclkin uses CSITE_CLK (gated) while atclk and pclkdbg use CSITE_CLK (ungated). TPIU considers traceclkin as async compared to atclk. Additionally, the TPIU downstream interface (which is on traceclkin, also known as, gated CSITE_CLK) drives input to Sync-FIFO, which is on an ungated version of CSITE_CLK. tracectl output from TPIU that indicates whether output is valid or not (and is used to insert data into the Sync-FIFO), is masked to 1b'0 when traceclkin is gated. This ensures that the Sync-FIFO does not continue to accept the same data multiple times when traceclkin is gated. **Note**: When not mentioned, CSITE_CLK typically refers to the CSITE_CLK without the SLCG.
- **CSITE_CLK and UPHY Tx clocks (UPHY_L#_TX_CLK for HSIO UPHY lanes 0/1/4/5)**: UPHY Tx clock is UPHY's clock for the parallel interface to its controller. This is $1/20^{th}$ the frequency of the serial transmission rate for the UPHY lane. For example, if serial transmission rate is 10Gbps, the corresponding UPHY Tx clock is 500MHz. The parallel clock is synchronous to serial clock to ensure no stray/invalid/empty cycles are observed on the serial differential interface, even when valid data is available on the parallel interface to UPHY on every cycle of the parallel clock.

The block diagram in the previous section provides a high-level illustration of clock domains in the HSSTP controller.

Overview of Reset Domain

The HSSTP controller has a software based L2warm reset, as it is located on the VDD_CORE power rail's Non-Power-Gated domain (also known as, VDD_CORE_NPG power domain).

HSSTP Architecture Specifications also require support for internally generated reset based on STPCR[nRST]. Additionally, due to its small size, HSSTP does not support localized power cycling, and hence STPCR[STP_PWRUP] also acts as a source of internally generated reset. To prevent deadlocks, the path from CoreSight Major to these bits (e.g., Debug-APB Async Bridge, HSSTP Collator APBIC, STPCR[nRST] and STPCR[STP_PWRUP]) must not contain any logic that is reset based on the internal reset generation sources, namely STPCR[nRST] and STPCR[STP_PWRUP].

Note: STPCR[nSW_RST] does not generate an internal reset; instead it is used as a control signal.

The block diagram in the previous section provides a high-level illustration of reset domains in the HSSTP controller.

HSSTP Programmer's Model

Overview

The HSSTP collator provides a single debug-APB interface from external upstream debug controller to the HSSTP controller. HSSTP collator consists of an APBIC (which contains a ROM Table) and two APB targets downstream of the APBIC, TPIU and HSSTP.

TPIU Register Layout

Refer to the CoreSight SOC-400 TRM, *TPIU register summary* section.

TPIU Register Details

Refer to the CoreSight SOC-400 TRM, *TPIU register descriptions* section.

TPIU supports tpctl and has a trace port width of 32b.

HSSTP Register Layout

The register layout below is primarily based on Arm's HSSTP Architecture Specification, section 11 *Appendix D: Example Programmer's Model Layout*. Modifications have been made where necessary.

Additional registers have been defined to cover custom functionality to support configurable options for the formatter unit and to self-contain certain aspects of some features like trigger.

Table 8.68 HSSTP Register Layout

Offset	Register	Type	Value	Description/Comment
0x500	STPSR	RO	Refer to the HSSTP Register Details section	Refer to the HSSTP Register Details section
0x504	STPCR	RW		
0x508	STPLSR	RW		
0x50C	PCSR	RW		
0x510	LLIR	RW		
0x514	LPCAPR	RO		
0x518	LPIDR	RO		
0x51C	LLIR2	RW		
0xD00	NVLPCR	R/W		NV Link and PHY Control Register
0xD04	NVFPSR	RO		NV Formatter to PHY Status Register
0xD08	NVFFCR	R/W		NV Formatter and Flush Control Register
0xD0C	NVFTCR	R/W		NV Formatter Throttling Control Register
0xD10	NVFLCR	R/W		NV Frame Length Control Register
0xD14	NVFLSR	RO		NV Frame Length Status Register
0xD18	NVPTCR	R/W		NV PHY Throttling Control Register
0xD1C	NVCRCCF	RO		NV CRC Coefficient

Offset	Register	Type	Value	Description/Comment
0xD20	NVTXDL0	R/W		NV TX DATA Lane-0
0xD24	NVTXDL1	R/W		NV TX DATA Lane-1
0xF00	ITCTRL	RO	0x00000000	RAZ/WI
0xFA0	CLAIMSET	RW	0x0000000F	0xF = Four claim tags have been implemented. Refer to the CoreSight Architecture Specification v2.0, section B2.5.10 Claim tag registers, for implementation details.
0xFA4	CLAIMCLR	RW	0x00000000	Refer CoreSight Architecture Specification v2.0, section B2.5.10 Claim tag registers, for implementation details.
0xFA8	DEVAFF0	RO	0x00000000	Value is zero.
0xFAC	DEVAFF1	RO	0x00000000	Value is zero.
0xFB0	LAR	RO	0x00000000	LAR does not exist (also known as, RAZ/WI) because LSR=0x0
0xFB4	LSR	RO	0x00000000	[2:0] = 3'b000 No lock mechanism exists.
0xFB8	AUTHSTATUS	RO	0x00000000	NSID, SNID, and SID are zero. NSNID is b00 **Note**: HSSTP does not support NSNID. Hence, the value is 0x0. Upstream trace components decide whether debug security permits them to transmit trace.
0xFBC	DEVARCH	RO	0x47710A50 b' 0100_0111_ 0111_0001_ 0000_1010_ 0101_0000	[31:21] = [ARCHITECT] = value is 0x23B (b'010_0011_1011) [20:20] = [PRESENT] = value is b1 [19:16] = [REVISION] = value equals LPIDR.STP_ARCH [15:0] = [ARCHID] value is 0x0A50
0xFC0	DEVID2	RO	0x0	Value is zero.
0xFC4	DEVID1	RO	0x0	Value is zero.
0xFC8	DEVID	RO	0x0	Value is zero.
0xFCC	DEVTYPE	RO	0x50	[3:0] = [MAJOR type] = Value is 0x0 ('Miscellaneous') [7:4] = [SUB type] = Value is 0x5 (This value falls in 'Reserved' category. Reallocated by Arm to HSSTP.)

Offset	Register	Type	Value	Description/Comment
0xFD0	PIDR4	RO	0x00000003	[7:4] = SIZE = 0x0 (4KB) [3:0] = DES_2 = 0x3 (JEP106 Continuation code of the HSSTP Designer; NVIDIA is in bank 4; hence 0x3)
0xFD4	PIDR5	RO	0x00	Reserved
0xFD8	PIDR6	RO	0x00	Reserved
0xFDC	PIDR7	RO	0x000	Reserved
0xFE0	PIDR0	RO	0x03	[7:0] = PartNumber[7:0] = PART_0 (for SOC_DFD: 0x1=CSITE_CFG, 0x2=SOC_ILA; 0x3=HSSTP)
0xFE4	PIDR1	RO	0xB1	[7:4] = JEP106 ID code [3:0] = 0xB (NVIDIA's JEP106 ID code = 0x6B) [3:0] = PartNumber[11:8] = PART_1 (0x1=SOC_DFD, 0x2=CCPLEX_DFD)
0xFE8	PIDR2	RO	0x1E	[7:4] = Revision = 0x1 (for SOC_DFD: 0x1=HSSTP) [3] = Uses JEP106 ID code = 0x1 [2:0] = JEP106 ID code [6:4] = 0x6
0xFEC	PIDR3	RO	0x00	[7:4] = Manufacturer Revision Number = REVAND = 0x0 [3:0] = Customer Modified = CMOD = 0x0
0xFF0	CIDR0	RO	0x0D	Value is 0x0D
0xFF4	CIDR1	RO	0x90	[7:4] = Component Class = Value is 0x9 (CoreSight component) [3:0] = Preamble-1 = Value is 0x0
0xFF8	CIDR2	RO	0x05	[7:0] = Preamble-2 = Value is 0x05
0xFFC	CIDR3	RO	0xB1	[7:0] = Preamble-3 = Value is 0xB1

Note: For the register names listed above, an appropriate prefix (e.g., "*CORESIGHT_HSSTP_*") is added to the registers in design to avoid namespace conflict and maintain existing nomenclature uniformity.

The registers are not protected by SCR. RO registers ignore writes. Reads for WO registers return '0' unless defined otherwise. Offsets behave as RAZ/WI unless defined otherwise.

HSSTP Register Details

HSSTP NV Custom Registers

NV Link and PHY Control Register, NVLPCR

[0:0] = [LINK_nDALT_VAL] = 1'b1 (default)

[1:1] = [FLIP_BIT_ORDER] = 1'b1 (default)

value (1'b1) : most significant bit of a symbol/octet generated by STP maps to the least significant bit of the same symbol/octet expected by the UPHY. Invert the order of bits in the 8b10b symbol sent to UPHY.

value (1'b0) : order of bits within the 8b10b symbol sent from STP controller to UPHY is not flipped.

[2:2] = [FLIP_BYTE_ORDER] = 1'b0 (default)

Each data lane from the HSSTP controller to UPHY contains two 8b10b symbols.

value (1'b1) : most significant byte in the pair generated by STP for a UPHY lane is mapped to the least significant byte in the pair sent to the UPHY.

value (1'b0) : most significant byte in the pair generated by STP for a UPHY lane is mapped to the most significant byte in the pair sent to the UPHY.

[3:3] = [PRE_ENC_FLIP_BYTE_ORDER] = 1'b0 (default)

value (1'b1): Swap Byte-0 and Byte-1 inputs of the 8b10b encoder. Also swap the character-type (data(D) vs. control(K)) associated with each of those bytes to ensure 8b10b encoder continues to encode the swapped bytes correctly as per their byte-type.

Note: Do not confuse control-character type 'K' with designator for Ordered Set, Comma, /K/. Also, do not confuse Ordered Sets with Control characters. Ordered Sets can contain data (e.g., D10.2, D12.1, etc.) as well as control (K28.5, K28.2, etc.) characters.

value (1'b0): Do not swap Byte-0 and Byte-1 inputs of the 8b10b encoder.

[4:4] = [TRACEDATA_FLIP_BYTE_ORDER] = 1'b0 (default)

value(1'b1): Swap Byte-0 with Byte-1 and swap Byte-2 with Byte-3 for the 32b trace data from TPIU before inserting it into the frame created by "Packing and Framing stage." It does not impact the byte order of trace data as seen by the CRC calculator, which always sees trace data in original byte order. Also swap Byte-0 and Byte-1 of 16b CRC before inserting it into the frame created by "Packing and Framing stage." It does not impact the position in which CRC and ECP are inserted into the frame. It only impacts the byte positions within the CRC halfword location in the frame where the two CRC bytes are placed.

value(1'b0): Do not do any byte-swapping for trace-data and CRC while inserting them into the frames created by "Packing and Framing stage."

[5:5] = [FORCE_SINGLE_LANE_CHANNEL_BONDING] = 1'b0 (default)

value(1'b0): Do not send channel bonding sequence as part of TIS when HSSTP is in single lane config. Send channel bonding sequence as part of TIS when HSSTP is configured over more than one lane (e.g., dual lane, also known as, x2 mode).

value(1’b1): Send channel bonding sequence as part of TIS irrespective of number of lanes over which HSSTP is configured. Although Xilinx Aurora specifications specify that channel bonding should be skipped during TIS for single lane mode and no Idle sequences (other than those that are part of TIS) should be scheduled during TIS, Xilinx VIP does not seem to fail initialization for single lane even when channel bonding is inserted.

[11:6] = reserved (RAZ/WI)

[12:12] = [SEND_PRIMING_SEQ] = 1’b1 (default)

value (1’b0) : Do not send Priming sequence

value (1’b1) : Send Priming sequence

[13:13] = [FORCE_ASSERT_STPCR_NRST] = 1’b0 (default)

AND’ed with output of STPCR[nRST]. Allows explicit control over when the first initialization sequence is sent.

value (1’b0) : Overrides output of nRST to drive effective value of ‘0’

value (1’b1) : Does not influence output of nRST

[14:14] = reserved (RAZ/WI)

[19:15] = [CHANNEL_BUSY_WAIT_INDEX] = 5b’10100 (default)

Used to derive CHANNEL_BUSY_WAIT_COUNT. Refer to the HSSTP Architecture Specification, *STP Status Register, STPSR* section for details.

[20:20] = [TX_DATA_EN_SRC_SEL] = 1’b0 (default)

value (1’b0) : CORESIGHT_CFG_NVUL*_MISC_CTL_1[TX_DATA_EN] drives corresponding TX_DATA_EN signal to HSIO/UPHY.

value (1’b1) : valid_symbol_indicator (corresponding to the HSSTP lane that is expected to be connected to the respective UPHY lane) drives corresponding TX_DATA_EN signal to HSIO/UPHY.

[21:21] = RSVD

[22:22] = [RETRAIN_LINK] = 1’b0 (default)

value (1’b0) : No action.

value (1’b1) : Actions listed under “Actions for link retraining” are performed when this bit is set. After completion of all the actions, this bit self-clears. Thus, it only remains set from the time it is programmed to ‘1’ to the time SCP and padding pair is sent as the last step in the retraining sequence.

Actions for link retraining

Action-1: If an Aurora frame is being transmitted, stall trace data upstream of “Packing and Framing Stage”.

Action-2: If there was an ongoing frame transmission, terminate the frame by sending CRC/ Padding and ECP.

Action-3: Send CC sequence and restart CC counter. (This reduces the probability of CC sequence occurring between the TIS sequence during retraining.)

Action-4: Send TIS sequence (link initialization, channel bonding, channel verification).

Action-5: Unblock traffic from upstream to “Packing and Framing Stage.” This causes SCP and padding pair to be sent, followed by the resumption of data transmission from upstream.

Note: Although link retraining is supported in the HSSTP controller, Arm's HSSTP Probe does not support it. Thus, do not program PERIODIC_RETRAIN_EN=1 or program RETRAIN_LINK=1.

[23:23] = [PERIODIC_RETRAIN_EN] = 1’b0 (default)

value (1’b0): Do not perform periodic retraining

value (1’b1): Perform periodic retraining. Basically, when the retraining counter expires, perform actions listed under “Actions for link retraining.”

[28:24] = [LINK_RETRAIN_INTERVAL] = 4’b01000 (default)

If periodic retraining is enabled, link retraining is scheduled after every ‘N’ number of CSITE_CLK cycles, where,

N = (2 ^ (LINK_RETRAIN_PERIOD)) * (2 ^ 24)

For CSITE_CLK=625MHz, this allows a minimum retraining interval of approximately 26ms and also intervals greater than a few hours with gradually increasing steps. Default value provides interval of approximately 6-7 seconds if periodic retraining is enabled.

NV Formatter to PHY Status Register, NVFPSR

This section is implementation specific. Status from different state machines used in the formatter can be displayed in different fields of this register.

NV Formatter and Flush Control Register, NVFFCR

[0:0] = [NOCRC_CFG] = 1’b0 (default)

value (1’b0) : CRC is added

value (1’b1) : CRC is not added

[1:1] = [reserved] = 1’b0 (default)

Not used.

[2:2] = [SW_TRIG_OUT] = 1’b0 (default)

[3:3] = [STPEN_CLR] = 1’b0 (default)

value (1’b0) : DLY_TRIG1 does not affect STPCR[STP_EN] when STPCR[STP_EN]==1

value (1’b1) : if STPCR[STP_EN]==1, then positive edge on DLY_TRIG1 clears STPCR[STP_EN]=0

[4:4] = [STPEN_SET] = 1’b0 (default)

value (1’b0) : DLY_TRIG1 does not affect STPCR[STP_EN] when STPCR[STP_EN]==0

value (1’b1) : if STPCR[STP_EN]==0, then positive edge on DLY_TRIG1 sets STPCR[STP_EN]=1

Note: The “STPCR[STP_EN]==0/1” condition above is important. It allows both STPEN_CLR and STPEN_SET to be ‘1’ and yet not have any conflicts/race conditions upon positive edge of DLY_TRIG1.

[5:5] = [NRST_CLR] = 1’b0 (default)

value (1’b0) : DLY_TRIG1 does not affect STPCR[nRST] when STPCR[nRST]==1

value (1’b1) : if STPCR[nRST]==1, then positive edge on DLY_TRIG1 clears STPCR[nRST]=0

[6:6] = [NRST_SET] = 1’b0 (default)

value (1’b0) : DLY_TRIG1 does not affect STPCR[nRST] when STPCR[nRST]==0

value (1’b1) : if STPCR[nRST]==0, then positive edge on DLY_TRIG1 sets STPCR[nRST]=1

[12:8] = [DLY_TRIG0_PERIOD] = 5’b00000 (default)

DLY_TRIG0 is delayed by 2^DLY_TRIG0_PERIOD number of CSITE_CLK cycles before driving any downstream signals. This is not a pulse extender. It delays both, assertions and deassertions in a symmetric way.

[17:13] = [DLY_TRIG1_PERIOD] = 5’b00000 (default)

DLY_TRIG1 is delayed by 2^DLY_TRIG1_PERIOD number of CSITE_CLK cycles before driving any downstream signals, similar to how DLY_TRIG0 is delayed using DLY_TRIG0_PERIOD.

Any mention of the use of DLY_TRIG0/DLY_TRIG1 driving logic refers to the signal post this delay logic unless explicitly mentioned otherwise (e.g., by using terms like non-delayed version). The output from CTIs only comes into this delay logic and all subsequent loads (downstream logic) are driven from this delay logic. As the minimum value is 0x0, the minimum delay on these triggers is 2^0=1 cycle. The maximum delay is 2^31=2x10 CSITE_CLK cycles.

[23:18] = [RSVD]

[24] = [SEL_NVTXDL0] = 1'b0 (default)

value(1'b0): Output from "8b10b Encoding and Bit/Byte Swizzling stage" acts as 20b parallel data source for 20b contents transmitted over HSSTP Lane-0.

value(1'b1): NVTXDL0[DPAT] acts as 20b parallel data source for 20b contents transmitted over HSSTP Lane-0.

[25] = [SEL_NVTXDL1] = 1'b0 (default)

value(1'b0): Output from "8b10b Encoding and Bit/Byte Swizzling stage" acts as 20b parallel data source for 20b contents transmitted over HSSTP Lane-1.

value(1'b1): NVTXDL1[DPAT] acts as 20b parallel data source for 20b contents transmitted over HSSTP Lane-1.

[26] = [PERIODIC_INV_DPAT] = 1'b0 (default)

value (1'b1) = After every eight clock cycles, invert NVTXDL#[DPAT]. Both, NVTXDL0[DPAT] and NVTXDL1[DPAT] should invert on same clock cycle. Count the cycles on which NVTXDL#[DPAT] is accepted by downstream async-FIFOs.

value (1'b0) = Do not periodically invert NVTXDL#[DPAT]

For inter-lane (TX1<>TX2) skew measurements, where longer pulse is needed on serial lines to avoid aliasing, use PERIODIC_INV_DPAT=1'b1 and NVTXDL#[DPAT]=0x00000. This causes periodic sequence of (8 x 20 =) **160 0's followed by 160 1's on each serial lane**, which enables finding inter-lane skews (TX1<>TX2) up to 320 bits without aliasing.

NV Formatter Throttling Control Register, NVFTCR

[5:0] = [MIN_FREE_CNT] = 0x0B (default)

[13:8] = [MAX_FREE_CNT] = 0x1D (default)

If available credits (empty slots) in Sync-FIFO (which is immediately downstream of Trace Formatter) go below MIN_FREE_CNT, enable Trace Formatter Throttling and if they go above MAX_FREE_CNT, disable Trace Formatter Throttling.

[16:16] = [SRC_THROTTLING] = 1'b0 (default)

value (1’b0) : Do not use source throttling during Trace Formatter Throttling

value (1’b1) : Use source throttling during Trace Formatter Throttling

When Trace Formatter Throttling is enabled, atreadys (output from TPIU) and atvalids (input to TPIU) is masked to 1’b0. Additionally, trigin and flushin (inputs to TPIU) are also masked to 1’b0.

[17:17] = [CG_THROTTLING] = 1’b1 (default)

value (1’b0) : Do not use clock gating during Trace Formatter Throttling.

value (1’b1) : Use clock gating during Trace Formatter Throttling.

This causes clock branch to traceclkin input of TPIU to be clock gated. It also masks indicator to Sync-FIFO downstream of Trace Formatter to ensure that the Sync-FIFO does not continue capturing the same data multiple times while the trace out port of TPIU is clock-gated (stalled).

Note: CG_THROTTLING is useful while operating TPIU in pattern generator mode when TPIU itself generates data to be transmitted downstream and does not rely on upstream data. In such cases, SRC_THROTTLING is not useful. But while TPIU is operating in normal mode (rather than pattern generator mode), either SRC_THROTTLING or CG_THROTTLING or both can be used.

Also, when CG_THROTTLING is enabled, if traceclkin input to TPIU is gated due to downstream FIFO being nearly full, APB accesses to following TPIU registers will hang until traceclkin is ungated:

TPIU Register Offset	TPIU Register Name	TPIU Register Description	Comments
0x004	CSPSR (or Current_port_size)	Current Port Size Register	Refer to the *TPIU Register Details* section for details.
0x204	CTPMR (or Current_test_pattern_mode)	Current Test Patterns/Modes Register	
0x208	TPRCR	Test Pattern Repeat Counter Register	
0x308	FSCR	Formatter Synchronization Count Register	

[18:18] = [TRACECTL_BASED_FILTER] = 1’b1 (default)

value (1’b1): Enable TRACECTL based filtering.

value (1’b0): Disable TRACECTL based filtering.

When TRACECTL based filtering is enabled, downstream FIFO ignores TPIU output packets when TRACECTL output signal of TPIU is 1’b1. When TRACECTL based filtering is disabled, TRACECTL

output signal of TPIU is not used to determine whether downstream FIFO should ignore TPIU output packets or not.

TRACECTL based filtering is useful when TPIU is used in Normal mode, to avoid generating downstream traffic when TPIU is idle.

[19:19] = [HALFWORD_SYNCPKT_FILTER] = 1'b1 (default)

value (1'b1): Enable filtering of Halfword Synchronization Packets

value (1'b0): Disable filtering of Halfword Synchronization Packets

Filtering of Halfword Synchronization packets is useful when TPIU is used in Continuous mode, to reduce downstream traffic when TPIU is idle.

[20:20] = [FULLFRAME_SYNCPKT_FILTER] = 1'b1 (default)

value (1'b1): Enable filtering of full frame synchronization packets.

value (1'b0): Disable filtering of full frame synchronization packets.

When TPIU does not contain any data and is not in the middle of a formatter frame, it sends full frame synchronization packets. It also sends these upon initialization until it receives data from upstream trace network.

When CG_THROTTLING=0, SRC_THROTTLING by itself is unable to prevent data generated in TPIU from flooding the downstream FIFO. This data is generated at 32b on 625MHz clock and hence can fill the downstream FIFO at a faster rate than it can be drained. This can trigger the throttling mechanism. In this case, TPIU is further deprived of real data from upstream trace network and hence continues to send full frame sync packets which in turn flood the downstream FIFO to cause the throttling mechanism to remain triggered. Thus, a livelock is formed where the only output from TPIU is full sync packets. (While SRC_THROTTLING is also not helpful when TPIU generates data while in pattern generation mode, as that is not typical during normal tracing, that case can be ignored.)

To avoid a livelock, the rate at which full frame sync packets flood the downstream FIFO needs to be reduced. However, filtering all full frame sync packets is not possible because full frame sync packets are also used for frame alignment. Hence, only the density of occurrence of full sync packets needs to be reduced without completely eliminating the occurrence of any entire consecutive sequence of full sync packets.

To achieve this, whenever a full frame sync packet occurs for first time or if it occurs after any non-full frame sync, it is transmitted. After that, the next 'n' (where 'n' = FSP_FILTER_LEN) consecutive occurrences of full frame sync packets are filtered before transmitting the next full frame sync packet. If a non-full frame sync packet occurs, then the filtering of full frame sync packets it stopped and reset. Thus, any occurrence of full-frame sync packet after a non-full frame sync packet is always transmitted.

Thus effectively, any consecutive sequence of full frame sync packets of non-zero length at least generates a corresponding consecutive sequence of full frame sync packets that is non-zero in length and whose length is less-than-or-equal to the original length.

[29:24] = [FSP_FILTER_LEN] = 0x10 (default)

Determines the number of full frame sync packets to filter that occur consecutively after a full frame sync packet occurs and is transmitted, before transmitting the next full frame sync packet.

NV Frame Length Control Register, NVFLCR

[11:0] = [MAX_VALID_FRAME_LENGTH] = 0x100 (default: 1KB)

User Data Frame length, in granularity of 32 bits (4 bytes). Thus, MAX_VALID_FRAME_LENGTH=0x3 implies, after 12 bytes (=(3 * 4 bytes) of data has been sent in the frame, end of frame set (including CRC, depending on no_CRC configuration value) is transmitted. Maximum amount of data between start and end of frame (excluding CRC) is when MAX_VALID_FRAME_LENGTH=0xFFF, when (4095x4)=16380 bytes of data between start and end of frame is achieved.

As the granularity of processing data in packing and framing stage is 4-bytes, a counter granularity of 4-bytes translates to incrementing the count for every cycle of valid formatted data from upstream being transmitted downstream.

If MAX_VALID_FRAME_LENGTH==0, then frame length is 16384 bytes (16KB).

[27:16] = [MAX_EMPTY_CYCLE_COUNT] = 0x100 (default: 256 cycles)

When no data is being sent downstream of TPIU for a considerable amount of time, this ensures any open frame is terminated gracefully. The number of consecutive empty cycles to be encountered is MAX_EMPTY_CYCLE_COUNT. Thus, a value of MAX_EMPTY_CYCLE_COUNT=0x2 implies after encountering two empty cycles, frame is terminated by sending end of frame set.

The cycle is considered empty only if the stage is capable of accepting data but there is no data available in the immediate upstream component to be sent. If the stage is not capable of accepting data, or if there is valid data available from upstream, then the cycle is not to be counted towards empty cycle count.

If MAX_EMPTY_CYCLE_COUNT==0, then counting of empty cycles is disabled.

NV Frame Length Status Register, NVFLSR

[12:0] = [CURRENT_VALID_COUNT_STATUS]

Indicates current counter value being used to check if frame length has reached MAX_VALID_FRAME_LENGTH. The status is at 4 byte granularity.

[27:16] = [CURRENT_EMPTY_COUNT_STATUS]

Indicates current counter value being used to check if frame length has reached MAX_EMPTY_CYCLE_COUNT.

NV PHY Throttling Control Register, NVPTCR

[4:0] = [MIN_FREE_CNT] = 0x03 (default)

[12:8] = [MAX_FREE_CNT] = 0x06 (default)

If available credits (empty slots) in the Async-FIFO go below MIN_FREE_CNT, indicate to the scheduler that FIFO is full, and if it goes above MAX_FREE_CNT, indicate to the scheduler that FIFO is ready to accept data.

NV CRC Coefficient, NVCRCCF

[15:0] = CRC_COEF = 16'h8005 (reset)

Represents the coef value used in CRC calculation.

The HSSTP specification recommends coef=16'h8005, which corresponds to

Polynomial: $G(x) = X^{16} + X^{15} + X^2 + 1$

NV TX DATA Lane-0, NVTXDL0

[19:0] = [DPAT] = 20'hFFC00

This is the 20b parallel data that overrides the output of "8b10b Encoding and Bit/Byte Swizzling stage" for HSSTP Lane-0 when SEL_NVTXDL0=1

NV TX DATA Lane-1, NVTXDL1

[19:0] = [DPAT] = 20'hFFC00

This is the 20b parallel data that overrides the output of "8b10b Encoding and Bit/Byte Swizzling stage" for HSSTP Lane-1 when SEL_NVTXDL1=1

HSSTP Arm Standard Registers

STP Status Register, STPSR

Refer to the HSSTP Architecture Specification, *STP Status Register, STPSR* section for details.

Bit[8]=CHANNEL_BUSY indicates High when STP is still processing data and is not 'always read as zero'.

STP Control Register, STPCR

Refer to the HSSTP Architecture Specification, *STP Control Register, STPCR* section for details.

AUX_UFC_EN

AUX_UFC_EN is RAZ/WI. HSSTP protocol does allow the ability to generate AUX UFCs to be optional and this bit being '0' indicates that UFCs are not supported.

HSSTP has three defined UFCs: Trigger UFC, FIFO overflow UFC, and Auxiliary UFC.

Upstream ETF has the capability to insert triggers into the formatted trace stream and, hence it is acceptable to not explicitly support Trigger UFCs in HSSTP.

The formatter and STP (in conjunction with upstream ATB protocol) executes flow control to prevent FIFO overflow. When the FIFO has no trace-data to transmit, additional IDLE or Clock compensation sequences are transmitted, and FIFO underflow scenarios are avoided.

nSW_RST

When asserted ('0'), the 'Packing and Framing' stage does not accept data from upstream FIFO.

nRST

The region indicated within the "CAR HSSTP + Internal HSSTP Reset Domain" as indicated in the HSSTP controller is reset upon assertion of this nRST bit. Unless indicated otherwise (e.g., STP_EN), none of the HSSTP registers (i.e., registers covered in the HSSTP register layout section) are cleared by this.

After CAR HSSTP reset deassertion, this bit transitions to '1'. However, NVLPCR[FORCE_ASSERT_STPCR_NRST] ensures this does not trigger transmission of initialization sequence. And, it ensures there is an opportunity to program various HSSTP registers after CAR reset deassertion and before the HSSTP initialization sequence is sent.

STP_EN

This bit is cleared upon assertion of nRST (i.e., (STPCR[nRST] & NVLPCR[FORCE_ASSERT_STPCR_NRST] & STPCR[STP_PWRUP]): 1->0). However, nRST is not the reset for this bit. It is possible to reprogram this bit even while nRST=0.

When deasserted ('0'), the 'Packing and Framing' stage does not accept data from upstream FIFO.

Also, if STP_EN:1->0 occurs while there is a frame transmission in progress (i.e., SCP has been sent but corresponding ECP has not been sent) in "Packing and Framing Stage", then clearing STP_EN gracefully terminates the frame by sending CRC/Padding and ECP after the last data accepted from upstream FIFO (before Packing and Framing Stage stops accepting more data) is sent downstream. Later, when STP_EN is programmed to '1' again, and data transmission needs to resume, SCP and Padding is sent by the Packing and Framing stage before starting to send the data. The CRC is also reset/initialized as required during the start of frame, accordingly. If STP_EN:1->0 occurs while there is no frame transmission is progress (i.e., no SCP has been sent

after the last ECP) in the “Packing and Framing Stage”, then clearing STP_EN does not cause an ECP to be inserted as that may cause an Aurora Frame Error.

STP_PWRUP

HSSTP is a small unit to implement independent power gating. Hence, STP_PWRUP has the equivalent effect of nRST.

Thus, the effective internal reset is:

```
(CAR HSSTP reset (active low)) &
(STPCR[nRST] (active low)) &
NVLPCR[FORCE_ASSERT_STPCR_NRST] &
(STPCR[STP_PWRUP] (active high))
```

STP Lanes Select Register, STPLSR

Refer to the HSSTP Architecture Specification, *STP Lanes Select Register, STPLSR* section for details.

[2:0] = [STP_LANESEL]: This field decides whether HSSTP streams trace over one lane (STP_LANESEL=3’b0) or two lanes (STP_LANESEL=3’b1).

As only single or dual lane is supported, STP_LANESEL[2:1] value is ignored. These are tied to b’00.

PHY Clock Synchronization Register, PCSR

Refer the HSSTP Architecture Specification, *PHY Clock Synchronization Register, PCSR* section for details.

Note: [15:0]=[SYNC_COUNT] is a 16b R/W field with reset value 0x2710 (decimal 10,000).

Link Layer Initialization Register, LLIR

Refer to the HSSTP Architecture Specification, *Link Layer Initialization Register, LLIR* section for details.

LINK and PHY Capability Register, LPCAPR

Refer to the HSSTP Architecture Specification, *LINK and PHY Capability Register, LPCAPR* section for details.

Notes:

LPCAPR[21:18] = [LINK_6LANE:LINK_3LANE] = 4’b0000

LPCAPR[17] = [LINK_2LANE] = (HSSTP_TO_UPHY_LANE_MAP == 4’b0011)

LPCAPR[16] = [LINK_1LANE] = ! (HSSTP_TO_UPHY_LANE_MAP == 4'b0000)

LPCAPR[13:10] = [PHY_FREQ]

LPCAPR[9:8] = [PHY_CLK] = 2'b00 // Clock config A

LPCAPR[5:5] = [LINK_nDALT] = LINK_nDALT_VAL

However, when two octets are presented to the UPHY in parallel, UPHY sends the least significant bit of the least significant octet first. To decouple a PHY's default ordering from the nDALT indicator, parallel ordering of bits fed to the UPHY are decoupled from a bit that drives the RO-bit: LINK_nDALT. Additionally, a bit for byte ordering is also provisioned.

LPCAPR[4:4] = [LINK_NOCRC] = 1'b0

LPCAPR[3:1] = [LINK_LANES] = 3'b110

LPCAPR[0:0] = [LINK_4BYTE] = 1'b0

LINK and PHY Identification Register, LPIDR

Refer to the HSSTP Architecture Specification, *LINK and PHY Identification Register, LPIDR* section for details.

[31:29] = [PHY_VER] = 0x1

[28:18] = [PHY_ID] = 11b'0011_110_1011 // (Continuation Code: 4'b0011, ID code: 7'b110_1011)

[17:15] = [LINK_VER] = 0x1

[14:4] = [LINK_ID] = 11b'0011_110_1011

[3:0] = [STP_ARCH] = 0x1

LINK Layer Initialization Register 2, LLIR2

Refer to the HSSTP Architecture Specification, *LINK Layer Initialization Register 2, LLIR2* section for details.

[0:0] = [LEN_CONF] = 'b1

CORESIGHT_CFG Registers Related to HSSTP

Certain HSSTP related registers need to be protected using SCRs. As SCRs are only supported in the CSITE-CFG debug-APB client, these registers are defined in the CSITE-CFG registers space.

Turtle mode does not clamp any of the I/O.

CORESIGHT_CFG_HSSTP_LANE_CFG

CORESIGHT_CFG_HSSTP_LANE_CFG is protected by an independent SCR.

[3:0] = [HSSTP_TO_UPHY_LANE_MAP] = 4'b0000 (default)

4'b0000 = No lanes selected for output

4'b0001 = UPHY Lane-0 mapped to HSSTP Lane-0

4'b0010 = UPHY Lane-1 mapped to HSSTP Lane-0

4'b0011 = UPHY Lane-0 mapped to HSSTP Lane-0; UPHY Lane-1 mapped to HSSTP Lane-1

4'b0100 = UPHY Lane-4 mapped to HSSTP Lane-0

4'b1000 = UPHY Lane-5 mapped to HSSTP Lane-0

All other values are reserved and, when selected, cause undefined behavior.

Note: HSSTP_TO_UPHY_LANE_MAP influences UPHY's lane and PLL controls in HSIO and CAR in addition to LPCAPR[LINK_2LANE] bit. This field does not influence the HSSTP controller behavior as to whether it sends trace over one lane or two lanes. That behavior is governed by STPLSR[STP_LANESEL]. Debugger-host checks the value of LPCAPR and programs STP_LANESEL accordingly. STP_LANESEL is used as an indicator by HSSTP controller to choose single-lane or dual-lane trace mode.

Selecting dual-lane mode (STP_LANESEL=0x1) if it is not supported (LPCAPR[LINK_2LANE]=1b'0), violates the Arm HSSTP Specification and Programmer Guidelines. In this case, the controller splits trace onto two lanes and as the second HSSTP lane would not be mapped to any UPHY lane, the trace data on that lane is lost and, hence the target is not able to reconstruct the trace.

In LOOPBACK mode, CAR assumes HSSTP uses the same lanes for Rx as the ones selected for Tx. For example, if L0+L1 is selected for Tx, then L0+L1 is selected for Rx. If only L1 is selected for Tx, then only L1 is selected for Rx. Hence, no separate lane-mapping is required for Rx in LOOPBACK mode for clocks.

CORESIGHT_CFG_HSSTP_SPEED_CFG

CORESIGHT_CFG_HSSTP_SPEED_CFG is protected by an independent SCR.

[2:0] = [RATE_ID] = 3'b000 (default)

Table 8.69 CORESIGHT_CFG_HSSTP_SPEED_CFG

HSSTP_SPEED_CFG [RATE_ID]	LPCAPR [PHY_FREQ]	Serial Frequency	Parallel Frequency (TX_CLK)	VCO Frequency	TXCLKREF (from UPHY to CAR)	Comments
010	4'b1010	10Gbps	500MHz	10GHz	500MHz	

HSSTP_SPEED_CFG [RATE_ID]	LPCAPR [PHY_FREQ]	Serial Frequency	Parallel Frequency (TX_CLK)	VCO Frequency	TXCLKREF (from UPHY to CAR)	Comments
001	4'b0101	5Gbps	250MHz	10GHz	500MHz	
000	4'b0001	2.5Gbps	125MHz	10GHz	500MHz	

CORESIGHT_CFG_HSSTP_ACTIVATE_CFG

CORESIGHT_CFG_HSSTP_ACTIVATE_CFG is protected by an independent SCR.

CORESIGHT_CFG_HSSTP_ACTIVATE_CFG is on L2warm reset (csite_sysreset_rstn). This ensures that the bits are deasserted post-L2 reset. UPHY PLLs turn off on L2 reset. L2 reset entry may occur abruptly. CAR ASIC relies on the valid indication from sidebands driven by bits in this register as trigger to start its sequence to enable the parallel clock for HSSTP (TX_CLK) to HSSTP and UPHY-LANEs. If CAR attempts to do this when UPHY PLL is OFF, that prevents the correct frequency being provided subsequently for HSSTP parallel interface. Hence, upon L2 reset, as UPHY-PLL is OFF, the signals in this register deassert.

[0:0] = [RATE_ID_VALID] = 1'b0 (default)

CAR provides parallel frequency input to HSSTP and UPHY for HSSTP-to-UPHY interface.

value (1'b0) : indicates to CAR that a valid RATE_ID is not present in CORESIGHT_CFG_HSSTP_SPEED_CFG[RATE_ID] and hence, it does not make any changes to existing frequency settings.

value (1'b1) : indicates to CAR that a valid RATE_ID is present in CORESIGHT_CFG_HSSTP_SPEED_CFG[RATE_ID] and parallel frequency should be updated accordingly.

[1:1] = [UPHY_LANE_MAP_VALID] = 1'b0 (default)

value (1'b0) : Informs CAR/HSIO that contents in CORESIGHT_CFG_HSSTP_LANE_CFG[HSSTP_TO_UPHY_LANE_MAP] are not valid.

value (1'b1) : Informs CAR/HSIO that contents in HSSTP_TO_UPHY_LANE_MAP are valid. HSIO has a per-lane controller selection register. Although bits in HSSTP_TO_UPHY_LANE_MAP are in the same register and hence, can be modified synchronously, as they are separate wires, if CAR desires an atomic indication, ACTIVATE_LANE_SEL can be used for it. For example, to switch to lanes simultaneously, bits of HSSTP_TO_UPHY_LANE_MAP can be individually AND-ed at target with corresponding select signals from HSIO and ACTIVATE_LANE_SEL to indicate readiness to switch the lane to HSSTP.

[2:2] = [UPHY_LANE_CTL_VALID] = 1'b0 (default)

This is an optional signal.

value (1’b0) : indicates to HSIO that contents of CORESIGHT_CFG_NVUL* are invalid

value (1’b1) : indicates to HSIO that contents of CORESIGHT_CFG_NVUL* are valid

CORESIGHT_CFG_HSSTP_STREAM_CTL

CORESIGHT_CFG_HSSTP_STREAM_CTL is protected by an independent SCR.

CORESIGHT_CFG_HSSTP_STREAM_CTL is on L2warm reset (csite_sysreset_rstn).

[0:0] = [ALLOW_IO_TRACE_STREAMING] = 1’b0 (default)

Value(1’b0) = Mask ATREADY output from HSSTP to CSITE and ATVALID output from CSITE to HSSTP to ‘0’. Also mask AFREADY input to HSSTP to ‘1’.

Value(1’b1) = Do not mask ATVALID and AFREADY inputs to HSSTP and ATREADY output from HSSTP to CSITE.

Used to gate traffic to HSSTP until HSSTP is ready to accept traffic post an L2 reset or a SC7-debug exit. Gives opportunity to program registers before traffic starts to stream into HSSTP during these events to prevent any trace, retained in ETF across the reset, from spilling on an unconfigured TPIU. In all other cases (i.e., upon L0/L1 reset or normal/non-debug SC7 exit), HSSTP can be configured before arming the upstream trace components to start streaming trace.

Note: In addition to its reset and MMIO writes, ALLOW_IO_TRACE_STREAMING is also set/cleared upon assertion (positive edge) of DLY_TRIG0 based on AITS_CLR and AITS_SET.

[1:1] = [AITS_CLR] = 1’b0 (default)

value (1’b0) : DLY_TRIG0 does not affect ALLOW_IO_TRACE_STREAMING when ALLOW_IO_TRACE_STREAMING==1

value (1’b1) : if ALLOW_IO_TRACE_STREAMING==1, then positive edge on DLY_TRIG0 clears ALLOW_IO_TRACE_STREAMING=0

[2:2] = [AITS_SET] = 1’b0 (default)

value (1’b0) : DLY_TRIG0 does not affect ALLOW_IO_TRACE_STREAMING when ALLOW_IO_TRACE_STREAMING==0

value (1’b1) : if ALLOW_IO_TRACE_STREAMING==0, then positive edge on DLY_TRIG0 sets ALLOW_IO_TRACE_STREAMING=1

CORESIGHT_CFG_HSSTP_INTERRUPT

CORESIGHT_CFG_HSSTP_INTERRUPT is protected by an independent SCR.

[0:0] = [INT_0] = 1’b0 (default)

The output of this bit is AND-ed with global_dbgen and the AND-ed signal drives trigger input to CTI1, LIC and BPMP-VIC. It is not cleared by CTITRIGINACK from the CTI1. It needs to be cleared via an MMIO write to it.

Debugger uses it to interrupt BPMP. Upon this interrupt, BPMPFW's UPHY driver programs the UPHY's *_CFG regs for switching them to HSSTP mode.

[15:1] = [MSG_0] = 15'b0 (default)

The following table describes the different values for MSG_0[14:0].

"Source" is the entity that posts this message.

Source	MSG_0 [3:0] = MSG_PID [3:0]	MSG_0 [7:4] = MSG_SID [3:0]	MSG_0 [14:8] = MSG_DATA [6:0]	Details/Comments
Debugger (INT_0=1)	0x1	0x1	MSG_DATA[1:0] =Maximum lane speed debugger desires. b'00=2.5GHz b'01=5GHz b'10=10GHz b'11=RSVD MSG_DATA[2:2] =Maximum lanes debugger can support b'0=single-lane b'1=dual-lane	Debugger notifies its desired frequency and number of lanes to BPMPFW.
BPMP (INT_0=0)	0xA	0x1	MSG_DATA[1:0] =Recommended lane speed b'00=2.5GHz b'01=5GHz b'10=10GHz b'11=RSVD MSG_DATA[2:2] =Recommended number of lanes b'0=single-lane b'1=dual-lane	BPMPFW notifies SOC capabilities best suited for debugger. For example: 1. If the debugger requests 5GHz and the SoC supports 10GHz or 2.5GHz, then BPMPFW responds with 2.5GHz. 2. If the debugger requests dual-lane but the SoC supports single-lane, then BPMPFW responds with single-lane. If the debugger requests single-lane but the SoC supports dual-lane, BPMPFW still responds with single-lane. Only if the debugger requests and the SoC supports dual-lane, BPMPFW responds with dual-lane.

Source	MSG_0 [3:0] = MSG_PID [3:0]	MSG_0 [7:4] = MSG_SID [3:0]	MSG_0 [14:8] = MSG_DATA [6:0]	Details/Comments
Debugger (INT_0=1)	0x1	0x2	MSG_DATA[1:0] =Maximum lane speed debugger desires in light of SoC capabilities suggested by BPMPFW. b'00=2.5GHz b'01=5GHz b'10=10GHz b'11=RSVD MSG_DATA[2:2] =Maximum lanes debugger desires in light of SoC capabilities suggested by BPMPFW b'0=single-lane b'1=dual-lane	Debugger notifies BPMPFW to enable CSITE_CLK, deassert CAR reset to HSSTP. Configure HSSTP, UPHY lanes/PLL and clocks.
BPMP (INT_0=0)	0xA	0x2	MSG_DATA[1:0] =Configured lane speed b'00=2.5GHz b'01=5GHz b'10=10GHz b'11=RSVD MSG_DATA[2:2] =Configured number of lanes b'0=single-lane b'1=dual-lane	HSSTP, UPHY lane/PLL and clocks are configured
Debugger (INT_0=1)	0xD	0x1	MSG_DATA[6:0]=RSVD	Assert CAR reset to HSSTP controller.
BPMP (INT_0=0)	0xD	0x1	MSG_DATA[6:0]=RSVD	CAR reset asserted to HSSTP controller
BPMP (INT_0=0)	0xE	0x1	MSG_DATA[6:3]=1	HSSTP_ERROR_IS_DISALBED BPMP-FW replies with this error if the debugger initiates HSSTP handshake when the platform BPMP-FW DTB does not have HSSTP enabled.
(same as above)	(same as above)	(same as above)	MSG_DATA[6:3]=2	HSSTP_ERROR_NO_PLL BPMP-FW replies with this error if the debugger initiates HSSTP handshake when the platform BPMP-FW DTB does not have a valid PLL for HSSTP lane 4/5. This is the #1 issue.

Source	MSG_0 [3:0] = MSG_PID [3:0]	MSG_0 [7:4] = MSG_SID [3:0]	MSG_0 [14:8] = MSG_DATA [6:0]	Details/Comments
(same as above)	(same as above)	(same as above)	MSG_DATA[6:3]=3	HSSTP_ERROR_INVALID_PARAMETER BPMP-FW replies with this error if the debugger initiates HSSTP handshake with invalid parameter, for example, max_speed=3, invalid pid/sid.
(same as above)	(same as above)	(same as above)	MSG_DATA[6:3]=4	HSSTP_ERRPR_FAILED_TO_ENABLE BPMP-FW replies with this error if BPMP-FW is not able to enable HSSTP. Possible reasons are: failed to enable CSITE clock, failed to de-assert HSSTP reset, failed to bring PLL to SEQ_ON state.
(same as above)	(same as above)	(same as above)	MSG_DATA[6:3]=5	HSSTP_ERRPR_ALREADY_ENABLED BPMP-FW replies with this error if the debugger initiates a new handshake after HSSTP has been successfully enabled.
BPMP (INT_0=0)	0xE	0x2	MSG_DATA[6:0]=RSVD	Security error

The following registers help drive the control interface to NV's UPHY brick.

NV UPHY Lane Related Registers

CORESIGHT_CFG_NVUL0_* registers interface with PEX (also known as HSIO) UPHY Lane-0 control interface

CORESIGHT_CFG_NVUL1_* registers interface with PEX (also known as HSIO) UPHY Lane-1 control interface

CORESIGHT_CFG_NVUL2_* registers interface with PEX (also known as HSIO) UPHY Lane-4 control interface

CORESIGHT_CFG_NVUL3_* registers interface with PEX (also known as HSIO) UPHY Lane-5 control interface

The registers corresponding to each lane are defined adjacent/consecutively. All registers are aligned at 4-byte boundary. The set of registers belonging to a lane are aligned at 64B boundary.

The *_OVRD fields are not expected to be driven by individual controllers. They are always driven from corresponding bits in aruphy_lane.h.

CORESIGHT_CFG_NVUL*_AUX_CTL_* registers are protected using a common SCR dedicated for them.

CORESIGHT_CFG_NVUL*_MISC_CTL_* registers are protected using a common SCR dedicated for them.

CORESIGHT_CFG_NVUL*_DYN_CTL_* registers are protected using a common SCR dedicated for them.

CORESIGHT_CFG_NVUL{0,1,2,3}_AUX_CTL_1

Bit	R/W	Reset	Description	Corresponding UPHY Lane Signal	Comments
6	RW	0x0	AUX_TX_RDET_CLK_EN	Enables transmitter clock (AUX_TX_RDET_CLK) to be used during aux receive detect.	
4	RW	0x0	AUX_TX_RDET_EN	AUX_TX_RDET_EN	
2	RW	0x0	AUX_TX_TERM_EN	AUX_TX_TERM_EN	
0	RW	0x1	AUX_TX_IDDQ	AUX_TX_IDDQ	

CORESIGHT_CFG_NVUL{0,1,2,3}_MISC_CTL_1

Bit	R/W	Reset	Description	Corresponding UPHY Signal	Comments
26:24	RO	0x0	TX_RATE_ID	TX_RATE_ID_[2:0]	Driven from HSSTP_SPEED_CFG[RATE_ID]
16	RW	0x0	RESET_	RESET_	
7	RW	0x0	TX_DATA_EN	TX_DATA_EN	Drives UPHY's TX_DATA_EN. Refer to TX_DATA_EN_SRC_SEL for details.
6	RW	0x0	TX_DATA_READY	TX_DATA_READY	
5:4	RW	0x3	TX_SLEEP	TX_SLEEP_[1:0]	
0	RW	0x1	TX_IDDQ	TX_IDDQ	

CORESIGHT_CFG_NVUL{0,1,2,3}_MISC_CTL_2

Bit	R/W	Reset	Description	Corresponding UPHY Signal	Comments
16	RW	0x0	TX_SYNC	TX_SYNC	

CORESIGHT_CFG_NVUL{0,1,2,3}_MISC_CTL_3

Bit	R/W	Reset	Description	Corresponding UPHY Signal	Comments
5:4	RW	0x1	TX_RATE_PDIV	TX_RATE_PDIV_[1:0]	

CORESIGHT_CFG_NVUL{0,1,2,3}_DYN_CTL_1

Bit	R/W	Reset	Description	Corresponding UPHY Signal	Comments
21:16	RW	0x27	TX_DRV_AMP	TX_DRV_AMP_[5:0]	
13:8	RW	0x0	TX_DRV_POST	TX_DRV_POST_[5:0]	
5:0	RW	0x0	TX_DRV_PRE	TX_DRV_PRE_[5:0]	

NV UPHY PLL Related Registers

All NV UPHY PLL related controls are driven from CAR or HSIO (aruphy_pll.h). CAR only expects RATE_ID and Lane-selection related information from HSSTP. BPMPFW/UPHY driver programs the UPHY PLL controls via aruphy_pll.h regs in the UPHY_PLL[0-3] apertures.

Hence, no UPHY PLL controls are provided by HSSTP to UPHY, HSIO, or CAR (except RATE_ID and lane selection information).

HSSTP Pipeline

Trace Formatter Stage

Trace Formatter Overview

Trace Formatter consists of an ATB interface to receive upstream trace data. It embeds Trace Source IDs into the trace data before sending it downstream via its Trace out port. It does this in accordance with CoreSight Architecture Specification v2.0, Trace Formatter chapter.

Trace Formatter utilizes the TPIU component from CoreSight SOC-400 to perform trace formatting. However, TPIU lacks flow-control on its Trace out port. There are two throttling mechanisms to allow the downstream component to prevent TPIU from sending more trace when it is unable to receive it.

TPIU Throttling & Filtering

TPIU Throttling Mechanisms

NVFTCR[CG_THROTTLING] enables clock-gating based throttling. It causes traceclkin input of TPIU to be gated and also masks trcfmt_to_syncfifo_data_valid input to the Sync-FIFO when the downstream Sync FIFO indicates that it is nearly full.

A softer way to throttle TPIU is by blocking upstream traffic (although this throttling method is only helpful when TPIU is not streaming data from its internal pattern generator). Such source throttling is enabled using NVFTCR[SRC_THROTTLING]. atvalid (input to TPIU) and atready (output from TPIU) are masked using src_throttle_en. Additionally, trigin and flushin (inputs to TPIU) are masked to 1'b0.

CG_THROTTLING is more efficient and hence, recommended instead of SRC_THROTTLING.

Filtering TPIU Output Data

Frame synchronization packets (0x7FFF_FFFF) and halfword synchronization packets (0x7FFF) are 16 bit aligned. Halfword synchronization packets typically indicate formatter is idle in Continuous mode. Hence, halfword synchronization packets need not be transmitted but Frame synchronization packets should be transmitted.

Only a 32-bit packet that contains two halfword sync packets, with HALFWORD_SYNCPKT_FILTER enabled, is skipped. However, higher 16 bits of Frame sync packet appear the same as a half sync packet. Hence, when lower 16 bits of current packet are 0x7FFF, it is necessary to know the higher 16 bits in the last packet to determine if the lower 16 bits of the current packet correspond to halfword sync packet or Frame sync packet.

In normal mode, TRACECTL is '0' when valid data packets are being transmitted. It is '1' while transmitting triggers out-of-band (e.g., using TPIU's *FFCR.TrigEvt or FFCR.TrigIn or FFCR.TrigFl*) or when TPIU is idle. When *FFCR.TrigEvt=FFCR.TrigIn=FFCR.TrigFl=0*, that reduces cases when tracectl is '1' to only when TPIU is idle. Upstream trace components like ETF are used if the need arises to insert triggers in the trace stream. These triggers propagate through the tracedata (with traceid: 0x7D) when FFCR.TrigEvt=FFCR.TrigIn=FFCR.TrigFl=0.

When FULLFRAME_SYNCPKT_FILTER is enabled, allowing a full frame sync packet from TPIU to enter the downstream FIFO, the subsequent consecutive occurrence of next FSP_FILTER_LEN full frame sync packets are filtered before allowing the next full frame sync packet from the same string of consecutive full frame sync packets to be allowed to pass into the downstream FIFO.

TPIU output data can be filtered in following scenarios:

- When it is a pair of Halfword Sync packets: Reduces downstream Link-Layer bandwidth by avoiding sending unnecessary packets.
- When traceclkin is gated: Avoids sampling the same tracedata multiple times when TPIU is being throttled using clock gating.

- When tracectl is '1': Causes out-of-band triggers and empty cycles (cycles with no data) to be filtered out and not sent downstream.
- When it is a full frame sync packet following another full frame sync packet. After transmitting a full frame sync packet, up to FSP_FILTER_LEN subsequent consecutive full frame sync packets are filtered before sending the next full frame sync packet in the series of consecutive full frame sync packets. This prevents a livelock scenario when CG_THROTTLING is disabled and only SRC_THROTTLING mechanism is being used.

Passing TPIU Data to Downstream FIFO

Output from TPIU that is not filtered out (based on tracectl, halfword sync packets, etc.) and that is not stale (avoid re-reading the same output multiple times when traceclkin is gated) is fed into the downstream FIFO.

assign trcfmt_to_syncfifo_data_valid = tracectl_data_valid & halfword_syncpkt_data_valid & traceclkin_clk_en & fullframe_syncpkt_data_valid;

TPIU Connectivity and Tie-offs

The table below provides connectivity details regarding TPIU instances within the Trace Formatter stage of HSSTP.

Signal/Port	Direction (relative to TPIU)	Connected Module/ Signal	Comments
tracedata[31:0]	OUT	Downstream Sync-FIFO	
tracectl	OUT	Custom logic used to determine valid data	
tpctl	IN	1'b1	
tpmaxdatasize	IN	5'b11111	
atclk, pclkdbg	IN	csite_clk	
traceclkin	IN	gated_csite_clk	
traceclk	OUT	-	
atclken, pclkendbg	IN	1'b1	
atresetn, presetdbgn, tresetn	IN	hsstp_reset_rstn	
extctlin[7:0]	IN	0x00	
extctlout[7:0]	OUT	-	
trigin, flushin	IN		See the previous section regarding how signals are masked before driving it to TPIU.
triginack, flushinack	OUT		

Signal/Port	Direction (relative to TPIU)	Connected Module/ Signal	Comments
ATB target port	IN (TPIU is target)	Upstream ATB Downsizer (128:32)	See the previous section regarding how atvalid, atready signals are masked before driving it to TPIU.
APB programming port	IN (TPIU is target)	HSSTP Collator APBIC	

Packing and Framing Stage

Start and End of Frames

Post reset or after an end-of-frame set has been transmitted, when the first valid 32b formatted data is received from the upstream FIFO, this stage first transmits a /SCP/ pair (/K28.2/K27.7/) joined by two padding octets /P/P/ (/K28.4/K28.4/). This ensures the first set of data transmitted by this stage is aligned to 32b and maintains that alignment on every cycle.

After transmitting the start-of-frame set (Byte3=/K28.4/, Byte2=/K28.4/, Byte1=/K27.7/, Byte0=/K28.2/), the frame length counter begins to increment for every cycle when a valid 32b formatted data from upstream is transmitted downstream.

Also, an empty frame counter is incremented when the stage does not transmit a valid set downstream. A valid set either constitutes start-of-frame set, end-of-frame set (which may include CRC) or valid formatted data from upstream forwarded downstream. If a valid set is transmitted downstream, then the empty frame counter is reset. Thus, the empty frame counter provides number of consecutive empty sets encountered.

If the frame length counter reaches MAX_VALID_FRAME_LENGTH or empty frame counter reaches MAX_EMPTY_CYCLE_COUNT, end-of-frame set is transmitted and both counters are reset.

End-of-frame set consists of /ECP/ pair (/K29.7/K30.7/). Also, if CRC is enabled, this is preceded by CRC octet-pair. If CRC is not enabled, precede /ECP/ by two padding octets (/P/P/). Thus, if no CRC, end-of-frame set appears as follows: Byte3=/K30.7/, Byte2=/K29.7/, Byte1=/K28.4/, Byte0=/K28.4/. If CRC is present, end-of-frame set appears as follows: Byte3=/K30.7/, Byte2=/K29.7/, Byte1=/CRC_2/, Byte0=/CRC_1/; where /CRC_1/ is lower significant byte of the 16-bit CRC and /CRC_2/ is most significant byte of the 16-bit CRC. This ensures that even during end of frame, the 32-bit alignment is maintained.

After transmitting end-of-frame set, do not send start-of-frame set until the next valid data is available from the upstream FIFO.

CRC Calculation

CRC calculation is described in the Arm HSSTP Architecture Specification, *Error detection* section.

Trace Data Byte Swizzling

When NVLPCR[TRACEDATA_FLIP_BYTE_ORDER]=1'b'1, for the 32b tracedata from TPIU that is being inserted into the frame, byte-0 and byte-1 of the tracedata are swapped and byte-2 and byte-3 of tracedata are swapped, while being inserted into the frame.

Note: The byte-order of the 32'b tracedata that is input to the CRC calculator is not affected by TRACEDATA_FLIP_BYTE_ORDER; only the order in which it is placed into the frame is impacted.

Similarly, when NVLPCR[TRACEDATA_FLIP_BYTE_ORDER]=1'b'1, and if CRC is being inserted into the frame, Byte-0 and Byte-1 or CRC are swapped while being inserted into the frame. Again, calculation of CRC is not affected by TRACEDATA_FLIP_BYTE_ORDER. Only the way (byte-order) in which the final 16b CRC is inserted into the frame is impacted and too, impact is limited to the halfword location assigned to CRC in that frame. That is, position of CRC and ECP is not swapped; only bytes within the CRC are swapped while being inserted into the frame.

Notes:

- Tracedata seen by CRC calculator is not impacted by TRACEDATA_FLIP_BYTE_ORDER.
- Byte ordering for bytes within SCP or ECP is not affected by TRACEDATA_FLIP_BYTE_ORDER.
- Halfword ordering between SCP and the padding octet-pair is not affected.
- Halfword ordering between ECP and the padding octet-pair is not affected.
- Halfword ordering between ECP and the CRC is not affected.

Lane Stripping

This stage tries to distribute the 32-bit wide trace stream received from the previous stage (buffered via intermediate FIFO) over configured HSSTP lanes. There are two configurations in terms of number of lanes used for HSSTP simultaneously: one-lane and two-lanes. Each lane accepts two bytes in parallel from the Lane Stripping stage. Hence, for configurations that support one lane, lower half-word (Byte-1 and Byte-0) are transmitted on first clock cycle and upper half-word (Byte-3 and Byte-2) are transmitted on the subsequent clock cycle before retrieving the next word from upstream FIFO. For configurations that support two lanes, lower half-word are transmitted on HSSTP Lane-0 and upper halfword are transmitted on HSSTP Lane-1. Thus, a new 32-bit data can be processed and stripped over two lanes every cycle as long as there is no back-pressure from downstream/scheduler.

The sch2strp_hold signal from the scheduler to Lane Stripping stage indicates whether the stage holds its current output for another cycle if the downstream pipeline is unable to accept its current output.

Similarly, the data valid signal indicator, strp2sch_data_valid, which is part of the downstream output interface of Lane Stripping stage indicates to scheduler whether current data is valid or not. This signal allows the scheduler to switch the mux to Idle Symbol Generator when there is no upstream data to be streamed.

The strp2isg_send_idle signal is asserted if (sch2strp_hold==0 && strp2sch_data_valid==1 && ((lane0==/P/P/)||(lane1==/P/P/)). Any lane that contains a pair of padding-octets (/P/P/) replaces its contents for that cycle with a pair of idle-octets (/I/I/), that would otherwise have been sent on that

cycle, had Idle Symbol Generator had been scheduled. Only the lane having the pair of padding-octets replaces its contents with a pair of idle-octets. In multi-lane modes, lanes that do not have a pair of padding-octets retain their contents and do not replace them with a pair of idle-octets.

Note: While interpreting whether a byte contains a padding character or not, also consider whether it is a data or control character. Do not replace /D28.4/D28.4/ with /I/I/. Only replace /K28.4/K28.4/ with /I/I/.

For example:

1: In Single lane case, if {L0B1,L0B0}=/P/P/, then replace as {L0B1,L0B0}=/I/I/

2. In Dual lane case, if {L0B1,L0B0} = /P/P/, then replace as {L0B1,L0B0}=/I/I/. If {L1B1,L1B0} is not a pair of padding-octets, do not replace {L1B1,L1B0}.

3. In Dual lane case, if {L1B1,L1B0} = /P/P/, then replace as {L1B1,L1B0}=/I/I/. If {L0B1,L0B0} is not a pair of padding-octets, do not replace {L0B1,L0B0}.

The reason to swap a pair of padding-octets with a pair of idle-octets at the output of Lane Stripper stage, is because in the Packing and Framing stage, it is not clear when a word being created by that stage gets scheduled to be sent, as input to the 8b10b encoder and thus, queued for transmission. Hence, which control character to use is unclear in the Packing and Framing stage. To avoid breaking the K-R-A requirement on idle symbols, the decision of which idle symbols to insert is made at the output of the Lane Stripping stage. The pair of padding-octets inserted at the Packing and Framing stage acts as an unambiguous placeholder; a pair of padding-octets is not expected under any other circumstance.

Sequence Generator Blocks

Transmitter Initialization Sequence Generator

This block generates Transmitter Interface's Lane Initialization, Channel Bonding and Channel Verification Procedure. Given that HSSTP is a multi-lane Serial Simplex Implementation of the Aurora Protocol, receiver related sequences do not apply.

Transmitter Initialization Sequence (TIS) refers to performing Transmit Interface Lane Initialization (TILI), followed by Transmit Interface Channel Bonding Procedure (TICB) and finally, Transmit Interface Channel Verification Procedure (TICV).

TIS generation is triggered via side-band (sch2tis_en) from the Scheduler Block. Upon reset or after completion of TIS, Transmitter IBV Sequence Generator remains in idle state with its output indicating "(no_symbol)" to be transmitted. If sch2tis_en is deasserted at any time before TIS is completed, it causes TIS to pause while sch2tis_en is deasserted and resume upon assertion. While sending its last symbol, TSI generator block asserts tis2sch_done. This indicates to the scheduler that TIS is completed. In response, scheduler must deassert tis2sch_en and start sending other sequence/symbols as required (e.g., IDLE, Clock Compensation, or User PDUs). The only condition

under which TIS is paused/resumed in between is when a clock synchronization sequence needs to be inserted or when downstream FIFO is full.

If sch2tis_en is asserted (after deassertion due to completion of TIS), TIS generation is triggered again.

As HSSTP is based on the Serial Simplex version of the Aurora Protocol, sections *B.5.1 Transmit Interface Lane initialization*, *B.5.3 Transmit Interface Channel Bonding Procedure* and *B.5.5 Transmit Interface Channel Verification Procedure* describe behavior for Transmitted Initialization Sequence Generator. As there is no feedback path from the HSSTP debugger to the HSSTP controller, the overall lane initialization, channel bonding and channel verification procedure is nothing more than transmitting certain control characters for a configurable duration based on Link Layer Initialization Register (LLIR) and LINK Layer Initialization Register 2 (LLIR2) described in the HSSTP Architecture Specification, *Programmer's Model* section.

The following is a summary:

1. Lane Initialization: Transmit symbol-set /SP/ for #LaneBytes÷4 times on each lane, where:

#LaneBytes = ALIGNMENT_PATTERN_MULTIPLIER*(VERIFY_LEN + 1)

where ALIGNMENT_PATTERN_MULTIPLIER is derived from (not equal to) LLIR2.ALIGN_MUL. See the corresponding field description in the Arm HSSTP Architecture Specification (section 6.9 *LINK Layer Initialization Register 2, LLIR2*) for details.

1. Channel Bonding: If the number of lanes is more than one (also see NVLPCR[FORCE_SINGLE_LANE_CHANNEL_BONDING] for exception), transmit symbol /I/ for following number of times on each lane:

#BondingBytes = BONDING_PATTERN_MULTIPLIER*(BOND_LEN + 1)

where BONDING_PATTERN_MULTIPLIER is derived from (not equal to) LLIR2.BOND_MUL. See the corresponding field description in the Arm HSSTP Architecture Specification (section 6.9 *LINK Layer Initialization Register 2, LLIR2*) for details.

1. Channel Verification: Transmit the set of "60 /I/ symbols followed by /V/ (where one /V/ symbol consists of four characters: /K28.5/D8.7 /D8.7/D8.7/)" for floor(#VerifBytes÷64) times on each lane, where:

#VerifBytes = VERIFY_PATTERN_MULTIPLIER*(VERIFY_LEN + 1)

where VERIFY_PATTERN_MULTIPLIER is derived from (not equal to) LLIR2.VERIFY_MUL. See the corresponding field description in the Arm HSSTP Architecture Specification (section 6.9 *LINK Layer Initialization Register 2, LLIR2*) for details.

Note: For LLIR2.VERIFY_MUL and LLIR.VERIFY_LEN, #VerifBytes is a multiple of 64 and #VerifBytes is at least 256.

Thus, VERIFY_LEN=63 and VERIFY_MUL=0 (i.e., VERIFY_PATTERN_MULTIPLIER=4) gives #VerifBytes=4*(63+1)=256 bytes, which is four sets of “60 /I/ symbols followed by /V/”.

Thus, #VerifBytes=256 implies 256/64=4 sets of “60 /I/ symbols followed by /V/” are transmitted on each of the lanes.

If VERIFY_LEN=127 and VERIFY_MUL=0 (i.e., VERIFY_PATTERN_MULTIPLIER=4) gives #VerifBytes=4*(127+1)=512 bytes, which is eight sets of “60 /I/ symbols followed by /V/”.

Thus, #VerifBytes=512 implies 512/64=8 sets of “60 /I/ symbols followed by /V/” are transmitted on each of the lanes.

While transmitting /I/, the sequence requirements between /K/, /R/ and /A/ must be met. As the logic to meet these requirements is already present in Idle Symbol Generator block, rather than replicating the logic again in TIS Generator, TIS Generator sends a side-band to scheduler requesting for the Idle Symbol Generator to be scheduled for as long as TIS Generator needs to send /I/. Once TIS Generator deasserts this side-band, scheduler again configures mux to select TIS Generator output to be sent over the mux. While scheduler selects Idle Symbol Generator upon explicit request from TIS Generator, the responsibility to count the idle symbols lies with TIS Generator. To know whether a pair of Idle symbols was accepted into the downstream FIFO, TIS Generator block must monitor input to Idle Symbol Generator Block to determine whether an Idle Symbol Pair was pushed into the downstream FIFO on the current cycle.

(No symbol) Generator

This block drives the signal pattern that must be ignored by the downstream FIFO. Thus, selecting this mux input prevents new packets from being inserted into the downstream FIFO. The output of this block can be as simple as driving a ‘0’ on the signal/bit-position that indicates whether the current input to FIFO is valid. Thus, when the mux selects this block, new entries do not get added to the FIFO allowing the FIFO to drain.

This is typically used to prevent overflows when the downstream FIFO is unable to accept more entries.

Clock Compensation Generator

This block transmits the Clock Compensation Sequence, which consists of six repetitions of the symbol pair: /CC/ = /CC/$_1$ /CC/$_2$ = /K23.7/K23.7/. Given that each lane accepts two symbols as parallel input, the same pair is transmitted six times.

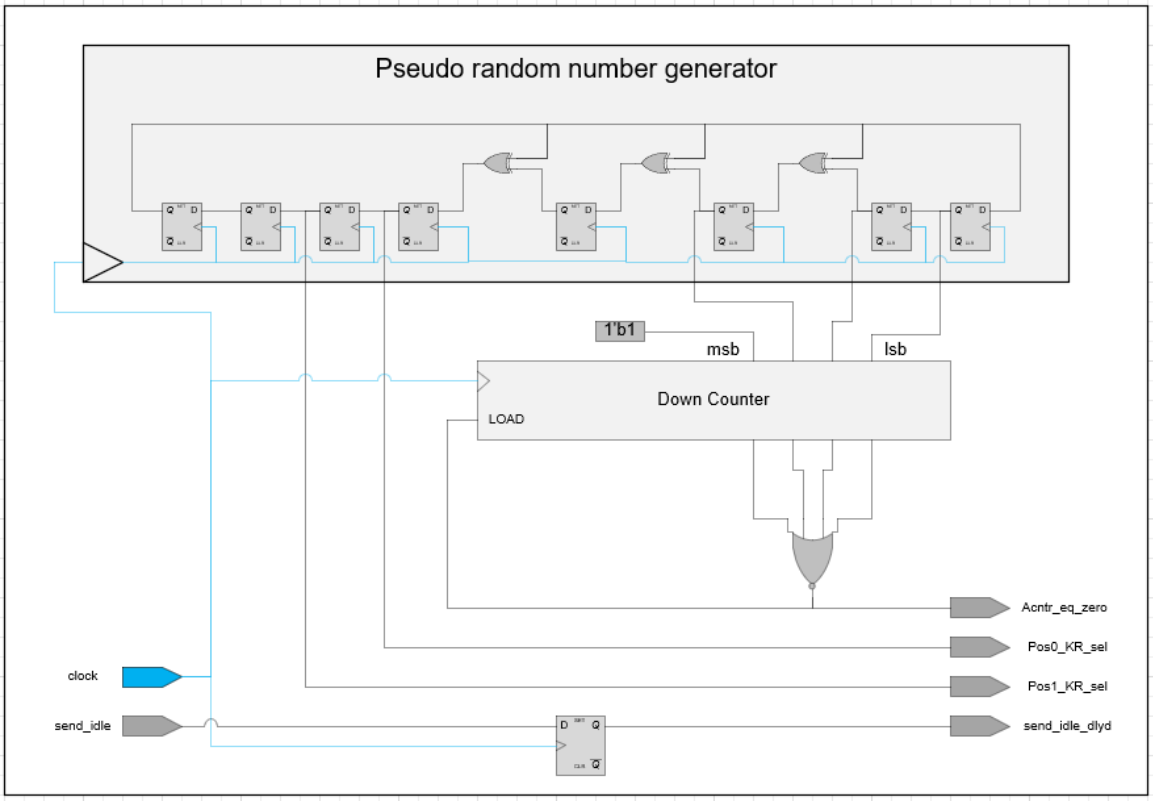
Refer to the Aurora 8b10b protocol specification, *Clock Compensation* section for details on the symbol and encoding for clock compensation sequences. However, it is the scheduler’s responsibility to ensure that six pairs of /CC/$_1$/CC/$_2$ are transmitted. If the downstream FIFO is full, Scheduler may switch to “(No symbol)” and then resume the clock compensation sequence transmission from where it left off. Also, it is the scheduler’s responsibility to ensure periodic transmission of clock compensation sequences compliant to the specification and configuration requirements.

Idle Symbol Generator

Refer to the Aurora 8b10b protocol specification, *Idle Sequence* section for details on how to determine whether to output /A/, /K/ or /R/ for a particular cycle. The scheduler decides if the idle symbol is transmitted, which then affects the decision as to which idle symbol to transmit next. A side-band, sch2isg_send_idle, is asserted by the scheduler to indicate to the Idle Symbol Generator as to whether an idle symbol is being selected for transmission. Additionally, strp2isg_send_idle is asserted when the lane-stripping stage is selected by the scheduler and its output contains a pair of padding octets (/P/P/) on any of its lanes. The OR'ed version of sch2isg_send_idle and strp2isg_send_idle is depicted as the input "send_idle." The "clock" input refers to the CSITE_CLK.

The circuit below illustrates a sample implementation. Two idle symbols need to be generated per cycle, as a symbol-pair is transmitted in parallel. If multiple lanes are transmitting idle symbols at a time, then the symbols transmitted on different lanes at a time are always the same. Hence, the decision of which symbol-pair to transmit occurs on every clock cycle. Different lanes, if transmitting idle simultaneously, use the same pair.

The down-counter does not progress if the Async-FIFO has back-pressured. This is to ensure that any idle symbol that is scheduled to be sent, does not get skipped because the downstream FIFO has backpressured and hence, unable to accept the idle symbols on that cycle.



The following sequencing meets the requirements described in the Aurora 8b10b Protocol specification, *Idle Sequence* section.

send_A_on_pos0 = send_idle & Acntr_eq_zero & send_idle_dlyd

send_R_on_pos0 = send_idle & (!Acntr_eq_zero) & send_idle_dlyd & Pos0_KR_sel

send_K_on_pos0 = send_idle & (!send_A_on_pos0) & (!send_R_on_pos0)

send_R_on_pos1 = send_idle & send_idle_dlyd & Pos1_KR_sel

send_K_on_pos1 = send_idle & (!send_R_on_pos1)

8b10b Encoding and Bit/Byte Swizzling

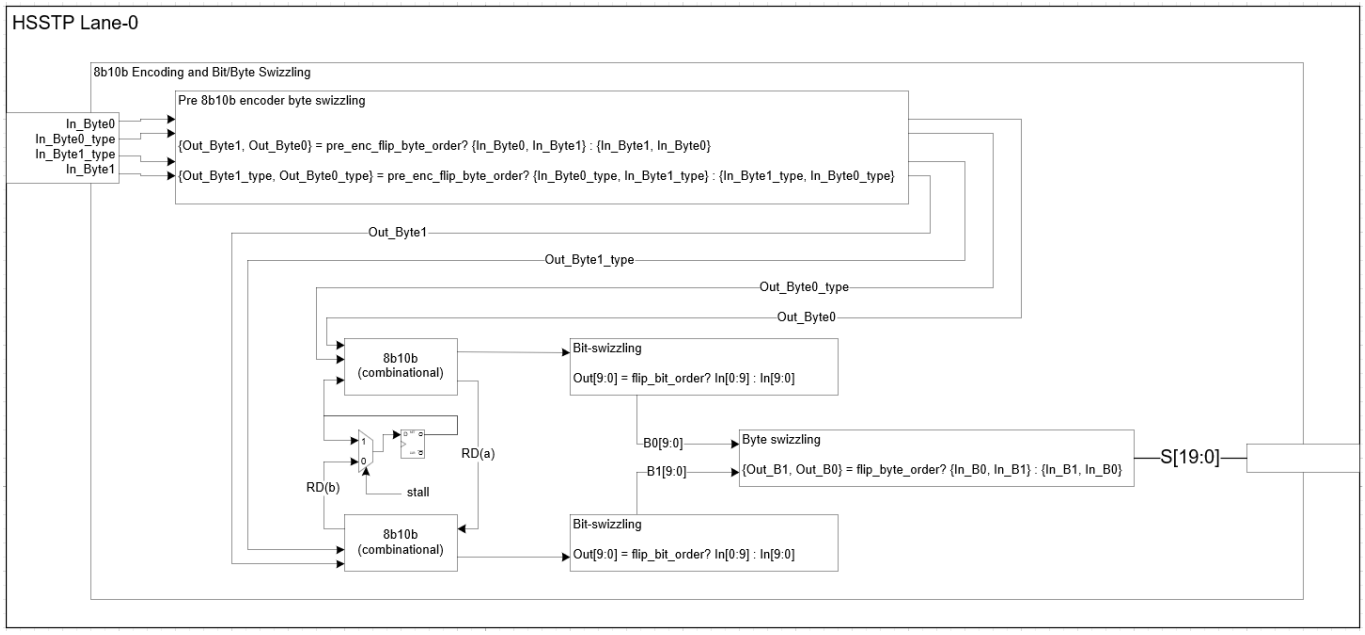
Refer to the Aurora 8b10b Protocol specification (section 5.4 *8B10B Transmission Code* and Appendix A.1 *8B/10B Encoding*) for converting data and control characters to 10b symbols. Refer to the HSSTP specification (section 5.2 *Trace data ordering*) for byte/bit ordering during serialization of data.

During serialization, lower characters/bytes/symbols should be transmitted before higher characters/bytes/symbols. And within a character/byte/symbol, if the encoding is for a control character, then the most significant bit should be transmitted first. For a data character, if LINK_nDALT is ‘1’ (preferred), the most significant bit should be transmitted first, else if LINK_nDALT is ‘0’, the least significant bit should be transmitted first.

In the SoC, LINK_nDALT=’1’, thus both Control and Data characters are sent most significant bit first.

Pre 8b10b encoder allows the higher-byte to be sent first followed by the lower-byte by swizzling their positions prior to 8b10b encoder. As post encoding, B0 must be sent first by the downstream serializer. If the higher byte needs to be sent first, swizzling must occur prior to the encoding stage, which is facilitated by NVLPCR[PRE_ENC_FLIP_BYTE_ORDER].

Figure 8.69 8b10b Encoding and Bit/Byte Swizzling Overview



The goal of post 8b10 encoder bit swizzling (flip_bit_order) and byte swizzling (flip_byte_order) is to make the controller generic to protect it from any changes to PHY – it is not to offer options to send the higher byte first or least significant bit within a byte first. Thus, if PHY changes cause serialization to occur differently than anticipated, using the byte/bit swizzling capability for the 20b/lane parallel input being fed to the PHY, the output of the PHY can be made to match the HSSTP requirements. The bit swizzling applies to bits within the 10b symbol and byte swizzling applies to the two 10b symbols.

The disparity resulting from current input to the Async FIFO passes through a delay flop and is used while generating the 8b10b encoding for the next pair of symbols per lane. Output ‘0’ from the delay block indicates negative running disparity (RD-), while output ‘1’ indicates positive running disparity (RD+). The reset value for the delay flop is ‘0’ (RD-). This logic is independent for each lane. (If there are timing issues due to a long combinational path from any of the generators until the Async FIFO, a sync FIFO can be added between the muxes and 8b10b encoding blocks.)

HSSTP Programming Guidelines

Enabling HSSTP Overview

The following steps provide a high-level sequence to enable HSSTP over UPHY.

Part-0a: Enable hot-plug capability

CORESIGHT_CFG_CBDCR_0 is programmed by MB2 based on MB2-BCT.

CORESIGHT_CFG_CBDCR_0[SWD_PORT_SEL] should be programmed 0/1 depending on which USB2 port (USB2 Port-0 versus USB2 Port-1) is mapped to the USB Type-C connector over which SWD needs to be mapped.

CORESIGHT_CFG_CBDCR_0[SWDDFD_NVDBGSEL_EN] needs to be '1' for hot-plug capability to work.

CORESIGHT_CFG_CBDCR_0[SWD_JTAG_CFG] should always be programmed '0'.

Enable/unmask HSSTP Interrupt source in BPMP-VIC so that when CORESIGHT_CFG_HSSTP_INTERRUPT[INT_0]=1 generates an interrupt to BPMP-VIC, BPMPFW sees the interrupt.

These values should be restored post-SC7 exit as well.

Part-0b: Disable PLLE spread, enable CSITE-CLK and deassert HSSTP CAR reset

```
IF (DT.BPMPFW_SSC_EN==DISABLE)
{
   BPMPFW disables PLLE SSC; // Disable PLLE SSC for HSSTP to work
}
else
{
BPMPFW enables PLLE SSC
}
```

Note: DT.BPMPFW_SSC_EN=DISABLE refers to following DT configuration:

```
clocks {
   clock@plle {
   clk-id = <TEGRA234_CLK_PLLE>;
   disable-spread = <1>;
   };
};


IF ((!(FUSE_PRODUCTION_MODE_0||FUSE_PRODUCTION_MODE_SHADOW_0)) ||
   ((PMC_MISC_DEBUG_AUTHENTICATION_0[JTAG_ENABLE] &&
   PMC_MISC_DEBUG_AUTHENTICATION_SHADOW_0[JTAG_ENABLE]) &&
   (!(FUSE_ARM_JTAG_DIS_0||FUSE_ARM_JTAG_DIS_SHADOW_0))))
{
```

```
   CLK_RST_CONTROLLER_CLK_OUT_ENB_CSITE_SET_0[SET_CLK_ENB_CSITE]= ENABLE // This is the reset
default; Do not bump CSITE-CLK to 625MHz at this point. Do that only after a request from
debugger or ETR-driver. This step is just to ensure CSITE-CLK is running and not OFF.
   BPMPFW should program CLK_RST_CONTROLLER_RST_DEV_HSSTP_0=0
}
```

Above values from Part-0{a,b} should be restored post-SC7 exit too.

Background: How HT-DAM hot-plugging works

The settings above do not enable SWD over USB2 port. The port is still usable as USB. These settings cause the USB2 port to switch to SWD controller when NVDBG_SEL pin is asserted. The HSSTP Debug Accessory Mode (HT-DAM) adapter that plugs into the USB Type-C port causes CC1=CC2=LOW. Logic on platform drives NVDBG_SEL=HIGH when it sees CC1=CC2=LOW. If CORESIGHT_CFG_CBDCR_0[SWDDFD_NVDBGSEL_EN]=0, NVDBG_SEL is ignored. However, if SWDDFD_NVDBGSEL_EN=1, then if NVDBG_SEL goes HIGH, it causes the USB2 pins corresponding to the port selected by CORESIGHT_CFG_CBDCR_0[SWD_PORT_SEL] to switch from USB2 controller to SWD controller. Now via SWD-over-USB2 pins, debugger can program CORESIGHT_CFG_HSSTP_INTERRUPT to interact with BPMPFW. BPMPFW based on these interactions configures and switches the UPHY lanes (USB3 pins) from USB3 (XUSB) controller to HSSTP controller. These interactions between debugger and BPMPFW and subsequent BPMPFW programming requirements are described in the subsequent Part-1 and Part-2 subsections.

Part-1: Debugger informs SoC about intention to connect HSSTP

Part-1a: Debugger notifies intent to connect

1. Debugger connects and programs CORESIGHT_CFG_HSSTP_INTERRUPT[MSG_0, INT_0] to send message with MSG_PID=0x1, MSG_SID=0x1 , INT_0=1 and applicable MSG_DATA values.
2. BPMPFW based on SoC/platform knowledge (e.g., based on BCT/DT) responds to the message with MSG_PID=0xA, MSG_SID=0x1, INT_0=0x0 and applicable MSG_DATA. If the debugger requests a frequency, BPMPFW responds with a frequency that is the highest supported by the SoC less than or equal to the one requested by the debugger. If the debugger requests a number of lanes, BPMPFW responds with the highest number of supported lanes by the SoC less than or equal to the one requested by the debugger.

   Note: BPMPFW does not need to poll for INT_0=1. When INT_0=1, it receives an interrupt.

3. The debugger waits until an ACK. When the debugger sees the ACK, if the SoC capabilities are conveyed by BPMPFW, the debugger sends a message with MSG_PID=0x1, MSG_SID=0x2, INT_0=0x1 and applicable MSG_DATA to BPMPFW to configure HSSTP. Typically, these settings are the same as those recommended by BPMPFW in the previous step. If the debugger knows the SoC capabilities, it can skip the previous two steps.

4. Based on configuration requests (MSG_PID=0x1, MSG_SID=0x2, INT_0=0x1 and applicable MSG_DATA), BPMPFW interprets frequency requested as the highest frequency supported by the SoC that is less than or equal to the requested frequency by the debugger. Similarly, it interprets the number of lanes requested as the highest number of lanes supported by the SoC that is less than or equal to the requested number of lanes by the debugger.

   Note: If BPMPFW does not understand the debugger request, it responds with MSG_PID=0xE, MSG_SID=0x1, MSG_DATA=0x0, INT_0=0. When the debugger sees this response, it rectifies its message and retries, if necessary.

Part-1b: BPMPFW does basic programming of controller with preferred defaults

This reduces the amount of programming the debugger needs to do if it is using preferred defaults. The debugger can choose to override some of these settings as described in Part-2.

1. (Optional) Security checks: Check if system permits switching the UPHY lanes. If not, respond with MSG_PID=0xE, MSG_SID=0x2, MSG_DATA=0x0, INT_0=0. Skip the rest of the programming and exit the flow.
2. BPMPFW configures CSITE_CLK to 625MHz.

   Ensure CSITE_CLK frequency is higher than hsstp_l%_clk (HSSTP parallel lane clock) frequency. The serial frequency can be inferred from MSG_DATA value provided by the debugger. The parallel frequency is $1/20^{th}$ the serial frequency. Typically, use CSITE_CLK=625MHz (REF_PLLE).

   CLK_RST_CONTROLLER_CLK_SOURCE_CSITE_0=PLLREFE_VCOCLK

   (This enum is called ‘refPLLE_gated’ because there is a clock gate on the DFD branch of the VCOOUT; use the CLK_RST_CONTROLLER_CLK_OUT_ENB_PLLREFE_0 register to enable it, if it is disabled.)

   CLK_RST_CONTROLLER_CLK_SOURCE_CSITE_0[CSITE_CLK_DIVISOR]=0
   CLK_RST_CONTROLLER_CLK_OUT_ENB_CSITE_SET_0[SET_CLK_ENB_CSITE]= ENABLE

3. BPMPFW programs DFD/HSSTP with recommended settings considering preferences suggested by the debugger.
   a. Use serial frequency from Part-1a:Step-4 in the RATE_ID table to infer RATE_ID.
   b. Program this RATE_ID in CORESIGHT_CFG_HSSTP_SPEED_CFG[RATE_ID]
   c. Infer lane mapping from MSG_DATA.
   d. Based on the number of lanes from Part-1a:Step-4 and based on the SoC/platform configuration, program lane map in CORESIGHT_CFG_HSSTP_LANE_CFG[HSSTP_TO_UPHY_LANE_MAP].
      For example, if the platform supports HSSTP over UPHY Lanes-0,1 and the debugger requests single lane, then program: HSSTP_TO_UPHY_LANE_MAP=4’b0001.
      If the platform supports HSSTP over UPHY Lanes-0,1 and the debugger requests dual lane, then program: HSSTP_TO_UPHY_LANE_MAP=4’b0011

If the platform supports HSSTP over UPHY Lanes-4 and the debugger requests single lane, then program: HSSTP_TO_UPHY_LANE_MAP=4'b0100.

e. (Optional) Functional checks: Inform the necessary software entities that the lanes will be repurposed for HSSTP and will not be available for other usage until a subsequent cold system reset (also known as L0/L1 reset).

f. Program CORESIGHT_CFG_HSSTP_INTERRUPT[MSG_0] with the inferred frequencies and number of lanes from Part-1a:Step-4, along with MSG_PID=0xA, MSG_SID=0x2. However, do not clear CORESIGHT_CFG_HSSTP_INTERRUPT[INT_0] yet as further settings are necessary.

4. BPMPFW programs DFD with general recommended settings. In most cases, these settings match the reset value in which case BPMPFW need not program them (noted below for information purposes).
Note: If they match the reset value, they are gray'ed out.

a. In NVLPCR
   i. LINK_nDALT_VAL=1'b1
   ii. FLIP_BIT_ORDER=1'b1
   iii. FLIP_BYTE_ORDER=1'b0
   iv. SEND_PRIMING_SEQ=1'b0
   v. FORCE_ASSERT_STPCR_NRST=1'b0
   vi. CHANNEL_BUSY_WAIT_INDEX=5b'10100
   vii. TX_DATA_EN_SRC_SEL=1'b0

b. In NVFFCR
   i. NOCRC_CFG=1'b0
   ii. SW_TRIG_OUT=1'b0

c. In NVFTCR
   i. MIN_FREE_CNT=0x0B
   ii. MAX_FREE_CNT=0x1D
   iii. SRC_THROTTLING=1'b0
   iv. CG_THROTTLING=1'b1
   v. TRACECTL_BASED_FILTER=1'b1
   vi. HALFWORD_SYNCPKT_FILTER=1'b1
   vii. FULLFRAME_SYNCPKT_FILTER=1'b0
   viii. FSP_FILTER_LEN=0x10

d. In NVFLCR
   i. MAX_VALID_FRAME_LENGTH=0x100
   ii. MAX_EMPTY_CYCLE_COUNT=0x100

e. In NVPTCR
   i. MIN_FREE_CNT=0x03
   ii. MAX_FREE_CNT=0x06

f. In NVUL{*}_AUX_CTL_1,

i. AUX_TX_IDDQ=0
ii. AUX_TX_TERM_EN=0
iii. AUX_TX_RDET_EN=0
iv. AUX_TX_RDET_BYP=0
v. AUX_TX_RDET_CLK_EN=0

Note: * indicates applicable only for lanes on which HSSTP is enabled.

g. In NVUL{*}_MISC_CTL_1
   i. RESET_=0
   ii. TX_IDDQ=1
   iii. TX_SLEEP=2'b11
   iv. TX_DATA_READY=0
   v. TX_DATA_EN=0
h. In NVUL{*}_DYN_CTL_1
   i. TX_DRV_AMP=0x27
   ii. TX_DRV_POST=0
   iii. TX_DRV_PRE=0
i. In NVUL{*}_MISC_CTL_3
   i. TX_RATE_PDIV_=2'b01 //(implies interface width = 20bits)

5. BPMPFW programs HSSTP with the recommended settings below:
   a. In STPCR
      i. nSW_RST=0
      ii. nRST=0
      iii. STP_EN=0
      iv. STP_PWRUP=0
   b. In STPLSR
      i. If HSSTP_TO_UPHY_LANE_MAP[3:0]==4'b0011, program STP_LANESEL[0]=1'b1; else leave it to its reset value (1'b0).
   c. In PCSR
      i. SYNC_COUNT=0x1000 (decimal 4096)
   d. Program LLIR = 0x7FFFF387
      i. VERIFY_LEN=9'h1FF (d'511)
      ii. BOND_LEN=9'h1FF (d'511)
      iii. ALIGN_LEN=13'h1387 (d'4999)
   e. Program LLIR2 = 0x000081
      i. VERIFY_MUL=0x0
      ii. BOND_MUL=0x0
      iii. ALIGN_MUL=0x8
      iv. LLIR2[3:0] should always be 4'b0001

Part-2: BPMP-FW sets up UPHY lanes and Clocks for HSSTP

Part-2a: BPMPFW qualifies the request and configures HSSTP controller.

1. BPMPFW enables MGMT_CLK for the associated UPHY PLL. This is normally enabled during boot and independent of which controller is selected on the UPHY lane. However, in case it is not enabled, enable relevant MGMT_CLK.

Part-2b: BPMPFW powers down UPHY lanes for HSSTP

1. (Optional) BPMPFW informs hypervisor/CPU that the functional interface/device on lanes being repurposed for HSSTP are no longer available as lanes are being reassigned for HSSTP.
2. BPMPFW powers down the intended UPHY lanes that are switched to HSSTP. This step can be skipped if the UPHY lane is already powered down. To power down the UPHY lanes, perform the following programming (perform each sub-step for all applicable lanes before moving to the next sub-step; i.e., first do 'a' for all HSSTP lanes, then do 'b' for all HSSTP lanes, etc.):
   a. UPHY_LANE_MISC_CTL_1_0: TX_DATA_EN, TX_DATA_READY, RX_DATA_EN =all(0)
   b. UPHY_LANE_MISC_CTL_1_0: TX_DATA_OVRD, RX_DATA_OVRD =all(1)
   c. UPHY_LANE_MISC_CTL_3_0: TX_RATE_PDIV, RX_RATE_PDIV =all(2'b01) //20bits
   d. UPHY_LANE_MISC_CTL_1_0: TX_SLEEP, RX_SLEEP =all(2'b11) //P3
   e. UPHY_LANE_MISC_CTL_1_0: TX_PWR_OVRD, RX_PWR_OVRD =all(1)
   f. UPHY_LANE_MISC_CTL_1_0: TX_IDDQ, RX_IDDQ =all(1)
   g. UPHY_LANE_MISC_CTL_1_0: TX_IDDQ_OVRD, RX_IDDQ_OVRD =all(1)

Part-2c: BPMPFW sets up UPHY PLL for HSSTP

Refer to the *HSSTP PLL Mappings* section regarding how BPMPFW decides which UPHY PLL to assign for HSSTP.

1. BPMPFW decides which UPHY PLL to use for UPHY lanes assigned to HSSTP.
   a. BPMPFW can read CLK_RST_CONTROLLER_HSIO_UPHY_PLL_SEL_CFG*_0[HSIO_UPHY_L*_PLL_SEL] to know which PLL the UPHY lane was previously using. BPMPFW can read UPHY_PLL_CTL_1_0[RATE_ID] to determine the frequency of a PLL.
   b. If the assigned UPHY PLL for the UPHY lanes in concern is already used for 10GHz serial frequency for its functional usage, then the same UPHY PLL can be used.
   c. If PLL<>LANE mapping or PLL frequency information cannot be scavenged easily at run-time, BPMPFW does know it, as part of boot-time configuration, because it initializes UPHY for functional reasons. It can use that information to determine which UPHY PLL to use for HSSTP UPHY lanes.
   d. For automotive platforms, HSSTP is primarily targeted over UPHY lanes 0/1, which are otherwise used for USB and use UPHY PLL0 at 10GHz. Hence, in typical automotive cases, HSSTP UPHY lanes are Lane-0/1 and using UPHY PLL0 at 10GHz.

2. BPMPFW polls for idle state of CAR PLL FSM (CLK_RST_CONTROLLER_HSIO_UPHY_PLL*_CFG0_0[HSIO_UPHY_PLL*_SEQ_STATE] != 2 for the appropriate UPHY PLL) and takes controls through software registers present in the CAR space.
   a. If the FSM does not come to idle state, then upon wait expiration, proceed to the next state (w*ait expiration limit should be **500us or more)***.
3. Always follow the steps below (a & b). If PLL is ON, these steps ensure that it remains ON. If PLL is OFF, they ensure that PLL gets turned ON. (**Note**: Whether PLL is at 10/16 is determined at boot based on BCT. So, at boot, it is known which PLLs are 10G versus 16GHz. Also, read the UPHY_PLL_CTL_1_0.RATE_ID registers for the PLLs of interest, to determine the frequency of the PLL being mapping to.)
   a. Program respective PLL's CLK_RST_CONTROLLER_HSIO_UPHY_PLL<0/1/2/3>_CFG0_0 registers which act as source with the following:
      i. HSIO_UPHY_PLL*_PLL_PD_OVERRIDE=1'b0
      ii. HSIO_UPHY_PLL*_LANE_PD_OVERRIDE=1'b0
      iii. HSIO_UPHY_PLL*_PLL_RESET_OVERRIDE=1'b0
      iv. HSIO_UPHY_PLL*_SEQ_IN_SWCTL=1'b1

   b. Poll for HSIO_UPHY_PLL*_SEQ_STATE to be ON

Part-2d: BPMPFW switches lane controller to HSSTP

1. BPMPFW switches UPHY lane controller for the respective UPHY lanes to HSSTP by programming the corresponding UPHY_LANE_MUX_0[HSSTP_ACTIVE]=1'b1

Part-2e: BPMPFW enables lane clocks for HSSTP

1. BPMPFW configures registers to indicate the lane_map and rate_id configuration is valid by programming CORESIGHT_CFG_HSSTP_ACTIVATE_CFG:
   a. {RATE_ID_VALID, UPHY_LANE_MAP_VALID, UPHY_LANE_CTL_VALID} = 3'b111
2. Program CLK_RST_CONTROLLER_HSIO_UPHY_HSSTP_CLK_CTRL_0[HSIO_UPHY_HSSTP_PLL_SEL_VALID] to 1'b1.
3. BPMPFW programs common clock enable CLK_RST_CONTROLLER_HSIO_UPHY_HSSTP_CLK_CTRL_0[HSIO_UPHY_HSSTP_CLK_ENABLE] =1'b1.
4. BPMPFW programs CAR registers to ungate the clock for the respective lanes mapped to HSSTP by programming the respective clock enable register fields HSIO_UPHY_HSSTP_L{0/1/4/5}_CLK_ENABLE of register CLK_RST_CONTROLLER_HSIO_UPHY_HSSTP_CLK_CTRL_0 to 1'b1.
5. As RX clocks are not needed during normal functional operation, HSIO_UPHY_HSSTP_L{0/1/4/5}_RX_CLK_ENABLE should be 1'b0.

Part-2f: BPMPFW resets, initializes, and powers up UPHY lanes for HSSTP

1. **[Action:Reset]** BPMPFW toggles functional and CFG reset for the lane using:

a. SWR_PEX_USB_UPHY_L{0/1/4/5}_RST and
b. UPHY_LANE_DIRECT_CTL_2[CFG_RESET_] for respective UPHY lanes, i.e., UPHY_LANE{0/1/4/5}.UPHY_LANE_DIRECT_CTL_2[CFG_RESET_]

2. **[Action:Initialize]** BPMPFW programs init array for HSSTP; configuring various internal registers within the UPHY using the UPHY's CFG interface (i.e., CFG_{WDS,RDS,ADDR,WDATA,RDATA}) using UPHY_LANE{0/1/4/5}.UPHY_LANE_DIRECT_CTL_2_0[CFG_RDS/WDS/ADDR/WDATA].

   [Action:Power Up] BPMPFW powers up the UPHY lane(s) intended for HSSTP and configures them for the desired operation. HSSTP controller configuration (**step-4**) and the control tie-offs take care of all the correct values. So, this part involves a smooth transition from removing the OVRD enforced values to those driven by the HSSTP controller and then powering up the lanes using the controls in the HSSTP controller.
3. For all applicable lanes where OVRDs were enforced previously, de-assert them as below:
   a. UPHY_LANE_MISC_CTL_1_0: TX_IDDQ_OVRD=0
   b. UPHY_LANE_MISC_CTL_1_0: TX_PWR_OVRD=0
   c. UPHY_LANE_MISC_CTL_1_0: TX_DATA_OVRD=0
4. Using controls in DFD, power up the lanes as shown below. '#' implies index for registers corresponding to applicable lanes (#=0/1/2/3 for UPHY lanes 0/1/4/5 resp.). Perform each step/sub-step for all lanes before moving to the next step.
   a. CORESIGHT_CFG_NVUL#_MISC_CTL_1:
      i. RESET_ =1. Wait 1us.
      ii. TX_IDDQ =0. Wait 50ns.
      iii. TX_SLEEP =2'b00. Wait 500ns.
      iv. TX_DATA_READY =1. Wait 50ns.
   b. NVLPCR
      i. TX_DATA_EN_SRC_SEL =1
      ii. FORCE_ASSERT_STPCR_NRST =1

Part-2g: BPMPFW acknowledges the debugger request

1. BPMPFW records the dynamic switch action so as to restore HSSTP as lane-controller, instead of boot-time configured lane-controller, after an SC7 entry-exit cycle. Upon SC7 exit, restore the state of CORESIGHT_CFG, HSSTP, UPHY lanes/PLL and clocks up to this point.
2. BPMPFW clears CORESIGHT_CFG_HSSTP_INTERRUPT[INT_0] as an ACK to the debugger.

Part-3: Debugger enables HSSTP controller

1. The debugger waits for an ACK before proceeding, then clears CORESIGHT_CFG_HSSTP_INTERRUPT.
2. (Optional) The debugger then programs HSSTP related configuration registers in CSITE_CFG, HSSTP and TPIU. Normally, BPMPFW configures HSSTP/TPIU as per recommendations. However, this provides an opportunity to change certain settings as required without requiring a BPMPFW upgrade for minor tweaks and experiments.

3. Debugger programs HSSTP to start (e.g., de-assert nRST, assert STP_EN, etc.)
   a. NVPTCR:
      i. MAX_FREE_CNT = 0x1 // prevents TIS from going downstream before downstream reset is deasserted by causing MIN_FREE_CNT>MAX_FREE_CNT
   b. STPCR:
      i. STP_PWRUP =1
      ii. nRST =1
      iii. nSW_RST =1
      iv. STP_EN =1
   c. NVPTCR:
      i. MAX_FREE_CNT = 0x6 // restores MIN_FREE_CNT<MAX_FREE_CNT
4. The debugger programs TPIU with the default settings below.
   a. CORESIGHT_TPIU_CSTPIU_REGS_LAR_0=0xC5ACCE55
   b. CSTPIU_REGS_CURRENT_PORT_SIZE =32'h1000_0000
   c. CSTPIU_REGS_TRIGGER_COUNTER_VALUE =0x0
   d. CSTPIU_REGS_TRIGGER_MULTIPLIER =0x0
   e. CSTPIU_REGS_CURRENT_TEST_PATTERN_MODE=0x0
   f. CSTPIU_REGS_TPRCR =0x0
   g. CSTPIU_REGS_FSCR =0x40
   h. CSTPIU_REGS_FFCR

   Until TPIU is in normal or bypass mode, tracectl is '1'. This allows trace-filtering to prevent garbage contents from TPIU to stream to downstream sync-FIFO. Then after TPIU is configured to the correct width, changing mode to continuous is okay because the full-sync packets are 32b wide and do not appear as garbage to the downstream 32b wide trace path.

   a. EnFCont=1'b1
   b. EnFTC=1'b1

   The following are recommendations while configuring TPIU, which acts as the formatter stage in the HSSTP pipeline:

   - Always use TPIU port width as 32-bit wide
   - Preferably use TPIU in continuous mode

   5. Debugger programs CORESIGHT_CFG_HSSTP_STREAM_CTL[ALLOW_IO_TRACE_STREAMING]=1

Part-4: Debugger enables tracing

1. The debugger programs ETF, Funnels and finally ETMs, PTMs, and other trace sources, as needed to start generating trace.
2. Configure downstream components before configuring upstream components to ensure that the downstream components can handle trace sent from upstream.

Boot Guidelines for HSSTP Support

Communication standards and certification bodies require limiting EMI for any communication interface. A normal fixed frequency square wave generates considerably high EMI. Hence, to reduce EMI, spread spectrum clocking (SSC) is used. This introduced a deterministic jitter in the communication stream, thus reducing EMI significantly at the cost of small signal quality.

However, such EMI restrictions are not applicable for debug environments and hence, to improve signal quality during debug, SSC may need to be disabled. Boot flow (in the following case, boot ROM) must disable SSC while initializing PLLE based on Fuse (fuse:opt_ssc_dis) to help improve signal quality. Below is the flow to initialize PLLE with the highlighted section being the change to disable SSC based on Fuse. Only if fuse:opt_ssc_dis==1 should SSC be disabled. Otherwise, enable SSC to match legacy behavior.

After powerup, IDDQ=1 and ENABLE=0

SETUP bits = (normally 0)

IDDQ = 0

Wait 5uS

With a stable input clock, program M/N/P and configuration bits

SSC_BYP=1, BYPASS_SS=1 INTERP_RESET=1 EN_SDM=0 EN_SSC=0

ENABLE = 1 and wait for PLL to lock

IF (fuse:opt_SSC_DIS==0)

{

Load Spread Coefficients

BYPASS_SS=0

SSC_BYP=0

Wait 300nS and INTERP_RESET=0

}

where,

'BYPASS_SS' is CLK_RST_CONTROLLER_PLLE_SS_CNTL_0[PLLE_BYPASS_SS]

'SSC_BYP' is CLK_RST_CONTROLLER_PLLE_SS_CNTL_0[PLLE_SSCBYP]

Also, the decision as to whether SSC is enabled/disabled during SC7 exit is based on the same fuse.

Boot-time (Static) and Run-time (Dynamic) use cases for switching to HSSTP

Common use cases for automotive platforms are:

1. Using HSSTP over USB Type-C connector where HSSTP is configured in dual-lane (x2) mode on UPHY Lane-0 and Lane-1
2. Using HSSTP over USB Type-A connector where HSSTP in configured in single-lane (x1) mode on UPHY Lane-0 or Lane-1.

At boot time, Lane-0 is used for USB-RCM. After the boot ROM phase of boot, USB-RCM mode is not supported. Also, typically self-driving automotive platforms do not use USB storage devices. If HSSTP must be configured via BCT, it is done during the BPMP-FW phase of boot. For example, during the boot ROM phase, the port is available for USB-RCM. Then after BPMP-FW starts execution, BPMP-FW switches the port from USB to HSSTP based on BCT. This BCT based switch to HSSTP is referred to as Boot-time (or Static) switch to HSSTP.

Automotive infotainment markets use USB storage, thus, it is necessary to be able to have USB as the default configuration on such platforms and only switch to HSSTP when requested by debugger. This ability to switch the lane from its existing boot-time configuration (e.g., USB, etc.) to HSSTP based on a run-time request by an agent (e.g., external debug host/debugger) is referred as Run-time (or Dynamic) switching to HSSTP.

Refer to the steps to be performed by BPMPFW in previous section for dynamically switch the lanes based on request from debugger.

To switch back a lane that has been dynamically switched to HSSTP, to its former configured controller (e.g., USB), the chip must undergo a cold system reset (e.g., L0 or L1 reset).

HSSTP Unit Programming Guidelines

VERIFY_LEN and VERIFY_MUL

#VerifBytes = VERIFY_PATTERN_MULTIPLIER*(LLIR.VERIFY_LEN + 1)

where, VERIFY_PATTERN_MULTIPLIER

= 4, if LLIR2.VERIFY_MUL is b0000

= not defined (actually, infer as 4 (default)), if LLIR2.VERIFY_MUL is b0001

= 2^(LLIR2.VERIFY_MUL), for all other values of LLIR2.VERIFY_MUL

Program LLIR.VERIFY_LEN and LLIR2.VERIFY_MUL such that:

1. #VerifBytes is a multiple of 64
2. #VerifBytes is at least 256

Enabling hot-plug support for HT-DAM

CORESIGHT_CFG_CBDCR_0 must be programmed by MB2 based on MB2-BCT for enabling hot-plug capability facilitated via the HT-DAM adapter.

Specifically, the following settings are needed:

- CORESIGHT_CFG_CBDCR_0[SWD_JTAG_CFG]=0 (reset default)
- CORESIGHT_CFG_CBDCR_0[SWD_PORT_SEL]=1 (if USB2 Port-1 is mapped to USB Type-C)
  - CORESIGHT_CFG_CBDCR_0[SWDDFD_NVDBGSEL_EN]=1 is needed to enable hot-plug

These should be done as early as possible in boot (in MB2 during cold boot) for the subsequent stages to benefit from closed box debug and hot-plug support. They should also be restored as early as possible during SC7-exit. (**Note**: Per boot arch, they are restored in MB2-RF (counterpart of MB2) in SC7-exit).

HSSTP Usage Guidelines

TPIU and Triggers

When using TPIU in Normal mode, program '*FFCR.TrigEvt=FFCR.TrigIn=FFCR.TrigFl=0*' because triggers communicated on TRACECTL are not converted to trigger packets downstream.

When TPIU is programmed in Bypass mode, HALFWORD_SYNCPKT_FILTER and FULLFRAME_SYNCPKT_FILTER are disabled. This is to avoid filtering trace data with values 0x7FFF_7FFF or 0x7FFF_FFFF at output of TPIU.

NVPTCR

To have the ability to change reset defaults, ensure NVPTCR[MIN_FREE_CNT] >= 0x2, NVPTCR[MAX_FREE_CNT] <= 0xA and NVPTCR[MIN_FREE_CNT] < NVPTCR[MAX_FREE_CNT].

NVFTCR

To have the ability to change reset defaults, ensure NVFTCR[MIN_FREE_CNT] >= 0x0A (if CG_THROTTLING=0) / 0x01 (if CG_THROTTLING=1), NVFTCR[MAX_FREE_CNT] <= 0x1F and NVFTCR[MIN_FREE_CNT] < NVFTCR[MAX_FREE_CNT].

It is also recommended to program FSP_FILTER_LEN greater than 0x8, if CG-throttling is disabled, to avoid livelocks.

DLY_TRIG0 and DLY_TRIG1

Background

Two triggers from CTI to HSSTP have programmable delays on them: DLY_TRIG0 and DLY_TRIG1. DLY_TRIG0 can modify ALLOW_IO_TRACE_STREAMING bit, using AITS_SET / AITS_CLR, which controls the flow of trace into TPIU. DLY_TRIG1 can modify STP_EN which in turn can cause an in-progress Aurora Frame to be completed gracefully by inserting an ECP (if there is not an Aurora

Frame in progress, then ECP is not inserted. “Aurora Frame in progress” implies no ECP has yet been sent after the last SCP). These allow avoiding trace loss during debug due to an L2 reset, as shown below.

Sample Configuration

Configure CTI to route pmc2dfd_CSYSREQ_inv to TPIU’s FLUSHIN, DLY_TRIG0 and DLY_TRIG1.

Program TPIU’s FFCR[FOnFlIn]=1 to generate a flush when TPIU’s FLUSHIN is asserted and also program TPIU’s FFCR[StopFl]=1. Configure DLY_TRIG0_PERIOD=5’b10000 (64K cycles of CSITE_CLK), AITS_CLR=1’b1, DLY_TRIG1_PERIOD=5’b10001 (128K cycles of CSITE_CLK) and STPEN_CLR=1’b1. DLY_TRIG1_PERIOD and DLY_TRIG0_PERIOD are configured such that DLY_TRIG1 asserts after DLY_TRIG0. Additionally, configure DLY_TRIG1_PERIOD such that DLY_TRIG1 occurs reasonably (~100 CSITE_CLK cycles) before L2 reset occurs so that frame contents downstream of “Packing & Framing Stage” drain off-chip. Also, DLY_TRIG1_PERIOD should be configured such that DLY_TRIG1 occurs reasonably (~100 CSITE_CLK cycles) after DLY_TRIG0 to ensure all trace data flushed from TPIU has passed through the Packing & Framing stage.

Note: If RAMDUMP_EN=0 and L2 reset occurs earlier after dfd2pmc_CSYSACK_inv, PMC’s L2 reset FSM still waits in state:WAIT_CLK_STOP for PMC_IMPL_RST_REQ_TIMER_0[WAIT_CLK_STOP_VAL] +1 32kHz cycles before asserting the reset even when state:WAIT_PMIC_SEQ is skipped. Default value of WAIT_CLK_STOP_VAL=0x6 provides seven 32kHz cycles in addition to one 32kHz cycle in RAMDUMP_FSM to assert WDT_DFD_RST_ACK (also known as dfdrst_ack). Thus, by default eight 32kHz cycles provide approximately 136K cycles on CSITE=625MHz clock).

Operation

Debug is enabled when HSSTP is expected to be used. Hence, L2-Reset FSM triggers RAMDUMP FSM upon an applicable L2 reset request. RAMDUMP FSM triggers pmc2dfd_CSYSREQ_inv.

pmc2dfd_CSYSREQ_inv triggers a CTITRIGIN which in turn is routed to CTITRIGOUTs that drive TPIU’s FLUSHIN, DLY_TRIG0 and DLY_TRIG1.

First, TPIU’s FLUSHIN input is asserted, which causes TPIU to request a flush upstream. This allows all upstream contents to be flushed out.

When DLY_TRIG0 asserts, it causes ALLOW_IO_TRACE_STREAMING to clear which prevents further trace from upstream components like ETF/downsizer (on L1warm reset) to drain into TPIU (on L2warm reset). Clearing ALLOW_IO_TRACE_STREAMING also forces AFREADY input to TPIU (HSSTP) to be forced to ‘1’ thus indicating completion of trace to TPIU. TPIU eventually sends any trace contents buffered in it followed by NULL trace packets to indicate completion of trace.

Later, when DLY_TRIG1 asserts, it clears STP_EN. If any Aurora Frame has not completed (no ECP sent after last SCP), this forces it to completion by insertion of ECP and preventing further contents from upstream to be sent. This allows graceful reception of all Aurora Frames with actual trace data by receiver.

As there are approximately 8K cycles of CSITE_CLK left, they are more than sufficient for contents to drain through HSSTP pipeline downstream of Packing and Framing stage.

Eventually, after PMC's L2 Reset FSM transitions from state:WAIT_CLK_STOP to state:RESET_ASSERT, it triggers L2 reset. But because either trace data downstream of ALLOW_IO_TRACE_STREAMING has been sent to TPA (HSSTP Probe) or it has been retained in between ETF and ATB-Downsizer, both of which are on L1warm reset and hence retain their contents, no trace data is lost.

This ability to avoid trace loss is useful when WAIT_CLK_STOP_VAL is configured to a smaller value and hence DLY_TRIG1_PERIOD cannot be large enough to drain entire ETF. In which case, post L2 reset, HSSTP can be re-enabled to either drain subsequent trace over HSSTP if needed. Requiring this is a low probability also because the point of interest in trace is when first or second expirations of the crashing WDT occur. As these events occur much earlier than the fourth expiration, in most cases, those trace points have already streamed through to the external TPA or can stream through with a smaller DLY_TRIG1_PERIOD). It usually takes less than 30us to stream out ETF contents while a WDT inter-expiry delay is of the order of at least a few milliseconds.

HSSTP PLL Mappings

If HSSTP is being configured over a lane previously assigned to USB, as USB and HSSTP can use same PLL, UPHY Lane<>PLL remapping is not needed.

However, if HSSTP is being configured over a lane previously assigned to PCIe Gen-3, then

- if there does not exist an unused PLL and if the available PLLs are being used either by MPHY or PCIE Gen-3 as depicted in **corner-case#1**, BPMPFW returns an error.
- if there exists a nearby unused (disabled) PLL, either that can be enabled for HSSTP and UPHY lane<>PLL mapping reconfigured for the lane over which HSSTP is being configured as depicted in **corner-case#2**

Corner-Case#1: In the following HSIO UPHY configurations, if HSSTPx1 is enabled over Lane-4/ Lane-5, BPMPFW returns an error: "HSSTP_ERROR_NO_PLL"

Config Number	PLL0	Lane 0	Lane 1	PLL1	Lane 2	Lane 3	PLL2	Lane 4	Lane 5	Lane 6	Lane 7	PLL3
3	Disabled	USB3.1 P0	USB3.1 P1	USB3 / PCIe G2	USB3.1 P2	PCIe x1 C1 (Gen2)	PCIe C4 only	PCIe x2 C4		UFS x2		MPHY
11	USB3 / PCIe G2	USB3.1 P0	USB3.1 P1	PCIe C0 only	PCIe x2 C0		PCIe C4 only	PCIe x2 C4		UFS x2		MPHY
19	PCIe G4	PCIe x1 C0	USB3.1 P1	USB3 / PCIe G2	USB3.1 P2	PCIe x1 C1 (Gen2)	PCIe C4 only	PCIe x2 C4		UFS x2		MPHY

Config Number	PLL0	Lane 0	Lane 1	PLL1	Lane 2	Lane 3	PLL2	Lane 4	Lane 5	Lane 6	Lane 7	PLL3
35	Disabled	PCIe x4 C0		PCIe C0 only	PCIe x4 C0		PCIe C4 only	PCIe x2 C4		UFS x2		MPHY

Corner-Case#2: In the following HSIO UPHY configurations, when HSSTP is going to be activated over a lane previously assigned to PCIe Gen-3 and the lane has a nearby unused ('DISABLED') PLL, BPMP-FW must re-configure HSSTP lane PLL_SEL and enable the previously disabled PLL0 or PLL3 at 10GHz.
(a) CAR PLL_SEL
CLK_RST_CONTROLLER_HSIO_UPHY_PLL_SEL_CFG0_0[HSIO_UPHY_L0_PLL_SEL] and/or
CLK_RST_CONTROLLER_HSIO_UPHY_PLL_SEL_CFG0_0[HSIO_UPHY_L1_PLL_SEL] or
CLK_RST_CONTROLLER_HSIO_UPHY_PLL_SEL_CFG1_0[HSIO_UPHY_L4_PLL_SEL] or
CLK_RST_CONTROLLER_HSIO_UPHY_PLL_SEL_CFG1_0[HSIO_UPHY_L5_PLL_SEL] or

(b) UPHY lane cfg PLL_SEL
The UPHY PLL_SEL lane mapping can be programmed through the lane CFG port, which has four bits, one for each rate_id.
8'd1: DLN_CFG_ID_CTRL {12'h0, PLL_SEL[3], PLL_SEL[2], PLL_SEL[1], PLL_SEL[0]}

Config Number	PLL0	Lane 0	Lane 1	PLL1	Lane 2	Lane 3	PLL2	Lane 4	Lane 5	Lane 6	Lane 7	PLL3
8	USB3 / PCIe G2	USB3.1 P0	USB3.1 P1	PCIe C0 only	PCIe x2 C0		PCIe C4 only	PCIe x4 C4				Disabled
32	Disabled	PCIe x4 C0		PCIe C0 only	PCIe x4 C0		PCIe C4 only	PCIe x4 C4				Disabled
33	Disabled	PCIe x4 C0		PCIe C0 only	PCIe x4 C0		USB3 / PCIe G2	PCIe x1 C2	PCIe x1 C3	PCIe x2 C4		PCIe G4
34	Disabled	PCIe x4 C0		PCIe C0 only	PCIe x4 C0		USB3 / PCIe G2	PCIe x1 C2	PCIe x1 C3	PCIe x1 C4	USB3.1 P3	PCIe G4
35	Disabled	PCIe x4 C0		PCIe C0 only	PCIe x4 C0		PCIe C4 only	PCIe x2 C4		UFS x2		MPHY

Config Number	PLL0	Lane 0	Lane 1	PLL1	Lane 2	Lane 3	PLL2	Lane 4	Lane 5	Lane 6	Lane 7	PLL3
36	Disabled	PCIe x4 C0		PCIe C0 only	PCIe x4 C0		USB3 / PCIe G2	PCIe x2 C4 (Gen2)		UFS x1	USB3.1 P3	MPHY
37	Disabled	PCIe x4 C0		PCIe C0 only	PCIe x4 C0		USB3 / PCIe G2	PCIe x1 C2 (Gen2)	PCIe x1 C3 (Gen2)	UFS x2		MPHY
38	Disabled	PCIe x4 C0		PCIe C0 only	PCIe x4 C0		USB3 / PCIe G2	PCIe x1 C2 (Gen2)	PCIe x1 C3 (Gen2)	UFS x1	PCIe x1 C4 (Gen2)	MPHY
39	Disabled	PCIe x4 C0		PCIe C0 only	PCIe x4 C0		USB3 / PCIe G2	PCIe x1 C2 (Gen2)	PCIe x1 C3 (Gen2)	UFS x1	USB3.1 P3	MPHY

HSSTP General Notes

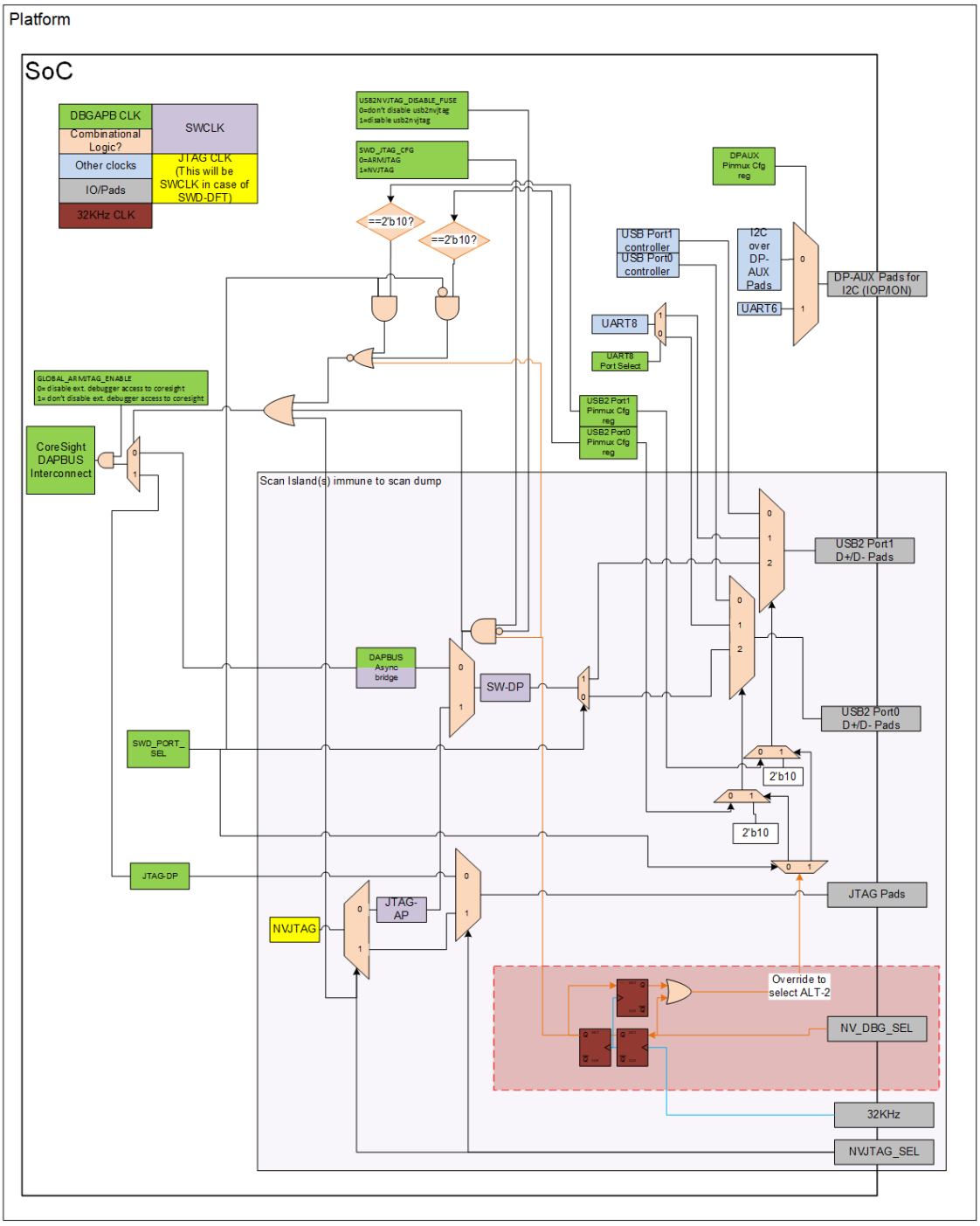
Reset Polarity and Nomenclature in CAR Register Specifications

- When CLK_RST_CONTROLLER_RST_DEV_HSSTP_0=0 (DISABLE), car2hsstp_rstn=1
- When CLK_RST_CONTROLLER_RST_DEV_HSSTP_0=1 (ENABLE), car2hsstp_rstn=0
- CLK_RST_CONTROLLER_RST_DEV_HSSTP_CLR_0=1 causes CLK_RST_CONTROLLER_RST_DEV_HSSTP_0=0
- CLK_RST_CONTROLLER_RST_DEV_HSSTP_SET_0=1 causes CLK_RST_CONTROLLER_RST_DEV_HSSTP_0=1

8.7.2.2 Closed Box Debug Over USB2

Debug (similar to ArmJTAG but over SWD) and Instrumentation (UART) features can be used via the USB Type-C connector port. In addition, UART is also supported over DP_AUX port to allow SWD and UART to be used in parallel for debug in closed box solutions where the required USB and DP_AUX functional pins (respectively) are available.

Figure 8.70 Closed Box Debug Block Diagram



8.7.2.2.1 USB2 and Dedicated JTAG Ports

Selection Matrix

The table below provides a summary of the values of different input signals and configurations, and the resulting features.

- Which on-chip controller, e.g., NVJTAG (Test Initiator, DFT JTAG), ArmJTAG (CoreSight JTAG, DFD JTAG), DFD-SWD (CoreSight SWD), USB2, and UART8, is connected to which port (USB2 Port0/1, Dedicated JTAG port) based on different configuration inputs, such as:
  - fuses (e.g., FUSE_OPT_USB2NVJTAG_DISABLE_0)
  - pins (e.g., NVDBG_SEL, NVJTAG_SEL)
  - software configurable MMIO register[bits] (e.g., SWDDFD_NVDBGSEL_EN, UART_PORT_SEL, SWD_PORT_SEL, etc.)

The colors in the System State columns highlight which debug connection (NVJTAG versus ArmJTAG versus DFD-SWD) is supported in that mode, while the green boxes with bold texts in the Inputs columns show some of the most important inputs in that connection.

Table 8.70 Selection Matrix Table

Inputs							**System State (USB2 and Dedicated JTAG ports)**		
NVJTAG_SEL	**NVDBG_SEL**	**SWD_PORT_SEL**	**UART_PORT_SEL**	**SWDDFD_NVDBGSEL_EN**	**USB2 Port0 PINMUX CFG**	**USB2 Port1 PINMUX CFG**	**Controller Connected to**		
							Dedicated JTAG pins	**USB2 Port0 pins**	**USB2 Port1 pins**
1	x	x	x	x	1	1	NVJTAG (DFT)	USB2	USB2
1	x	x	1	x	1	3	NVJTAG (DFT)	USB2	UART8
1	x	x	x	x	1	2	NVJTAG (DFT)	USB2	N/A
1	x	x	0	x	3	1	NVJTAG (DFT)	UART8	USB2
1	x	x	0	x	3	2	NVJTAG (DFT)	UART8	N/A
1	x	x	x	x	2	1	NVJTAG (DFT)	N/A	USB2
1	x	x	1	x	2	3	NVJTAG (DFT)	N/A	UART8
0	0	x	x	x	1	1	ArmJTAG (CoreSight)	USB2	USB2

Inputs							System State (USB2 and Dedicated JTAG ports)		
NVJTAG_SEL	NVDBG_SEL	SWD_PORT_SEL	UART_PORT_SEL	SWDDFD_NVDBGSEL_EN	USB2 Port0 PINMUX CFG	USB2 Port1 PINMUX CFG	Controller Connected to		
							Dedicated JTAG pins	USB2 Port0 pins	USB2 Port1 pins
0	0	x	1	x	1	3	ArmJTAG (CoreSight)	USB2	UART8
0	0	1	x	x	1	**2**	N/A	USB2	DFD-SWD (CoreSight)
0	0	x	0	x	3	1	ArmJTAG (CoreSight)	UART8	USB2
0	0	1	0	x	3	**2**	N/A	UART8	DFD-SWD (CoreSight)
0	0	0	x	x	**2**	1	N/A	DFD-SWD (CoreSight)	USB2
0	0	0	1	x	**2**	3	N/A	DFD-SWD (CoreSight)	UART8
0	1	x	x	0	1	1	ArmJTAG (CoreSight)	USB2	USB2
0	1	x	1	0	1	3	ArmJTAG (CoreSight)	USB2	UART8
0	1	1	x	0	1	**2**	N/A	USB2	DFD-SWD (CoreSight)
0	1	x	0	0	3	1	ArmJTAG (CoreSight)	UART8	USB2
0	1	1	0	0	3	**2**	N/A	UART8	DFD-SWD (CoreSight)
0	1	0	x	0	**2**	1	N/A	DFD-SWD (CoreSight)	USB2
0	1	0	1	0	**2**	3	N/A	DFD-SWD (CoreSight)	UART8
0	1	**0**	x	1	x	1	N/A	DFD-SWD (CoreSight)	USB2
0	1	**0**	1	1	x	3	N/A	DFD-SWD (CoreSight)	UART8

Inputs							System State (USB2 and Dedicated JTAG ports)		
NVJTAG_SEL	NVDBG_SEL	SWD_PORT_SEL	UART_PORT_SEL	SWDDFD_NVDBGSEL_EN	USB2 Port0 PINMUX CFG	USB2 Port1 PINMUX CFG	Controller Connected to		
							Dedicated JTAG pins	USB2 Port0 pins	USB2 Port1 pins
0	1	**1**	x	1	1	x	N/A	USB2	DFD-SWD (CoreSight)
0	1	**1**	0	1	3	x	N/A	UART8	DFD-SWD (CoreSight)

Notes:

- Once NVDBG_SEL is asserted, do not change its value.
- NVDBG_SEL is driven via USB Type-C microcontroller or as NVDBG_SEL = ~ (CC1 || CC2). Refer to the *How HT-DAM hot-plugging works* section for more details.
- USB2NVJTAG_DISABLE_FUSE input from fuse is forced to '1' in DFD (so actual fuse value is ignored in DFD)
- Fuse2all_fuse_valid indicates whether all the fuses have been sensed.
- USB2 PINMUX CFG register behavior depends on a combination of different inputs/settings and, hence, a signal in isolation may not be able to describe the state of the system.

Switching from USB2/UART to SWD after a Hard-Hang (Hot-plugging)

The BCT entries only indicate the recommended system configuration that should be configured by Boot Software post L0/L1/L2/warm reset. The entries do not indicate the registers into which these settings need to be copied.

Note: Pinmux for both Port 0 and Port 1 should not be configured for UART8 at the same time as there is only one UART8 controller. Similarly, they should not both be configured for SWD at the same time as there is only one SW-DP. However, they can both be configured for USB at the same time as they are connected to two separate USB controllers.

When the USB2 port is not configured for SWD and a hard-hang occurs, the port can be forced to SWD mode by assertion of NVDBG_SEL. To allow such dynamic switching based on NVDBG_SEL, during boot set SWDDFD_NVDBGSEL_EN to '1' as early as possible but after programming SWD_PORT_SEL.

UART_PORT_SEL, SWD_PORT_SEL, and SWD_JTAG_CFG are programmed based on MB2-BCT. These need to be restored by MB2RF post SC7 exit.

Configuring SWD, UART, or USB on a USB2 Port-0/1 can be done during boot. However, if the user needs the USB2 port for USB2 or UART purposes during normal operation, and only wants to use it for SWD when a hang occurs, this is made possible using NVDBG_SEL pin. To do this, configure the

port(s) for the required mode – USB2 or UART and program SWD_PORT_SEL to 0/1, depending on which port you want to use for SWD when the hang occurs, and set SWDDFD_NVDBGSEL_EN=1. Continue normal functional operation/use case. When the hang occurs, assert NVDBG_SEL. Based on SWD_PORT_SEL=0/1, the existing configuration of the respective USB2 Port-0/1 is ignored and instead, that port is switched to SWD mode. This ability to switch to SWD at run-time after a hang occurs, is indicated by the rows in the Selection Matrix where NVDBG_SEL=1 and SWDDFD_NVDBGSEL_EN=1. Also refer to the *How HT-DAM hot-plugging works* section for more details.

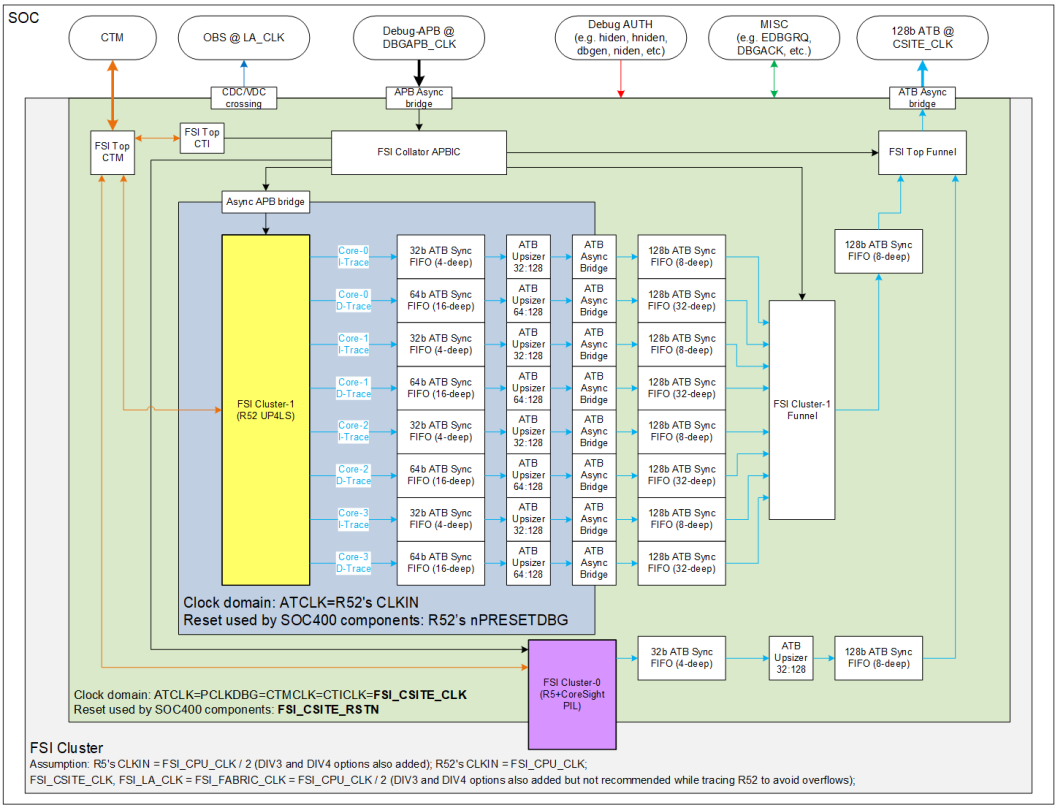
It is also possible to statically dedicate one USB2 Port for SWD by programming the pinmux for the required USB2 port as SWD during boot.

8.7.2.2.2 FSI Debug and Trace Support

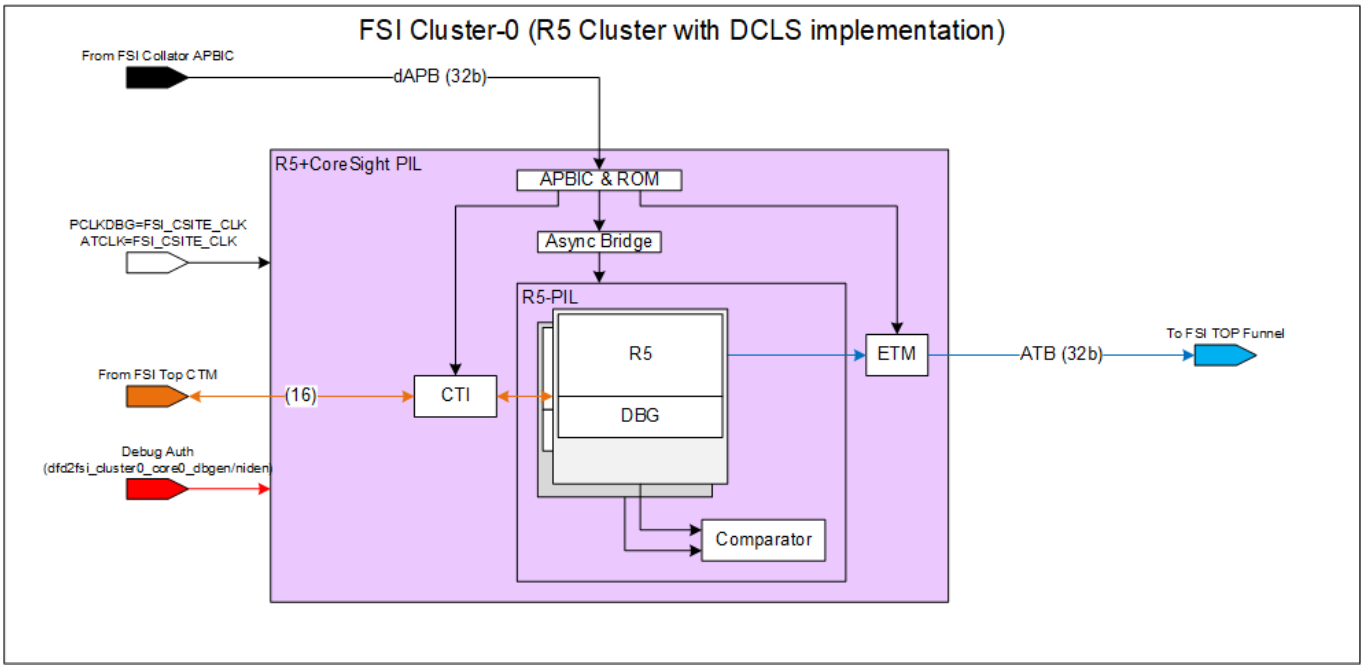
Functional Safety Island (FSI) contains two processor clusters:

- Cluster-0: An Arm Cortex-R5 based cluster with two cores acting in lock-step as one independent core. This is also referred to as Crypto Hardware Security Module (CHSM).
- Cluster-1: An Arm Cortex-R52 based cluster with eight cores acting in lock-step as four independent cores.

Figure 8.71 FSI Block Diagram



where, “FSI Cluster-0 (Cortex-R5-PIL) is as below”



FSI Clocks and Resets

Cortex-R5 processor clock (FSI_CHSM_CPU_CLK) is driven at lower (/2, /3, /4) frequency than R52 processor clock (FSI_CPU_CLK).

FSI_CSITE_CLK is used for shared debug and trace logic in FSI cluster. FSI_CSITE_CLK is an FLCG version of FSI_FABRIC_CLK. FSI_CSITE_CLK is enabled by default out of reset.

FSI_FABRIC_CLK is derived from R52 core clock (FSI_CPU_CLK) independent of the Cortex-R5 clock (FSI_CHSM_CPU_CLK). Hence FSI_CSITE_CLK can run at faster frequency (e.g., /2 frequency of R52 clock) even if Cortex-R5 is running at a slow frequency (e.g., /3-/4 frequency of R52 clock).

Clocks and Reset Connectivity for Various SOC400 Components Used in FSI

FSI_CSITE_RSTN is a software controlled L2warm reset provisioned by CAR in the FSI cluster for some CoreSight (SOC400) components in FSI. By default, FSI_CSITE_RSTN is deasserted (also known as ‘disabled’).

FSI_LA_RSTN is also a software controlled L2warm reset provisioned by CAR in the FSI cluster. By default, FSI_LA_RSTN is deasserted (also known as ‘disabled’).

The following table provides a brief overview of these connections.

Component	Port	Driven by	Comment
APB async bridge (for R52 debug-APB)	pclks	FSI_CSITE_CLK	
(see above)	presetsn	FSI_CSITE_RSTN	Sync to corresponding pclks
(see above)	pclkm	R52's CLKIN	
(see above)	presetmn	R52's nPRESETDBG	Sync to corresponding pclkm
APB async bridge (for SOC-to-FSI debug-APB crossing)	pclks	DBGAPB_CLK	
(see above)	presetsn	csitel2rstn	Sync to corresponding pclks
(see above)	pclkm	FSI_CSITE_CLK	
(see above)	presetmn	FSI_CSITE_RSTN	Sync to corresponding pclkm
FSI ATB Funnels (both, 8:1 and 2:1)	clk	FSI_CSITE_CLK	
(see above)	resetn	FSI_CSITE_RSTN	Sync to corresponding clk
ATB sync bridges (which are immediately preceding the funnels) and ATB sync bridge (which is immediately preceding the upsizer on Cortex-R5 trace path)	clk	FSI_CSITE_CLK	
(see above)	resetn	FSI_CSITE_RSTN	Sync to corresponding clk
ATB sync bridges (which are immediately preceding the upsizers on R52 trace path)	clk	R52's CLKIN	
(see above)	resetn	R52's nPRESETDBG	Sync to corresponding clk
ATB async bridges (for FSI-to-SOC ATB crossing)	clks	FSI_CSITE_CLK	
(see above)	resetsn	FSI_CSITE_RSTN	Sync to corresponding clks
(see above)	clkm	CSITE_CLK	
(see above)	resetmn	csitel2rstn	Sync to corresponding clkm
ATB async bridges (which take trace from R52's CLKIN to FSI_CSITE_CLK)	clks	R52's CLKIN	
(see above)	resetsn	R52's nPRESETDBG	Sync to corresponding clks
(see above)	clkm	FSI_CSITE_CLK	
(see above)	resetmn	FSI_CSITE_RSTN	Sync to corresponding clkm

Component	Port	Driven by	Comment
ATB upsizer on R52 ETM outputs (both, 32:128 and 64:128)	clk	R52's CLKIN	
(see above)	resetn	R52's nPRESETDBG	Sync to corresponding clk
ATB upsizer on Cortex-R5 ETM output (32:128)	clk	FSI_CSITE_CLK	
(see above)	resetn	FSI_CSITE_RSTN	Sync to corresponding clk
FSI Collator APBIC	clk	FSI_CSITE_CLK	
	resetn	FSI_CSITE_RSTN	Sync to corresponding clk
FSI Top CTM	ctmclk	FSI_CSITE_CLK	
	ctmclken	Tie-off HIGH (1)	
	ctmresetn	FSI_CSITE_RSTN	Sync to corresponding ctmclk
FSI Top CTI	cticlk, pclkdbg	FSI_CSITE_CLK	
	cticlken, pclkendbg	Tie-off HIGH (1)	
	ctiresetn, presetdbgn	FSI_CSITE_RSTN	Sync ctiresetn to cticlk; Sync presetdbgn to pclkdbg;

Connectivity for CSYREQ and CSYSACK of APB/ATB async bridges used for voltage crossing

The Arm SOC400 components, APB Async Bridge and ATB Async Bridge, used for SOC-to-FSI and FSI-to-SOC power-domain crossings of debug-APB and ATB interfaces, respectively, have their CSYSREQ/CSYSACK connectivity as follows:

RTL Guideline

1. FSI side of async bridge uses FSI_CSITE_RSTN
2. SoC side of async bridge uses CSITEL2RSTN
3. Add the following MMIO reg in FSI: FSI_MISC_ISOLATION_DFD_0 and FSI_MISC_ISOLATION_DFD_ACK_0
   a. FSI_MISC_ISOLATION_DFD_0[ATB] (R/W) drives CSYSREQ input of ATB async bridge; FSI_MISC_ISOLATION_DFD_0[APB] (R/W) drives CSYSREQ input of APB async bridge; their reset value is '1' to ensure default cluster interface is not off after power-up.
   b. FSI_MISC_ISOLATION_DFD_ACK_0[ATB] (RO) is driven by CSYSACK output of ATB async bridge. FSI_MISC_ISOLATION_DFD_ACK_0[APB] (RO) is driven by CSYSACK output of APB async bridge.
4. FSI2SOC signals: FSI_MISC_ISOLATION_DFD_0[ATB] and FSI_MISC_ISOLATION_DFD_0[APB] have clamp value '0'

Programming Guideline

When FSI-SW decides to isolate debug interfaces in mission mode (when FSI debug is disabled) by asserting FSI_CSITE_RSTN and gating FSI_CSITE_CLK, follow the sequence below:

1. Program FSI_MISC_ISOLATION_DFD_0[ATB]=FSI_MISC_ISOLATION_DFD_0[APB]=0
2. Poll FSI_MISC_ISOLATION_DFD_ACK_0[ATB]=FSI_MISC_ISOLATION_DFD_ACK_0[APB]=1
3. Assert FSI_CSITE_RSTN
4. Turn off FSI_CSITE_CLK

Debug interfaces use Arm's SOC400 async bridges for MV crossing on Arm defined interfaces. Arm recommends using CSYSREQ and CSYSACK to quiesce the interface before turning either side OFF.

FSI Cache and TCM Debug

R52 Caches

CFGL1CACHEINVDISx inputs to each core are driven by independent MMIO bits (reset default: '0') in FSI (FSI_MISC_CPU_x_CFG_0.nCACHE_INVALIDATE). Each control is AND'ed with corresponding R52's HIDEN before driving the R52's CFGL1CACHEINVDISx input. This is to facilitate per-core control and security granularity. The MMIO bit survives R52 cluster resets - nCORERESETx and nCPUPORESETx. It is accessible via CBB accesses with MSTRID:CSITE when FSI debug is enabled.

When driven high, it disables automatic hardware controlled L1-cache invalidation across nCORERESETx or nCPUPORESETx. Thus, post these resets, the debugger can connect to the core and read out the cache contents.

R52 being an Armv8 based architecture provides instructions for much better visibility into its caches compared to Cortex-R5, which is an Armv7 based architecture. Refer to the R52-TRM section 7.4 *Debug access to internal memory* for details regarding R52 cache debug.

R52 TCMs

R52 TCMs can be accessed using the AXIS interface into the corresponding R52 cores.

Cortex-R5 Caches and TCMs

FSI-R5's R5AXISLAVE AMAP aperture does not include the Cortex-R5 cache range. Thus, FSI-R5 caches are not accessible via its R5AXISLAVE port.

FSI Cross-trigger Support

Refer to the *CTI Mappings in R5-CoreSight PILs* section for CTI connectivity within the R5-Cluster.

Refer to the *CTI Mappings in R52 Cluster* section for CTI connectivity within the R52-Cluster.

Refer to the *FSI Top CTI Mappings* section for CTI connectivity of FSI Top CTI.

FSI EDBGRQ, DBGACK and Freezing Timers

Refer to the *EDBGRQ, DBGACK and freezing timers during debug* section. The SOC2FSI and FSI2SOC signals (e.g., EDBGRQ, DBGACK and dfd2all_freeze_timers_req) are additionally masked/clamped as described in the *FSI FFI in context of debug* section.

FSI WDT

Refer to the *Watchdog for Debug and Recovery* section for details on WDT expiration behavior, PMC_IMPL_RAMDUMP_CTL_STATUS_0, TKE debug controls, wdt-freeze, etc.

At a high level:

FSI WDTs expirations are connected/mapped as follows:

1st expiration – corresponding servicing core's IRQ

2nd expiration – corresponding servicing core's FIQ

3rd expiration – FSI Top CTI's CTITRIGIN (refer to the *FSI Top CTI Mappings* section for details)

Note: FSI TKE exports only one/common 3rd expiration, although it has five WDTs in it. This is also routed to FSI-R5's AVIC and FSI-R52's GIC.

4th expiration – L2 reset request to PMC (also, with RAMDUMP support)

5th expiration – L1 reset request to PMC

dfd2all_wdt_freeze drives WDT's freeze input. As FSI-TKE exports a single/common port for all the WDT instances contained in it, its DBGEN input is driven by AND-ed version of all dbgen inputs of cores associated with the contained WDTs instances.

FSI TSC

FSI TSC Freeze

FSI has a separate TSC from the SoC TSC. However, dfd2all_freeze_timers_req drives the freeze_timers_req input for FSI TSC. As this signal is masked/clamped to '0' as part of FFI debug signals, FSI TSC's freeze_timers_req input being driven from dfd2all_freeze_timers_req in the SoC is not a concern.

FSI TSC Unit of Increment

A single unit in FSI TSC represents 32ns (31.25MHz). However, the FSI TSC interface is on a slightly faster 40MHz clock (fsi_xtal_clk). For comparison, a single unit increment in the SoC TSC also

represents 32ns (31.25MHz) but the SoC TSC interface is on a tscref_clk, which is also 31.25MHz (a pulse skipped version of slightly faster 38.4MHz (SOC-OSC) clock).

Using FSI-TSC versus SOC-TSC for R52 ETM Timestamping

Use the same FSI-TSC to correlate FSI-CPU actions and trace with other events in FSI. It does make correlating FSI trace with SoC trace a bit difficult given that FSI TSC starts from 0 when FSI boots and SoC TSC is at a different value by that time. The probability of correlating FSI software events with FSI hardware events during debug is higher than the probability of correlating FSI software with SoC software or hardware events. Hence, using FSI-TSC for R52 is the best option.

Using Binary Timestamp

For uniformity, binary timestamping is used. As the binary format of TSC is used only within the chip and async FIFOs do clock crossings, using binary TSC offers no disadvantage while ensuring uniformity with existing architecture.

FSI_CSITE_TSINTP Render Configuration

Use the following render configurations:

SCLK_R=64 (default)
CLK_SCLK_R=256 (default)

Note: While maximum R52 is 1200MHz and TSC unit increment is constant at 31.25MHz allowing SCLK_R=1 and CLK_SCLK_R=39 (and hence N=6), ARM recommends SCLK_R=64 and CLK_SCLK_R=256. As using the recommended settings has no major drawback, render configurations are set per recommended settings.

Connectivity

High-level connectivity:

1. FSI-TSC (56b binary) -> Async-FIFO -> FSI_CSITE_TSINTP -> R52's TSVALUEB input.
2. FSI_CSITE_TSINTP
   a. R52's TSVALUEB[63:0] = FSI_CSITE_TSINTP's tsvalueintpb[63:0]
   b. FSI_CSITE_TSINTP's tsvalueb[63:0] = {8'b0, 56'b(56bit binary timestamp output from Async FIFO)}
   c. FSI_CSITE_TSINTP's clk = R52's CLKIN
3. Async-FIFO
   a. Async-FIFO's read-side (also known as output side) should be on R52's CLKIN
   b. Async-FIFO's write-side (also known as input side) should be FSI_XTAL_CLK, which is a separate crystal-based clock input to FSI operating at 40MHz.
   Note: TSC tick is still 31.25 MHz (32ns), however the launch clock is FSI_XTAL_CLK (40MHz). That is, timestamp changes at 31.25MHz but changes are conveyed at 40MHz (so two subsequent cycles of 40MHz may not always convey a change as that is faster than the rate at which change occurs, which is 31.25MHz).

c. FSI_CSITE_TSINTP does not have a signal to indicate whether its input is valid or not. Hence, when there is no data to be read from Async-FIFO, the last valid data read from Async-FIFO should be held at the input of FSI_CSITE_TSINTP until the next valid data can be read from the Async-FIFO (e.g., use data_valid to enable/disable latch that drives; the example is just a suggestion for implementation).
d. Async-FIFO must never backpressure on the write-side (e.g., write_enable connected to 1'b1; the example is just a suggestion for implementation).

4. Ownership
   a. Only FSI_CSITE_TSINTP is provided by DFD with default render configurations. Async-FIFO is not provided by DFD.
   b. FSI must hook it up as per the connectivity described above, including the Async-FIFO.

Calculating Time-Interval Using Timestamps

Let TS_A be timestamp at time T_A and TS_B be timestamp at a later time, T_B.

Let T_{B-A} be time difference between T_B and T_A. It is calculated as follows:

$T_{B-A} = ((TS_B-TS_A)/256)*32ns = ((TS_B-TS_A)/8)ns$

FSI ALIVE STATUS

The following signals from FSI are exported to DFD for proof-of-life and access control.

Register: CORESIGHT_CFG_FSI_CLUSTER1_ALIVE_STATUS_0

Bit	Type	Category	Signal(s)	Comments
[0]	-	-	-	RAZ/WI
[18:0]	RO	Resets	[nTOPRESET, nPRESETDBG, nCORERESETx, nCPUPORESETx, nCORERESETDCLSx, nCPUPORESETDCLSx]	Input(s) to R52 cluster
[22:19]	RO	Reset related	CPUHALTx	Input(s) to R52 cluster
[26:23]	-	-		RAZ/WI
[27]	RO	Isolation	FSI-SOC MV APB Async Bridge's CSYSACK	
[28]	RO	Isolation	FSI-SOC MV ATB Async Bridge's CSYSACK	
[29]	RO	Isolation	FSI-SOC MV APB Async Bridge's CSYSREQ	
[30]	RO	Isolation	FSI-SOC MV ATB Async Bridge's CSYSREQ	
[31]	RO	-	-	-

Register: CORESIGHT_CFG_FSI_DBG_ALIVE_STATUS_0

Bit	Type	Category	Signal(s)	Comments
[15:0]	RO	Debug	dfd2fsi_cluster1_core<x>_<d>	These signals are procured locally from within DFD and not exported from FSI to DFD. <x> ∈ {0,1,2,3}, <d> ∈ {hiden, hniden, dbgen, niden},
[17:16]	RO	Debug	dfd2fsi_cluster0_core0_<p>	These signals are procured locally from within DFD and not exported from FSI to DFD. <p> ∈ {dbgen, niden};
[21:18]	RO	Resets	FSI Cortex-R5's {nRESET, DBGRESETn, PRESETDBGn, nSYSPORESET}	Input(s) to Cortex-R5 cluster
[22]	RO	Resets	FSI_CSITE_RSTN	Input(s) to some (not all) SOC400 components in FSI cluster
[23]	RO	-		
[24]	RO	Clock	FSI_FABRIC_CLK_TOGGLE	This is a 32b counter on FSI_FABRIC_CLK. Overflow bit toggles side-band to SOC-DFD. This bit toggles at a very low frequency of around 1/(2^32) the frequency of the FSI_FABRIC_CLK.
[25]	RO	Clock enable	FSI_CHSM_CPU_CE	FSI-R5 clock (FSI_CHSM_CPU_CLK) enable
[26]	RO	Clock enable	FSI_CSITE_CE	FSI_CSITE_CLK's enable.
[27]	RO	-		
[28]	RO	-	-	-
[30:29]	RO	Power	FSI's Power clamps	Signals come from PMC (pmc2fsi_rg_clamp and the pmc2fsi_clamp_pre_fs).
[31]	RO	Debug	FSI Debug-APB ACCESS-TIMEOUT	Generated within SOC-DFD locally.

CORESIGHT_CFG_FSI_CLUSTER1_ALIVE_STATUS_0 and CORESIGHT_CFG_FSI_DBG_ALIVE_STATUS_0 are protected by a common SCR: CORESIGHT_CFG_SCR_FSI_ALIVE_STATUS_0.

Note: The counter used in FSI_FABRIC_CLK_TOGGLE is enabled only when FSI-R5's DBGEN (also known as dfd2fsi_cluster0_core0_dbgen) is HIGH.

FSI FFI in Context of Debug

Freedom From Interference (FFI) implies methods to ensure FSI functionality is not impacted by other units that do not meet the necessary safety rating. Provisions are available to prevent DFD from impacting FSI functional behavior due to any DFD failures.

Isolation through reset and clock gating

This approach relies on FSI_CSITE_RSTN assertion and FSI_CSITE_CLK gating (refer to the *Connectivity for CSYREQ and CSYSACK of APB/ATB async bridges used for voltage crossing* section).

To avoid flush from hanging when FSI is not being traced but logically isolated, ensure Major ATB Funnel does not enable FSI ATB port.

CNA Fault Detection

Check Not Asserted (CNA) Fault Detection is a mechanism to generate faults when a signal takes a value that it is otherwise not expected to take during mission mode. In addition to signal clamping/masking, debug related signals that are not expected to be active can be plugged into such CNA logic to generate faults.

FSI Cluster-0 debug related signals included in CNA list are:

- inv(ETMPWRUP)

ETMPWRUP must be '1' for R5-DCLS to function correctly. As R5-DCLS CNA mechanism generates faults when selected signals are '1', this signal is inverted before plugging into the Cortex-R5 cluster's CNA logic.

FSI Cluster-1 debug related signals included in CNA list are:

- R52 cluster's HIDENx, HNIDENx, DBGENx, NIDENx

By default, CNA faults only cause interrupts to R52 GIC/Cortex-R5 VIC. The FSI-R52 GIC/FSI-R5 VIC by default ignore interrupts out of reset. FSI software must follow guidelines mentioned in the *FSI Boot/Software Impact* section to not mask these CNA errors/faults when debug intent is established for FSI in the system. As debug intent is established much before FSI software starts execution, this should not be a problem for FSI software.

FSI Boot/Software Impact

Software configures FSI_CSITE_RSTN and FSI_CSITE_CLK based on BCT. FSI software programs FSI_MISC_ISOLATION_DFD_0[ATB, APB] on the platform when it asserts FSI_CSITE_RSTN and disables FSI_CSITE_CLK for isolation.

If FSI debug is enabled, FSI software must not engage debug signals isolation while entering "cocoon mode" or while trying to isolate itself. This is necessary to debug issues around cocoon mode and hence, debug interfaces have separate isolation controls.

FSI-SW must mask all CNA errors/faults/interrupts associated with Cortex-R5 and R52 debug signals if FSI debug is enabled in the system.

FSI_LA_CLK has separate FLCG enable/disable controls for software in FSI-CAR. FSI-SW can disable FSI_LA_CLK.

8.7.2.2.3 Debug Address Map

Detailed ROM Table Hierarchy and Components

Table 8.71 Color Coding Table for ROM Table Hierarchy (for use reading the ROMs Table below)

Color	Table Hierarchy
	Top ROM Table
	1st level nested ROM Table
	2nd level nested ROM Table
	3rd level nested ROM Table
	4th level nested ROM Table

Table 8.72 ROMs Table

APBIC ROM Table	ROM Table Entry (in HEX)	Component Type	Component Name	Component Offset Relative to Top APBIC (in HEX)	Component Address When Accessed by External Debugger (in HEX)	Component Address When Accessed by Self-hosted Software (MMIO) (in HEX)
CoreSight Major APBIC	20003	Custom	CORESIGHT_CFG	20000	80020000	24020000
	40003	Arm ATB Funnel	Major Funnel	40000	80040000	24040000
	50003	ARB ETF	ETF	50000	80050000	24050000
	60003	Arm ATB Replicator	Replicator	60000	80060000	24060000
	70003	Arm ETR	ETR	70000	80070000	24070000
	80003	Arm STM	STM	80000	80080000	24080000
	90003	Arm CTI	CTI-TOP0	90000	80090000	24090000
	A0003	Arm CTI	CTI-TOP1	A0000	800A0000	240A0000
	B0003	Arm APBIC	HSSTP Collator APBIC	B0000	800B0000	240B0000

APBIC ROM Table	ROM Table Entry (in HEX)	Component Type	Component Name	Component Offset Relative to Top APBIC (in HEX)	Component Address When Accessed by External Debugger (in HEX)	Component Address When Accessed by Self-hosted Software (MMIO) (in HEX)
	C0003			C0000	800C0000	240C0000
	400003	Arm APBIC	CoreSight Minor APBIC	400000	80400000	24400000
	800003	Arm APBIC	FSI Collator APBIC	800000	80800000	24800000
	2000003	Arm APBIC	CCPLEX Collator APBIC	2000000	82000000	26000000
HSSTP Collator APBIC	1003	Arm TPIU	TPIU	B1000	800B1000	240B1000
	2003	HSSTP	HSSTP	B2000	800B2000	240B2000
CoreSight Minor APBIC	10003	Custom	NVCDC1	410000	80410000	24410000
	20003	Arm ATB Funnel	Minor Funnel	420000	80420000	24420000
	40003	Arm APBIC	BPMP APBIC	440000	80440000	24440000
	-	-	-	-	-	-
	80003	Arm APBIC	SPE APBIC	480000	80480000	24480000
	A0003	Arm APBIC	APE APBIC	4A0000	804A0000	244A0000
	C0003	Arm APBIC	SCE APBIC	4C0000	804C0000	244C0000
	E0003	Arm APBIC	RCE APBIC	4E0000	804E0000	244E0000
	100003	Arm APBIC	DCE APBIC	500000	80500000	24500000
	200003	Arm APBIC	PVA Collator APBIC	600000	80600000	24600000
BPMP APBIC	10003	Arm Cortex-R5 Core DBG	BPMP CPU DBG	450000	80450000	24450000
	18003	Arm CTI	BPMP CTI	458000	80458000	24458000
	1C003	Arm Cortex-R5 ETM	BPMP ETM	45C000	8045C000	2445C000
-	-	-	-	-	-	-
	-	-	-	-	-	-
SPE APBIC	10003	Arm Cortex-R5 Core DBG	SPE CPU DBG	490000	80490000	24490000
	18003	Arm CTI	SPE CTI	498000	80498000	24498000

APBIC ROM Table	ROM Table Entry (in HEX)	Component Type	Component Name	Component Offset Relative to Top APBIC (in HEX)	Component Address When Accessed by External Debugger (in HEX)	Component Address When Accessed by Self-hosted Software (MMIO) (in HEX)
	1C003	Arm Cortex-R5 ETM	SPE ETM	49C000	8049C000	2449C000
APE APBIC	10003	Arm Cortex-A9 CPU DBG	APE CPU DBG	4B0000	804B0000	244B0000
	11003	Arm Cortex-A9 PMU	APE PMU	4B1000	804B1000	244B1000
	18003	Arm CTI	APE CTI	4B8000	804B8000	244B8000
	1C003	Arm Cortex-A9 PTM	APE PTM	4BC000	804BC000	244BC000
SCE APBIC	10003	Arm Cortex-R5 Core DBG	SCE CPU DBG	4D0000	804D0000	244D0000
	18003	Arm CTI	SCE CTI	4D8000	804D8000	244D8000
	1C003	Arm Cortex-R5 ETM	SCE ETM	4DC000	804DC000	244DC000
RCE APBIC	10003	Arm Cortex-R5 Core DBG	RCE CPU DBG	4F0000	804F0000	244F0000
	18003	Arm CTI	RCE CTI	4F8000	804F8000	244F8000
	1C003	Arm Cortex-R5 ETM	RCE ETM	4FC000	804FC000	244FC000
DCE APBIC	10003	Arm Cortex-R5 Core DBG	DCE CPU DBG	510000	80510000	24510000
	18003	Arm CTI	DCE CTI	518000	80518000	24518000
	1C003	Arm Cortex-R5 ETM	DCE ETM	51C000	8051C000	2451C000
PVA Collator APBIC	10003	Arm ATB Funnel	PVA Collator Funnel	610000	80610000	24610000
	100003	Arm APBIC	PVA0 APBIC	700000	80700000	24700000
PVA0 APBIC	20003	Arm APBIC	PVA0 Cortex-R5 APBIC	720000	80720000	24720000
	40003	Arm APBIC	PVA0 VPU0 APBIC	740000	80740000	24740000
	60003	Arm APBIC	PVA0 VPU1 APBIC	760000	80760000	24760000
PVA0 Cortex-R5 APBIC	10003	Arm Cortex-R5 Core DBG	PVA0 Cortex-R5 CPU DBG	730000	80730000	24730000

APBIC ROM Table	ROM Table Entry (in HEX)	Component Type	Component Name	Component Offset Relative to Top APBIC (in HEX)	Component Address When Accessed by External Debugger (in HEX)	Component Address When Accessed by Self-hosted Software (MMIO) (in HEX)
	18003	Arm CTI	PVA0 Cortex-R5 CTI	738000	80738000	24738000
	1C003	Arm Cortex-R5 ETM	PVA0 Cortex-R5 ETM	73C000	8073C000	2473C000
PVA0 VPU0 APBIC	10003	VPU Core DBG	PVA0 VPU0 Core DBG	750000	80750000	24750000
	18003	Arm CTI	PVA0 VPU0 CTI	758000	80758000	24758000
PVA0 VPU1 APBIC	10003	VPU Core DBG	PVA0 VPU1 Core DBG	770000	80770000	24770000
	18003	Arm CTI	PVA0 VPU1 CTI	778000	80778000	24778000
FSI Collator APBIC	10003	Arm ATB Funnel	FSI Collator Funnel	810000	80810000	24810000
	20003	Arm ATB Funnel	FSI Cluster-1 Funnel	820000	80820000	24820000
	30003	Arm CTI	FSI TOP CTI	830000	80830000	24830000
	40003	Arm APBIC	FSI CLuster0 APBIC	840000	80840000	24840000
	400003	Arm APBIC	FSI Cluster1 APBIC	C00000	80C00000	24C00000
FSI Cluster0 APBIC	10003	Arm Cortex-R5 Core DBG	FSI Cluster0 Core0 CPU DBG	850000	80850000	24850000
	18003	Arm CTI	FSI Cluster0 Core0 CTI	858000	80858000	24858000
	1C003	Arm Cortex-R5 ETM	FSI Cluster0 Core0 ETM	85C000	8085C000	2485C000
FSI Cluster1 APBIC	10003	Arm R52 Core DBG	FSI Cluster1 Core0 CPU DBG	C10000	80C10000	24C10000
	20003	Arm CTI	FSI Cluster1 Core0 CTI	C20000	80C20000	24C20000
	30003	Arm R52 PMU	FSI Cluster1 Core0 PMU	C30000	80C30000	24C30000
	40003	Arm R52 ETM	FSI Cluster1 Core0 ETM	C40000	80C40000	24C40000

APBIC ROM Table	ROM Table Entry (in HEX)	Component Type	Component Name	Component Offset Relative to Top APBIC (in HEX)	Component Address When Accessed by External Debugger (in HEX)	Component Address When Accessed by Self-hosted Software (MMIO) (in HEX)
	110003	Arm R52 Core DBG	FSI Cluster1 Core1 CPU DBG	D10000	80D10000	24D10000
	120003	Arm CTI	FSI Cluster1 Core1 CTI	D20000	80D20000	24D20000
	130003	Arm R52 PMU	FSI Cluster1 Core1 PMU	D30000	80D30000	24D30000
	140003	Arm R52 ETM	FSI Cluster1 Core1 ETM	D40000	80D40000	24D40000
	210003	Arm R52 Core DBG	FSI Cluster1 Core2 CPU DBG	E10000	80E10000	24E10000
	220003	Arm CTI	FSI Cluster1 Core2 CTI	E20000	80E20000	24E20000
	230003	Arm R52 PMU	FSI Cluster1 Core2 PMU	E30000	80E30000	24E30000
	240003	Arm R52 ETM	FSI Cluster1 Core2 ETM	E40000	80E40000	24E40000
	310003	Arm R52 Core DBG	FSI Cluster1 Core3 CPU DBG	F10000	80F10000	24F10000
	320003	Arm CTI	FSI Cluster1 Core3 CTI	F20000	80F20000	24F20000
	330003	Arm R52 PMU	FSI Cluster1 Core3 PMU	F30000	80F30000	24F30000
	340003	Arm R52 ETM	FSI Cluster1 Core3 ETM	F40000	80F40000	24F40000
CCPLEX Collator APBIC	10003	Custom	NVCDC2	2010000	82010000	26010000
	20003	Arm ATB Funnel	CCPLEX TOP ATB Funnel	2020000	82020000	26020000
	30003	Arm ATB Funnel	CCPLEX Cluster 0 ATB Funnel	2030000	82030000	26030000
	40003	Arm ATB Funnel	CCPLEX Cluster 1 ATB Funnel	2040000	82040000	26040000
	50003	Arm ATB Funnel	CCPLEX Cluster 2 ATB Funnel	2050000	82050000	26050000

APBIC ROM Table	ROM Table Entry (in HEX)	Component Type	Component Name	Component Offset Relative to Top APBIC (in HEX)	Component Address When Accessed by External Debugger (in HEX)	Component Address When Accessed by Self-hosted Software (MMIO) (in HEX)
	60003	Arm ATB Funnel	CCPLEX Cluster 3 ATB Funnel	2060000	82060000	26060000
	70003	Arm Uncore Perfmon	SCF Perfmon	2070000	82070000	26070000
	80003	Arm CTI	CCPLEX CTI	2080000	82080000	26080000
	1000003	Arm APBIC	CCPLEX Cluster0 APBIC	3000000	83000000	27000000
	1400003	Arm APBIC	CCPLEX Cluster1 APBIC	3400000	83400000	27400000
	1800003	Arm APBIC	CCPLEX Cluster2 APBIC	3800000	83800000	27800000
	1C00003	Arm APBIC	CCPLEX Cluster3 APBIC	3C00000	83C00000	27C00000
CCPLEX Cluster0 APBIC	10003	Arm PE DBG	CCPLEX Cluster0 Core0 DBG	3010000	83010000	27010000
	20003	Arm PE CTI	CCPLEX Cluster0 Core0 CTI	3020000	83020000	27020000
	30003	Arm PE PMU	CCPLEX Cluster0 Core0 PMU	3030000	83030000	27030000
	40003	Arm PE ETM	CCPLEX Cluster0 Core0 ETM	3040000	83040000	27040000
	C0003	Arm PE ELA	CCPLEX Cluster0 Core0 ELA	30C0000	830C0000	270C0000
	D0003	Arm Cluster ELA	CCPLEX Cluster0 ELA	30D0000	830D0000	270D0000
	E0003	Arm Cluster CTI	CCPLEX Cluster0 CTI	30E0000	830E0000	270E0000
	F0003	Arm PE ACTMON	CCPLEX Cluster0 Core0 ACTMON	30F0000	830F0000	270F0000

APBIC ROM Table	ROM Table Entry (in HEX)	Component Type	Component Name	Component Offset Relative to Top APBIC (in HEX)	Component Address When Accessed by External Debugger (in HEX)	Component Address When Accessed by Self-hosted Software (MMIO) (in HEX)
	110003	Arm PE DBG	CCPLEX Cluster0 Core1 DBG	3110000	83110000	27110000
	120003	Arm PE CTI	CCPLEX Cluster0 Core1 CTI	3120000	83120000	27120000
	130003	Arm PE PMU	CCPLEX Cluster0 Core1 PMU	3130000	83130000	27130000
	140003	Arm PE ETM	CCPLEX Cluster0 Core1 ETM	3140000	83140000	27140000
	1C0003	Arm PE ELA	CCPLEX Cluster0 Core1 ELA	31C0000	831C0000	271C0000
	1D0003	Arm Cluster ELA	CCPLEX Cluster0 ELA	31D0000	831D0000	271D0000
	1E0003	Arm Cluster CTI	CCPLEX Cluster0 CTI	31E0000	831E0000	271E0000
	1F0003	Arm PE ACTMON	CCPLEX Cluster0 Core1 ACTMON	31F0000	831F0000	271F0000
	210003	Arm PE DBG	CCPLEX Cluster0 Core2 DBG	3210000	83210000	27210000
	220003	Arm PE CTI	CCPLEX Cluster0 Core2 CTI	3220000	83220000	27220000
	230003	Arm PE PMU	CCPLEX Cluster0 Core2 PMU	3230000	83230000	27230000
	240003	Arm PE ETM	CCPLEX Cluster0 Core2 ETM	3240000	83240000	27240000
	2C0003	Arm PE ELA	CCPLEX Cluster0 Core2 ELA	32C0000	832C0000	272C0000
	2D0003	Arm Cluster ELA	CCPLEX Cluster0 ELA	32D0000	832D0000	272D0000
	2E0003	Arm Cluster CTI	CCPLEX Cluster0 CTI	32E0000	832E0000	272E0000

APBIC ROM Table	ROM Table Entry (in HEX)	Component Type	Component Name	Component Offset Relative to Top APBIC (in HEX)	Component Address When Accessed by External Debugger (in HEX)	Component Address When Accessed by Self-hosted Software (MMIO) (in HEX)
	2F0003	Arm PE ACTMON	CCPLEX Cluster0 Core2 ACTMON	32F0000	832F0000	272F0000
	310003	Arm PE DBG	CCPLEX Cluster0 Core3 DBG	3310000	83310000	27310000
	320003	Arm PE CTI	CCPLEX Cluster0 Core3 CTI	3320000	83320000	27320000
	330003	Arm PE PMU	CCPLEX Cluster0 Core3 PMU	3330000	83330000	27330000
	340003	Arm PE ETM	CCPLEX Cluster0 Core3 ETM	3340000	83340000	27340000
	3C0003	Arm PE ELA	CCPLEX Cluster0 Core3 ELA	33C0000	833C0000	273C0000
	3D0003	Arm Cluster ELA	CCPLEX Cluster0 ELA	33D0000	833D0000	273D0000
	3E0003	Arm Cluster CTI	CCPLEX Cluster0 CTI	33E0000	833E0000	273E0000
	3F0003	Arm PE ACTMON	CCPLEX Cluster0 Core3 ACTMON	33F0000	833F0000	273F0000
CCPLEX Cluster1 APBIC	10003	Arm PE DBG	CCPLEX Cluster1 Core0 DBG	3410000	83410000	27410000
	20003	Arm PE CTI	CCPLEX Cluster1 Core0 CTI	3420000	83420000	27420000
	30003	Arm PE PMU	CCPLEX Cluster1 Core0 PMU	3430000	83430000	27430000
	40003	Arm PE ETM	CCPLEX Cluster1 Core0 ETM	3440000	83440000	27440000
	C0003	Arm PE ELA	CCPLEX Cluster1 Core0 ELA	34C0000	834C0000	274C0000

APBIC ROM Table	ROM Table Entry (in HEX)	Component Type	Component Name	Component Offset Relative to Top APBIC (in HEX)	Component Address When Accessed by External Debugger (in HEX)	Component Address When Accessed by Self-hosted Software (MMIO) (in HEX)
	D0003	Arm Cluster ELA	CCPLEX Cluster1 ELA	34D0000	834D0000	274D0000
	E0003	Arm Cluster CTI	CCPLEX Cluster1 CTI	34E0000	834E0000	274E0000
	F0003	Arm PE ACTMON	CCPLEX Cluster1 Core0 ACTMON	34F0000	834F0000	274F0000
	110003	Arm PE DBG	CCPLEX Cluster1 Core1 DBG	3510000	83510000	27510000
	120003	Arm PE CTI	CCPLEX Cluster1 Core1 CTI	3520000	83520000	27520000
	130003	Arm PE PMU	CCPLEX Cluster1 Core1 PMU	3530000	83530000	27530000
	140003	Arm PE ETM	CCPLEX Cluster1 Core1 ETM	3540000	83540000	27540000
	1C0003	Arm PE ELA	CCPLEX Cluster1 Core1 ELA	35C0000	835C0000	275C0000
	1D0003	Arm Cluster ELA	CCPLEX Cluster1 ELA	35D0000	835D0000	275D0000
	1E0003	Arm Cluster CTI	CCPLEX Cluster1 CTI	35E0000	835E0000	275E0000
	1F0003	Arm PE ACTMON	CCPLEX Cluster1 Core1 ACTMON	35F0000	835F0000	275F0000
	210003	Arm PE DBG	CCPLEX Cluster1 Core2 DBG	3610000	83610000	27610000
	220003	Arm PE CTI	CCPLEX Cluster1 Core2 CTI	3620000	83620000	27620000
	230003	Arm PE PMU	CCPLEX Cluster1 Core2 PMU	3630000	83630000	27630000

APBIC ROM Table	ROM Table Entry (in HEX)	Component Type	Component Name	Component Offset Relative to Top APBIC (in HEX)	Component Address When Accessed by External Debugger (in HEX)	Component Address When Accessed by Self-hosted Software (MMIO) (in HEX)
	240003	Arm PE ETM	CCPLEX Cluster1 Core2 ETM	3640000	83640000	27640000
	2C0003	Arm PE ELA	CCPLEX Cluster1 Core2 ELA	36C0000	836C0000	276C0000
	2D0003	Arm Cluster ELA	CCPLEX Cluster1 ELA	36D0000	836D0000	276D0000
	2E0003	Arm Cluster CTI	CCPLEX Cluster1 CTI	36E0000	836E0000	276E0000
	2F0003	Arm PE ACTMON	CCPLEX Cluster1 Core2 ACTMON	36F0000	836F0000	276F0000
	310003	Arm PE DBG	CCPLEX Cluster1 Core3 DBG	3710000	83710000	27710000
	320003	Arm PE CTI	CCPLEX Cluster1 Core3 CTI	3720000	83720000	27720000
	330003	Arm PE PMU	CCPLEX Cluster1 Core3 PMU	3730000	83730000	27730000
	340003	Arm PE ETM	CCPLEX Cluster1 Core3 ETM	3740000	83740000	27740000
	3C0003	Arm PE ELA	CCPLEX Cluster1 Core3 ELA	37C0000	837C0000	277C0000
	3D0003	Arm Cluster ELA	CCPLEX Cluster1 ELA	37D0000	837D0000	277D0000
	3E0003	Arm Cluster CTI	CCPLEX Cluster1 CTI	37E0000	837E0000	277E0000
	3F0003	Arm PE ACTMON	CCPLEX Cluster1 Core3 ACTMON	37F0000	837F0000	277F0000
CCPLEX Cluster2 APBIC	10003	Arm PE DBG	CCPLEX Cluster2 Core0 DBG	3810000	83810000	27810000

APBIC ROM Table	ROM Table Entry (in HEX)	Component Type	Component Name	Component Offset Relative to Top APBIC (in HEX)	Component Address When Accessed by External Debugger (in HEX)	Component Address When Accessed by Self-hosted Software (MMIO) (in HEX)
	20003	Arm PE CTI	CCPLEX Cluster2 Core0 CTI	3820000	83820000	27820000
	30003	Arm PE PMU	CCPLEX Cluster2 Core0 PMU	3830000	83830000	27830000
	40003	Arm PE ETM	CCPLEX Cluster2 Core0 ETM	3840000	83840000	27840000
	C0003	Arm PE ELA	CCPLEX Cluster2 Core0 ELA	38C0000	838C0000	278C0000
	D0003	Arm Cluster ELA	CCPLEX Cluster2 ELA	38D0000	838D0000	278D0000
	E0003	Arm Cluster CTI	CCPLEX Cluster2 CTI	38E0000	838E0000	278E0000
	F0003	Arm PE ACTMON	CCPLEX Cluster2 Core0 ACTMON	38F0000	838F0000	278F0000
	110003	Arm PE DBG	CCPLEX Cluster2 Core1 DBG	3910000	83910000	27910000
	120003	Arm PE CTI	CCPLEX Cluster2 Core1 CTI	3920000	83920000	27920000
	130003	Arm PE PMU	CCPLEX Cluster2 Core1 PMU	3930000	83930000	27930000
	140003	Arm PE ETM	CCPLEX Cluster2 Core1 ETM	3940000	83940000	27940000
	1C0003	Arm PE ELA	CCPLEX Cluster2 Core1 ELA	39C0000	839C0000	279C0000
	1D0003	Arm Cluster ELA	CCPLEX Cluster2 ELA	39D0000	839D0000	279D0000
	1E0003	Arm Cluster CTI	CCPLEX Cluster2 CTI	39E0000	839E0000	279E0000

APBIC ROM Table	ROM Table Entry (in HEX)	Component Type	Component Name	Component Offset Relative to Top APBIC (in HEX)	Component Address When Accessed by External Debugger (in HEX)	Component Address When Accessed by Self-hosted Software (MMIO) (in HEX)
	1F0003	Arm PE ACTMON	CCPLEX Cluster2 Core1 ACTMON	39F0000	839F0000	279F0000
	210003	Arm PE DBG	CCPLEX Cluster2 Core2 DBG	3A10000	83A10000	27A10000
	220003	Arm PE CTI	CCPLEX Cluster2 Core2 CTI	3A20000	83A20000	27A20000
	230003	Arm PE PMU	CCPLEX Cluster2 Core2 PMU	3A30000	83A30000	27A30000
	240003	Arm PE ETM	CCPLEX Cluster2 Core2 ETM	3A40000	83A40000	27A40000
	2C0003	Arm PE ELA	CCPLEX Cluster2 Core2 ELA	3AC0000	83AC0000	27AC0000
	2D0003	Arm Cluster ELA	CCPLEX Cluster2 ELA	3AD0000	83AD0000	27AD0000
	2E0003	Arm Cluster CTI	CCPLEX Cluster2 CTI	3AE0000	83AE0000	27AE0000
	2F0003	Arm PE ACTMON	CCPLEX Cluster2 Core2 ACTMON	3AF0000	83AF0000	27AF0000
	310003	Arm PE DBG	CCPLEX Cluster2 Core3 DBG	3B10000	83B10000	27B10000
	320003	Arm PE CTI	CCPLEX Cluster2 Core3 CTI	3B20000	83B20000	27B20000
	330003	Arm PE PMU	CCPLEX Cluster2 Core3 PMU	3B30000	83B30000	27B30000
	340003	Arm PE ETM	CCPLEX Cluster2 Core3 ETM	3B40000	83B40000	27B40000
	3C0003	Arm PE ELA	CCPLEX Cluster2 Core3 ELA	3BC0000	83BC0000	27BC0000

APBIC ROM Table	ROM Table Entry (in HEX)	Component Type	Component Name	Component Offset Relative to Top APBIC (in HEX)	Component Address When Accessed by External Debugger (in HEX)	Component Address When Accessed by Self-hosted Software (MMIO) (in HEX)
	3D0003	Arm Cluster ELA	CCPLEX Cluster2 ELA	3BD0000	83BD0000	27BD0000
	3E0003	Arm Cluster CTI	CCPLEX Cluster2 CTI	3BE0000	83BE0000	27BE0000
	3F0003	Arm PE ACTMON	CCPLEX Cluster2 Core3 ACTMON	3BF0000	83BF0000	27BF0000
CCPLEX Cluster3 APBIC	10003	Arm PE DBG	CCPLEX Cluster3 Core0 DBG	3C10000	83C10000	27C10000
	20003	Arm PE CTI	CCPLEX Cluster3 Core0 CTI	3C20000	83C20000	27C20000
	30003	Arm PE PMU	CCPLEX Cluster3 Core0 PMU	3C30000	83C30000	27C30000
	40003	Arm PE ETM	CCPLEX Cluster3 Core0 ETM	3C40000	83C40000	27C40000
	C0003	Arm PE ELA	CCPLEX Cluster3 Core0 ELA	3CC0000	83CC0000	27CC0000
	D0003	Arm Cluster ELA	CCPLEX Cluster3 ELA	3CD0000	83CD0000	27CD0000
	E0003	Arm Cluster CTI	CCPLEX Cluster3 CTI	3CE0000	83CE0000	27CE0000
	F0003	Arm PE ACTMON	CCPLEX Cluster3 Core0 ACTMON	3CF0000	83CF0000	27CF0000
	110003	Arm PE DBG	CCPLEX Cluster3 Core1 DBG	3D10000	83D10000	27D10000
	120003	Arm PE CTI	CCPLEX Cluster3 Core1 CTI	3D20000	83D20000	27D20000
	130003	Arm PE PMU	CCPLEX Cluster3 Core1 PMU	3D30000	83D30000	27D30000

APBIC ROM Table	ROM Table Entry (in HEX)	Component Type	Component Name	Component Offset Relative to Top APBIC (in HEX)	Component Address When Accessed by External Debugger (in HEX)	Component Address When Accessed by Self-hosted Software (MMIO) (in HEX)
	140003	Arm PE ETM	CCPLEX Cluster3 Core1 ETM	3D40000	83D40000	27D40000
	1C0003	Arm PE ELA	CCPLEX Cluster3 Core1 ELA	3DC0000	83DC0000	27DC0000
	1D0003	Arm Cluster ELA	CCPLEX Cluster3 ELA	3DD0000	83DD0000	27DD0000
	1E0003	Arm Cluster CTI	CCPLEX Cluster3 CTI	3DE0000	83DE0000	27DE0000
	1F0003	Arm PE ACTMON	CCPLEX Cluster3 Core1 ACTMON	3DF0000	83DF0000	27DF0000
	210003	Arm PE DBG	CCPLEX Cluster3 Core2 DBG	3E10000	83E10000	27E10000
	220003	Arm PE CTI	CCPLEX Cluster3 Core2 CTI	3E20000	83E20000	27E20000
	230003	Arm PE PMU	CCPLEX Cluster3 Core2 PMU	3E30000	83E30000	27E30000
	240003	Arm PE ETM	CCPLEX Cluster3 Core2 ETM	3E40000	83E40000	27E40000
	2C0003	Arm PE ELA	CCPLEX Cluster3 Core2 ELA	3EC0000	83EC0000	27EC0000
	2D0003	Arm Cluster ELA	CCPLEX Cluster3 ELA	3ED0000	83ED0000	27ED0000
	2E0003	Arm Cluster CTI	CCPLEX Cluster3 CTI	3EE0000	83EE0000	27EE0000
	2F0003	Arm PE ACTMON	CCPLEX Cluster3 Core2 ACTMON	3EF0000	83EF0000	27EF0000
	310003	Arm PE DBG	CCPLEX Cluster3 Core3 DBG	3F10000	83F10000	27F10000

APBIC ROM Table	ROM Table Entry (in HEX)	Component Type	Component Name	Component Offset Relative to Top APBIC (in HEX)	Component Address When Accessed by External Debugger (in HEX)	Component Address When Accessed by Self-hosted Software (MMIO) (in HEX)
	320003	Arm PE CTI	CCPLEX Cluster3 Core3 CTI	3F20000	83F20000	27F20000
	330003	Arm PE PMU	CCPLEX Cluster3 Core3 PMU	3F30000	83F30000	27F30000
	340003	Arm PE ETM	CCPLEX Cluster3 Core3 ETM	3F40000	83F40000	27F40000
	3C0003	Arm PE ELA	CCPLEX Cluster3 Core3 ELA	3FC0000	83FC0000	27FC0000
	3D0003	Arm Cluster ELA	CCPLEX Cluster3 ELA	3FD0000	83FD0000	27FD0000
	3E0003	Arm Cluster CTI	CCPLEX Cluster3 CTI	3FE0000	83FE0000	27FE0000
	3F0003	Arm PE ACTMON	CCPLEX Cluster3 Core3 ACTMON	3FF0000	83FF0000	27FF0000

[1] CoreSight Major APBIC is the Top APBIC. All APBICs include a ROM Table in them.

[2] STM's AXI interface is accessible by self-hosted software at 16 MiB MMIO address range: [0x25000000 - 0x25FFFFFF]

8.7.2.2.4 Interfaces

Table 8.73 System Interconnect Interfaces

IP	Data width (bits)	Addr Width (bits)	Protocol	Clock	Direction	Connected to
STM	32	32	AXI3	CSITE	Target	CBB
STM	32	-	ATB3	CSITE	Initiator	32:128 Upsizer
STM	32	32	dAPB	CSITE	Target	CoreSight Major APBIC
ETR	128	40	AXI-DBB (AXI3)	CSITE	Initiator	DBB
ETR	128	-	ATB3	CSITE	Target	Replicator

IP	Data width (bits)	Addr Width (bits)	Protocol	Clock	Direction	Connected to
ETR	32	32	dAPB	CSITE	Target	CoreSight Major APBIC
HSSTP	20	-	8b10b	PEX-UPHY's L0_TX_CLK	Initiator	PEX-UPHY Lane-0 mux
HSSTP	20	-	8b10b	PEX-UPHY's L1_TX_CLK	Initiator	PEX-UPHY Lane-1 mux
HSSTP	20	-	8b10b	PEX-UPHY's L4_TX_CLK	Initiator	PEX-UPHY Lane-4 mux
HSSTP	20	-	8b10b	PEX-UPHY's L5_TX_CLK	Initiator	PEX-UPHY Lane-5 mux
HSSTP	32	-	ATB4	CSITE	Target	Replicator (the signal passed through 128:32 downsizer before connecting to HSSTP)
HSSTP	32	32	dAPB	CSITE	Target	CoreSight Major APBIC
CTI0 (in DFD partition)	32	32	dAPB	DBGAPB	Target	CoreSight Major APBIC
CTI1 (in DFD partition)	32	32	dAPB	DBGAPB	Target	CoreSight Major APBIC
CTI0		16	events	CSITE	in/out	ETF/ETR/STM/etc.
CTI1		16	events	CSITE	in/out	ETF/ETR/STM/etc.
AXI-AP	64	40	ACE-Lite	DBGAPB	Initiator	
DFD 1x2 NIC	64	40	AXI-CBB(AXI4)	DBGAPB	Initiator	CBB
DFD 1x2 NIC	64	40	AXI-DBB(AXI4)	DBGAPB	Initiator	DBB
CoreSight Major APBIC	32	32	APB3	DBGAPB	Target	CBB
CoreSight Major APBIC	32	32	APB3	DBGAPB	Target	APB-AP
ArmJTAG		1-TDI1-TDO1-TMS1-TCK	4-pin JTAG	TCK	Initiator	DAP (see *Closed Box Debug over USB2* section for details)
SWD		1-SWDIO1-SWCLK	2-pin SWD	SWCLK	Initiator	DAP (see *Closed Box Debug over USB2* section for details)

IP	Data width (bits)	Addr Width (bits)	Protocol	Clock	Direction	Connected to
BPMP/ PVA Collator/SPE/ SCE/ RCE/DCE/ FSI Collator/ CCPLEX Collator		16 (per interface) as below:4 chout4 choutack4 chin4 chinack	CTMevent+ack	DBGAPB		DFD
APE		16 (per interface) as below:4 chout4 choutack4 chin4 chinack	CTMevent+ack	CSITE		DFD
APE/BPMP/ SPE/SCE/ RCE/DCE	32 (per interface)		ATB4	CSITE	Initiator	DFD
PVA Collator	64		ATB4	CSITE	Initiator	DFD
FSI Collator/CCPLEX Collator	128		ATB4	CSITE	Initiator	DFD
BPMP/PVA Collator/ SPE/SCE/RCE/DCE	32 (per interface)	32 (per interface)	dAPB (APB3)	DBGAPB	Target	CoreSight Minor APBIC
APE	32 (per interface)	32 (per interface)	dAPB (APB3)	CSITE	Target	CoreSight Minor APBIC
CCPLEX Collator / FSI Collator	32	32	dAPB (APB3)	DBGAPB	Target	CoreSight Major APBIC

Table 8.74 External I/O Interfaces

Interface	Pins	Protocol	Pins over which muxed	Description
Dedicated JTAG	(TMS, TCK, TDI, TDO)	4-pin JTAG	-	ArmJTAG
UART3	RX, TX	2-pin UART	-	Debug AON UART (also used by boot ROM)
UART6	RX, TX	2-pin UART	DPAUX0 (I2C – IOP/ ION)	Closed Box Debug over DPAUX
UART8	RX, TX	2-pin UART	USB2 (D+/D-)	Closed Box Debug over USB2
PEX-UPHY Lane-0 (HSIO UPHY Brick)	TX+/TX-	HSSTP	USB3.1 Port-0 (TX+/ TX-)	HSSTP

Interface	Pins	Protocol	Pins over which muxed	Description
PEX-UPHY Lane-1 (HSIO UPHY Brick)	TX+/TX-	HSSTP	USB3.1 Port-1 (TX+/TX-)	HSSTP
PEX-UPHY Lane-4 (HSIO UPHY Brick)	TX+/TX-	HSSTP	SD7.0/UFS (TX+/TX-)	HSSTP
PEX-UPHY Lane-5 (HSIO UPHY Brick)	TX+/TX-	HSSTP	NVMe Cartridge (TX+/TX-)	HSSTP
SWD	SWCLK, SWDIO	SWD	USB2 (D+/D-)	Closed Box Debug over USB2
NVJTAG_SEL	NVJTAG_SEL	-	-	NVJTAG over Dedicated JTAG interface
NVDBG_SEL	NVDBG_SEL	-	-	NVJTAG over USB2/SWD interface
CCLA_LA_TRIGGER_MUX	GPIO	-	-	Trigger output from CCLA for external Benchtop Logic Analyzer. (Requires GPIO to route the trigger)

For details of the debugging environment settings, refer to the *Selection Matrix in the USB2 and Dedicated JTAG Ports* section.

8.7.2.2.5 Cross Trigger Events

The Cross Trigger Interface allows collection of events from various components and facilitates communication using these events among the participants using Cross Trigger Matrix. The following table lists connectivity of various CTI events in the system.

Table 8.75 CTI0 Mappings in CoreSight

CTI Trigger	Signal Name of Connected Module	Connected Module	Comments
CTITRIGIN0	FULL	ETF	
CTITRIGIN1	ACQCOMP	ETF	
CTITRIGIN2	FULL	ETR	
CTITRIGIN3	ACQCOMP	ETR	
CTITRIGIN4	ASYNCOUT	STM	
CTITRIGIN5	TRIGOUTSPTE	STM	
CTITRIGIN6	TRIGOUTSW	STM	
CTITRIGIN7	-	-	
CTITRIGOUT0	TRIGIN	ETF	

CTI Trigger	Signal Name of Connected Module	Connected Module	Comments
CTITRIGOUT1	FLUSHIN	ETF	
CTITRIGOUT2	TRIGIN	ETR	
CTITRIGOUT3	FLUSHIN	ETR	
CTITRIGOUT4	TRIGIN	TPIU (in HSSTP)	
CTITRIGOUT5	FLUSHIN	TPIU (in HSSTP)	
CTITRIGOUT6	DLY_TRIG0	HSSTP	CTITRIGOUTACK is driven by non-delayed version of corresponding CTITRIGOUT.
CTITRIGOUT7	-	-	

Notes:

- Unused CTITRIGINACK[7:0] outputs from CTI are left unconnected.
- Unused CTITRIGOUTACK[7:0] inputs to CTI are tied to '1.'

Table 8.76 CTI1 Mappings in CoreSight

CTI Trigger	Signal Name of Connected Module	Connected Module	Comments
CTITRIGIN0	pmc2dfd_CSYSREQ_inv	PMC	
CTITRIGIN1	NVFFCR[SW_TRIG_OUT]	HSSTP	
CTITRIGIN2	CORESIGHT_CFG_HSSTP_INTERRUPT[INT_0]	DFD (CSITE-CFG)	ACK is not connected. Software or debugger is expected to clear the interrupt source.
CTITRIGIN3	(SOC-HWPM's pmm2dfd_trigger output)	SOC-HWPM, (i)GPU-HWPM	Chaining SOC-DFD, SOC-HWPM and (i)GPU-HWPM trigger networks. CTITRIGINACK is left unconnected. The source is expected to keep the trigger asserted for a long time and it is acceptable if a trigger is missed due to its existence for a very short duration.
CTITRIGIN4-7	[private]	NVCDC1	
CTITRIGOUT0	ccplex_timer_freeze_trig	CCPLEX Timer Freeze/ Unfreeze Request Generator in CoreSight Major	Used to generate timer freeze request. CTITRIGOUTACK0 is driven from Timer Freeze/Unfreeze Request Generator logic in CoreSight Major
CTITRIGOUT1	ccplex_timer_unfreeze_trig	CCPLEX Timer Freeze/ Unfreeze Request Generator in CoreSight Major	Used to generate timer unfreeze request. CTITRIGOUTACK1 is driven from Timer Freeze/Unfreeze Request Generator logic in CoreSight Major

CTI Trigger	Signal Name of Connected Module	Connected Module	Comments
CTITRIGOUT2	DLY_TRIG1	HSSTP	CTITRIGOUTACK is driven by non-delayed version of corresponding CTITRIGOUT.
CTITRIGOUT3	(SOC-HWPM's pmm2dfd_trigger output) (CTI trigout from DFD)	SOC-HWPM, (i)GPU-HWPM	Chaining SOC-DFD, SOC-HWPM and (i)GPU-HWPM trigger networks. CTITRIGOUTACK is tied to '0'.
CTITRIGOUT4-7	[private]	NVCDC1	

CTI Mappings in Cortex-R5 CoreSight PILs

Applicable for BPMP, SPE, SCE, RCE, DCE, and PVA0's Cortex-R5 clusters.

The following table provides information, but for more details refer to the Cortex-R5 CoreSight PIL specification.

Table 8.77 CTI Mappings in Cortex-R5 CoreSight

CTI Trigger	Signal Name of Connected Module	Connected Module	Comments
CTITRIGIN0	DBGTRIGGER	Cortex-R5	PIL logic ensures CTITRIGIN0 can be asserted by DBGTRIGGER only if CTITRIGINACK0 is LOW. CTITRIGIN0 held using PIL-logic (even if source deasserts trigger) until CTITRIGINACK0 is asserted by CTI.
CTITRIGIN1	nPMUIRQ	Cortex-R5's PMU	Trigger input to CTI held using PIL-logic (even if source deasserts trigger) until corresponding CTITRIGINACK is asserted by CTI
CTITRIGIN2	ETMEXTOUT[0]	ETM in Cortex-R5 CoreSight PIL	(same as above)
CTITRIGIN3	ETMEXTOUT[1]	ETM in Cortex-R5 CoreSight PIL	(same as above)
CTITRIGIN4	COMMRX	Cortex-R5	(same as above)
CTITRIGIN5	COMMTX	Cortex-R5	(same as above)
CTITRIGIN6	ETMTRIGGER	ETM in Cortex-R5 CoreSight PIL	CTITRIGINACK6 connected to ETM's TRIGGERACK by PIL-logic
CTITRIGIN7	0	Tied-off	CTITRIGINACK7 is left unconnected

CTI Trigger	Signal Name of Connected Module	Connected Module	Comments
CTITRIGOUT0	EDBGRQ	Cortex-R5	CTITRIGOUT0 is be OR-ed with EDBGRQ_<P> signal (corresponding to the processor/cluster) from DFD. The OR-ed output is then connected to EDBGRQ input of the processor. CTITRIGOUTACK0 is tied to '0' and CTI relies of clearing CTITRIGOUT0 via Software ACK
CTITRIGOUT1	ETMEXTIN[0]	ETM in Cortex-R5 CoreSight PIL	CTITRIGOUTACK1 is driven by CTITRIGOUT1
CTITRIGOUT2	ETMEXTIN[1]	ETM in Cortex-R5 CoreSight PIL	CTITRIGOUTACK2 is driven by CTITRIGOUT2
CTITRIGOUT3	nCTIIRQ	VIC in Cortex-R5 Cluster	Interrupt from Cortex-R5 PIL connected to local VIC in Cortex-R5 cluster. CTITRIGOUTACK3 is tied to '0'
CTITRIGOUT4	-	Not connected	CTITRIGOUTACK4 is tied to '0'
CTITRIGOUT5	-	Not connected	CTITRIGOUTACK5 is tied to '0'
CTITRIGOUT6	-	Not connected	CTITRIGOUTACK6 is tied to '0'
CTITRIGOUT7	DBGRESTART	Cortex-R5	CTITRIGOUTACK7 is driven from DBGRESTARTED signal from Cortex-R5

CTI Mappings in R52 Cluster

The following connections apply for each R52 core and its associated CTI.

Table 8.78 CTI Mappings in R52 Cluster

CTI Trigger	Signal Name of Connected Module	Connected Module	Comments
CTITRIGIN0	DBGTRIGGER	R52 Core Debug	CTI operates on the same clock as R52 core, PMU and ETM. Hence, triggers can be captured in single clock cycle (CTITRIGINACKs and CTITRIGOUTACKs are not documented here).
CTITRIGIN1	nPMUIRQ	PMU	(same as above)
CTITRIGIN2			(same as above)
CTITRIGIN3			(same as above)
CTITRIGIN4	ETM trace unit external output 0	ETM	(same as above)
CTITRIGIN5	ETM trace unit external output 1	ETM	(same as above)

CTI Trigger	Signal Name of Connected Module	Connected Module	Comments
CTITRIGIN6	ETM trace unit external output 2	ETM	(same as above)
CTITRIGIN7	ETM trace unit external output 3	ETM	(same as above)
CTITRIGOUT0	Requests the processor to enter debug state	R52 Core Debug	(same as above)
CTITRIGOUT1	Requests the processor to exit debug state	R52 Core Debug	(same as above)
CTITRIGOUT2	CTI interrupt	GIC Distributor Unit in Governor	(same as above)
CTITRIGOUT3	-	-	(same as above)
CTITRIGOUT4	ETM trace unit external input 0	ETM	(same as above)
CTITRIGOUT5	ETM trace unit external input 1	ETM	(same as above)
CTITRIGOUT6	ETM trace unit external input 2	ETM	(same as above)
CTITRIGOUT7	ETM trace unit external input 3	ETM	(same as above)

FSI Top CTI Mappings

Table 8.79 FSI Top CTI Mappings

CTI Trigger	Signal Name of Connected Module	Connected Module	Comments
CTITRIGIN0	WDT 3rd expiration (also known as remote interrupt)	FSI-TKE-WDT	FSI-TKE contains five WDTs but exports only one 3rd expiration common across all WDTs. CTITRIGINACK is not connected (TKE does not expect an ACK for 3rd expiration)
CTITRIGIN1	0	Tied-off	CTITRIGINACK is not connected
CTITRIGIN2	0	Tied-off	CTITRIGINACK is not connected
CTITRIGIN3	0	Tied-off	CTITRIGINACK is not connected
CTITRIGIN4	0	Tied-off	CTITRIGINACK is not connected
CTITRIGIN5	0	Tied-off	CTITRIGINACK is not connected
CTITRIGIN6	0	Tied-off	CTITRIGINACK is not connected
CTITRIGIN7	0	Tied-off	CTITRIGINACK is not connected

CTI Trigger	Signal Name of Connected Module	Connected Module	Comments
CTITRIGOUT0	-	Not connected	CTITRIGOUTACK is tied to '0'
CTITRIGOUT1	-	Not connected	CTITRIGOUTACK is tied to '0'
CTITRIGOUT2	-	Not connected	CTITRIGOUTACK is tied to '0'
CTITRIGOUT3	-	Not connected	CTITRIGOUTACK is tied to '0'
CTITRIGOUT4	-	Not connected	CTITRIGOUTACK is tied to '0'
CTITRIGOUT5	-	Not connected	CTITRIGOUTACK is tied to '0'
CTITRIGOUT6	-	Not connected	CTITRIGOUTACK is tied to '0'
CTITRIGOUT7	-	Not connected	CTITRIGOUTACK is tied to '0'

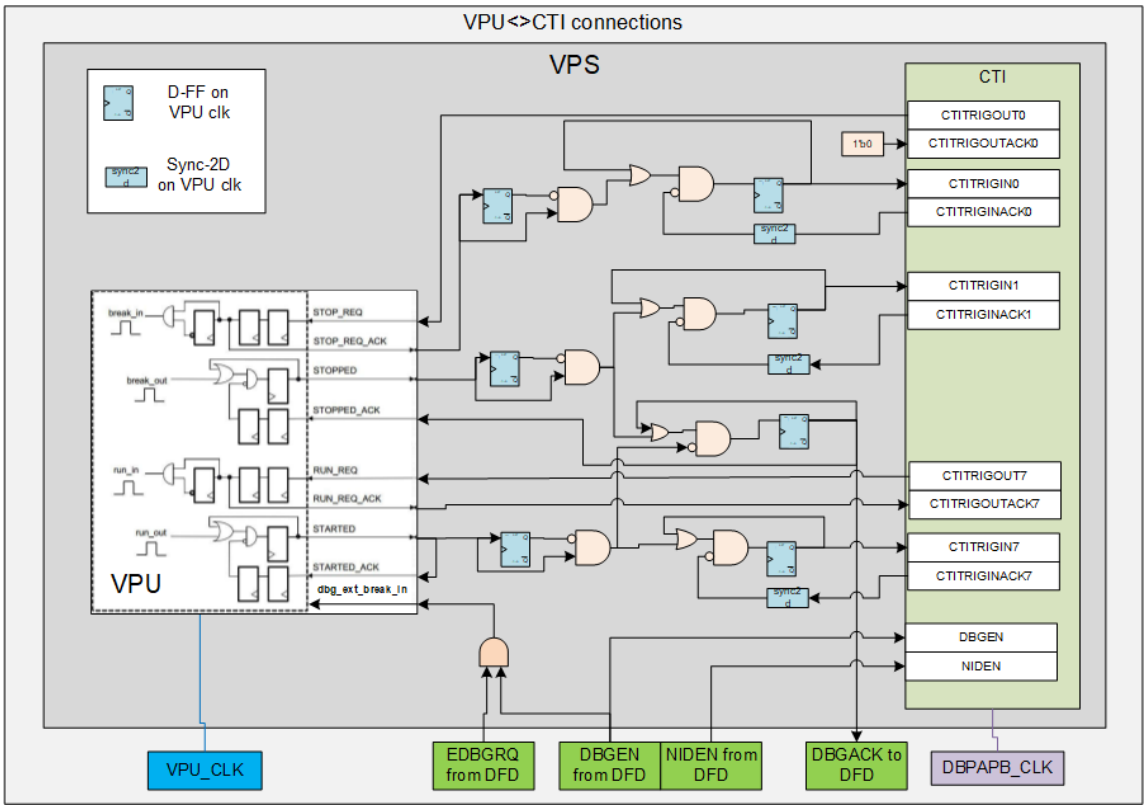
CTI Mappings in VPU

VPU natively has the following signals:

- input cti_stop_req – When this signal is asserted, VPU enters debug mode.
- output cti_stop_req_ack – VPU asserts this signal to indicate that the positive transition on cti_stop_req has been detected by the VPU and that it enters debug mode.
- output cti_stopped – VPU asserts this signal when it enters debug mode (irrespective of the reason that caused VPU to enter debug mode). Signal is deasserted when cti_stopped_ack is asserted.
- input cti_stopped_ack – When this signal is asserted, VPU deasserts cti_stopped. This signal must be asserted for further operation of VPU in debug mode.
- input cti_run_req – When this signal is asserted, VPU exits debug mode.
- output cti_run_req_ack – VPU asserts this signal to indicate that the positive transition on cti_run_req has been detected by the VPU and that it exits debug mode.
- output cti_started – VPU asserts this signal to indicate that it has exited debug mode.
- input cti_started_ack – This signal must be asserted for VPU to deassert cti_started, so that VPU can resume normal operation.
- input dbg_ext_break_in – When this signal is asserted, VPU enters debug mode.

The signals listed above are connected as shown below.

Figure 8.72 VPU to CTI Connections



Summary of Behavior

- VPU enters Debug state when the EDBGRQ signal corresponding to the specific VPU is asserted.
- The DBGACK signal to DFD remains asserted for the entire duration that the VPU is in Debug state.
- Assertion of CTITRIGOUT0 causes VPU to start entering Debug state.
- CTITRIGOUT0 is cleared via software write to CTIINTACK[0] and not by asserting CTITRIGOUTACK0 (similar to Cortex-R5 CTI connections).
- CTITRIGIN0 is asserted when VPU has started entering Debug state (based on CTITRIGOUT0 request) and remains asserted until CTI ACKs this trigger input.
- CTITRIGIN1 is asserted when VPU has entered Debug state and remains asserted until CTI ACKs this trigger input.
- Assertion of CTITRIGOUT7 causes the VPU to exit Debug state and this output remains asserted until ACKed.
- CTITRIGIN7 is asserted when VPU has exited Debug state and remains asserted until CTI ACKs this trigger input.

Table 8.80 CTI Mappings in APE

CTI Trigger	Signal Name of Connected Module	Connected Module	Comments
CTITRIGIN0	DBGTRIGGER	Cortex-A9	APE cluster logic ensures CTITRIGIN0 remains asserted until CTITRIGINACK0 is asserted.
CTITRIGIN1	nPMUIRQ[0]	APE's PMU	Corresponding CTITRIGINACK not connected
CTITRIGIN2	PTMEXTOUT[0]	PTM in APE cluster	(same as above)
CTITRIGIN3	PTMEXTOUT[1]	PTM in APE cluster	(same as above)
CTITRIGIN4	COMMTX	Cortex-A9 in APE cluster	(same as above)
CTITRIGIN5	COMMRX	Cortex-A9 in APE cluster	(same as above)
CTITRIGIN6	PTMTRIGGER	PTM in APE cluster	(same as above)
CTITRIGIN7	0	Tied-off	(same as above)
CTITRIGOUT0	EDBGRQ	Cortex-A9	(EDBGRQ input to Cortex-A9) = (CTITRIGOUT0) |(EDBGRQ_APE from DFD) |(PTMDBGREQ from APE PTM's) CTITRIGOUTACK0 is tied to '0'and CTI relies of clearing CTITRIGOUT0 via software ACK
CTITRIGOUT[4:1]	PTMEXTIN[3:0]	PTM in APE cluster	corresponding CTITRIGOUTACK is tied to '0'
CTITRIGOUT5	CTIEXTTRIG0	Not connected	CTITRIGOUTACK5 (also known as 'CTIEXTTRIGACK0') is tied to '0'.
CTITRIGOUT6	CTIIRQ	GIC in APE cluster	corresponding CTITRIGOUTACK is tied to '0'
CTITRIGOUT7	DBGRESTART	Cortex-A9	

CTI Mappings in CCPLEX

Refer to the Arm DSU TRM, *C1.6 CTI Triggers* section for details.

8.7.2.2.6 DFD Interrupts

nCTIIRQ and nPNUIRQ interrupts are connected to the respective clusters local GIC/VIC and do not have dedicated interrupt slots on the LIC. Refer to Interrupt specifications for details on the GIC/VIC connections to the LIC.

8.7.2.2.7 Clocks

DFD clocks (CSITE, DBGAPB, HSSTP_CLK, and LA) have clock enables implemented in CAR and do not have second level clock gating implemented on them inside different clusters.

The following table lists the clock targets for clocks used by DFD.

Clock	Target	Clock Sources for Clock Mux
CSITE	625MHz @ 0.72V VDD_{SOC/CPU/RTC} (scaled at other MV targets)	clk_s (32kHz), clk_m (default) (=OSC/2=19.2MHz), PLLP (408MHz), REFPLLE (625MHz), PLLC, PLLC2, PLLC3 and PLLC4
DBGAPB	136MHz @ all PVT corners	clk_s (32kHz), clk_m (default) (=OSC/2=19.2MHz), PLLP (408MHz) REFPLLE (625MHz)
HSSTP_CLK	500MHz @ 0.72V (SoC)	UPHY_TX_CLK
HSSTP_RX_CLK	500MHz @ 0.72V (SoC)	UPHY_RX_CLK
JTAG_TCK (JTAG I/O Clock)	50 MHz @ all corners **Note**: DFT needs this to be 100MHz so DFD synthesizes/retimes logic on this clock at 100MHz as well	Derived from TCK clock input on JTAG interface
SWD_IP_CLK (SWD I/O Clock)	10MHz @ all corners	Derived from SWCLK clock input on SWD interface. Also referred to as SWCLK in the document.

Debug APB Clock

This clock (also referred to as the DBGAPB_CLK) is synthesized to operate at 136MHz at all process-voltage-temperature (PVT) corners. While this clock can be completely gated to save power, to avoid debug-lockout, the recommendation for power savings is to program DBGAPB_CLK to a lower frequency (e.g., 19.2MHz). DBGAPB_CLK drives PCLK input of the Cortex-A78 AE core/ DSU and they need the input PCLK running to be able to successfully power-up or power-down. DBGAPB_CLK also needs to be ON while powering up or powering down CCPLEX or individual CPU core. Hence, it is recommended to always keep DBGAPB_CLK ON.

CoreSight ATB Clock

This clock (also referred to as the CSITE_CLK) can be used at frequencies up to 625MHz depending on the rail voltages.

JTAG Clock

This refers to the JTAG interface clock driven by TCK (also referred to as the JTAG_TCK), which works up to 50MHz frequency.

SWD Clock

This refers to the SWD interface clock driven by SWCLK (also referred to as the SWD_IP_CLK), which works up to 1 0MHz frequency.

8.7.2.2.8 Debug Reset

DFD uses a configurable L1 warm reset. This reset supports debug features such as DFD-RAMDUMP and SC7-Debug.

Warm reset is expected to be asserted during SC7 exit and then de-asserted. This is an important aspect for DFD. Warm reset must not be asserted while entering SC7 or while in SC7 -- warm reset must only be asserted during SC7 exit and not earlier.

Also, as L2 reset is sequenced during DFD-RAMDUMP, logic (in DFD, MSS, PMC, PADCTL, etc.) essential for retaining DRAM in self-refresh must be on L1 warm reset to prevent it from getting reset during L2 reset.

DP_AUX Port

Table 8.81 DP_AUX Port

Inputs		**System State (DPAUX port)**
E_I2C CFG	**DPAUX PINMUX CFG**	**Controller active on DPAUX pins**
0	x	Functional (DPAUX)
1	0	Functional (I2C)
1	1	Debug (UART6)

8.7.2.3 Miscellaneous Debug Features

8.7.2.3.1 Reading Device ECID/UID

To obtain the ECID/UID of the device, place the device in recovery mode. On the Linux host machine, open a terminal window to check if the device is listed by entering the lsusb command.

The system returns a listing similar to the following:

Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

Bus 003 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub

Bus 004 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub

Bus 005 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub

Bus 006 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub

Bus 007 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub

Bus 008 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub

Bus 001 Device 002: ID 0409:005a NEC Corp. HighSpeed Hub

Bus 008 Device 002: ID 046d:c05a Logitech, Inc. Optical Mouse M90

Bus 001 Device 003: ID 0403:6011 Future Technology Devices International, Ltd FT4232H Quad HS USB-UART/FIFO IC

Bus 001 Device 021: ID 0955:7021 NVIDIA Corporation

Execute the following command to obtain the UID of the device.

```
sudo ./tegrarcm_v2 --uid
```

This command returns a 16 Byte string similar to the following:

```
BR_CID: 0x018010016419d18714000000009038380
```

8.7.2.3.2 Accessing CBB and DBB using AXI-AP and ETR

AXI-AP issues transactions for CBB and DBB. ETR can stream trace over DBB to memory.

The following subsections describe the connectivity from AXI-AP and ETR to CBB and DBB.

Network Overview

CBB does not support a path to memory; therefore, DFD needs to route memory accesses over DBB and I/O access over CBB.

CBB versus DBB Address Routing

Control Plane (including PCIe apertures) accesses are routed to CBB. Data Plane accesses are routed to DBB.

DBB User Bit Assignments

Table 8.82 AXI Initiator Write Request Channel

AXI-DBB awuser Bits	ETR	AXI-AP
awuser[7:0] = c2mc_StreamID[7:0]	CORESIGHT_CFG_ETR_MC_WR_CTRL_0 [ETR_SID]	CORESIGHT_CFG_CSITE_MC_WR_CTRL_0 [CSITE_SID]
awuser[8] = Coh_IO_acc	CORESIGHT_CFG_ETR_MC_WR_CTRL_0 [COH]	^AWDOMAIN[1:0]
awuser[9] = wsb_ns	AWPROT[1]	AWPROT[1]
awuser[10] = sc2mc_vpr_wr	CORESIGHT_CFG_ETR_MC_WR_CTRL_0 [VPR]	CORESIGHT_CFG_CSITE_MC_WR_CTRL_0 [VPR]
awuser[15:11] = GSC_AID[4:0]	CORESIGHT_CFG_ETR_MC_WR_CTRL_0 [GSC_AID]	CORESIGHT_CFG_CSITE_MC_WR_CTRL_0 [GSC_AID]
awuser[17:16] = GSC_AL[1:0]	CORESIGHT_CFG_ETR_MC_WR_CTRL_0 [GSC_AL]	CORESIGHT_CFG_CSITE_MC_WR_CTRL_0 [GSC_AL]
awuser[18]= user_adr[5]	AWADDR[5]	
awuser[21:19]= { c2mc_adr1[8:6] }	AWADDR[8:6]	
awuser[22]= c2mc_sp_aware	CORESIGHT_CFG_ETR_MC_WR_CTRL_0 [SP_AWARE]	CORESIGHT_CFG_CSITE_MC_WR_CTRL_0 [SP_AWARE]
awuser[25:23]= reserved	-	-
awuser[28:26] = c2mc_user_size	CORESIGHT_CFG_ETR_MC_WR_CTRL_0 [HINT_SIZE]	CORESIGHT_CFG_CSITE_MC_WR_CTRL_0 [HINT_SIZE]

Table 8.83 AXI Initiator Read Request Channel

AXI-DBB aruser Bits	ETR	AXI-AP
aruser[7:0] = c2mc_StreamID[7:0]	CORESIGHT_CFG_ETR_MC_RD_CTRL_0 [ETR_SID]	CORESIGHT_CFG_CSITE_MC_RD_CTRL_0 [CSITE_SID]

AXI-DBB aruser Bits	ETR	AXI-AP
aruser[8] = Coh_IO_acc	CORESIGHT_CFG_ETR_MC_RD_CTRL_0 [COH]	^ARDOMAIN[1:0]
aruser[9] = rsb_ns	ARPROT[1]	ARPROT[1]
aruser[10] = sc2mc_vpr_rd	CORESIGHT_CFG_ETR_MC_RD_CTRL_0 [VPR]	CORESIGHT_CFG_CSITE_MC_RD_CTRL_0 [VPR]
aruser[15:11] = GSC_AID[4:0]	CORESIGHT_CFG_ETR_MC_RD_CTRL_0 [GSC_AID]	CORESIGHT_CFG_CSITE_MC_RD_CTRL_0 [GSC_AID]
aruser[17:16] = GSC_AL[1:0]	CORESIGHT_CFG_ETR_MC_RD_CTRL_0 [GSC_AL]	CORESIGHT_CFG_CSITE_MC_RD_CTRL_0 [GSC_AL]
aruser[18]= user_adr[5]	ARADDR[5]	
aruser[21:19]= { c2mc_adr1[8:6] }	ARADDR[8:6]	
aruser[22]= c2mc_sp_aware	CORESIGHT_CFG_ETR_MC_RD_CTRL_0 [SP_AWARE]	CORESIGHT_CFG_CSITE_MC_RD_CTRL_0 [SP_AWARE]
aruser[25:23]= reserved	-	-
aruser[28:26] = c2mc_user_size	CORESIGHT_CFG_ETR_MC_RD_CTRL_0 [HINT_SIZE]	CORESIGHT_CFG_CSITE_MC_RD_CTRL_0 [HINT_SIZE]

AXI Initiator Read Data Channel

The only user bit driven on the Read Data channel is rnv_vpr_resp_user. This signal indicates that this response data was sourced from VPR aperture. This offers a hardware-only mechanism for the Initiator to know if the data is from the VPR region or not. Given that neither AXI-AP nor ETR are allowed to read from the VPR region, this optional bit is not taken in by DFD partition.

Register Definitions

The table below lists the register definitions of the CORESIGHT_CFG_{ETR,CSITE}*MC*{RD,WR}_CTRL_0.

Bits	ETR	CSITE
[28:26] = c2mc_user_size	RW=0	RW=0
[25:23] = reserved	RW=0 (not used)	RW=0 (not used)
[22] = SP_AWARE	RW=0	RW=0

Bits	ETR	CSITE
[21:19] = ADR1_Bit8toBit6	RW=0	RW=0
[18] = USER_ADR_5	RW=0	RW=0
[17:16] = GSC_AL	RO=0	RO=0
[15:11] = GSC_AID	RW=0	RW=0
[10] = VPR_{RD,WR}	RO=0	RO=0
[9] = {RSB,WSB}_NS	RW=0 (not used)	RW=0 (not used)
[8] = COH	RW=0	RW=0 (not used)
[7:0] = SID	RW=0	RW=0

Sub-Partition Bits

Hardware Design Guidelines

```
If (sp_aware==0)
 {
 c2mc_adr1[8:6] = AxADDR[8:6];
 user_adr[5] = AxADDR[5];
 }
 Else
 {
 c2mc_adr1[8:6] =
 CORESIGHT_CFG_{CSITE}_MC_{RD,WR}_CTRL_0[ADR1_Bit8toBit6];
 user_adr[5] =
 CORESIGHT_CFG_{CSITE}_MC_{RD,WR}_CTRL_0[USER_ADR_5];
 }
```

Software Usage Guidelines

Software should not program the SP_AWARE field unless it wishes to use the sub-partition feature in MC. To use the sub-partition aware feature for accesses from the AXI-AP (CSITE) port, program the SP_AWARE ADR1_Bit8toBit6 fields in the configuration register corresponding to the access {CSITE}x{RD,WR}. Then make the sp-aware access on DBB.

CBB User Bit Assignment

Table 8.84 AXI Initiator Write Request Channel

AXI-CBB awuser Bits	CSITE
awuser[16:11] = MSTR_ID[5:0]	CSITE's MSTR-ID (0xA)
awuser[10:9] = VQC[1:0]	CORESIGHT_CFG_CSITE_CBB_WR_CTRL_0 [VQC]

AXI-CBB awuser Bits	CSITE
awuser[8:2] = GRPSEC[6:0]	MISCREG_CSITE_SECURITY_0 [G7W:G1W]
awuser[1:0] = FALCONSEC[1:0]	CORESIGHT_CFG_CSITE_CBB_WR_CTRL_0 [FALCONSEC]

Table 8.85 AXI Initiator Read Request Channel

AXI-CBB aruser Bits	CSITE
aruser[16:11] = MSTR_ID[5:0]	CSITE's MSTR-ID (0xA)
aruser[10:9] = VQC[1:0]	CORESIGHT_CFG_CSITE_CBB_RD_CTRL_0 [VQC]
aruser[8:2] = GRPSEC[6:0]	MISCREG_CSITE_SECURITY_0 [G7R:G1R]
aruser[1:0] = FALCONSEC[1:0]	CORESIGHT_CFG_CSITE_CBB_RD_CTRL_0 [FALCONSEC] RO = b'00

Register Definitions

The following table lists the register definitions of the CORESIGHT_CFG_{CSITE}_*CBB*_{WR,RD}_CTRL_0 registers described above.

Bits	CSITE/AXI-AP
[16:11] = MSTR_ID[5:0]	RO = 0xA (not used)
[10:9] = VQC[1:0]	RW = b'00
[8:2] = GRPSEC[6:0]	RO = b'000_0000 (not used)
[1:0] = FALCONSEC[1:0]	RO = b'00

CORESIGHT_CFG_CSITE_CBB_{WR,RD}_CTRL_0 are protected by an independent SCR.

ETR AXI Interface

Although ETR uses an AXI3 interface, it does not support Transaction IDs (AWID, WID, BID, ARID, and RID). The ETR's AXI interface does not have ports for these signals.

Since ETR is an initiator on DBB, the Transaction ID signals for ETR must be driven as follows:

- For the ETR AXI-DBB port, ARID is tied to 0x0 while AWID is tied to 0x1.
- Ignore BID and RID (as these are driven by target).
- AXI-DBB does not support WID (although it claims to be AXI3 based socket), but the absence of WID on the ETR interface is not an issue. If AXI-DBB has a WID port, it is tied to 0x1 to indicate that ETR does not support write interleaving.

SMMU Bypass Ability for AXI-AP

MSS restricts non-SMMU translated (PA) accesses only to GSCs. However, for debug purposes, AXI-AP needs the ability to bypass SMMU and still access non-GSC regions using PA, in addition to regular abilities to access GSC regions using PA and non-GSC regions using VA. MSS supports two methods for AXIAP to be able to bypass SMMU and still be able to access a non-GSC region in memory:

- Using AXIAP_SMMU_BYPASS_CARVEOUT_CHECK
- Using AXI_AP_VIRTUAL_CLIENT

Ensure BCT setting is such that AXIAP_SMMU_BYPASS_CARVEOUT_CHECK is ‘CARVEOUT_ONLY’, {AXIAPR,AXIAPW}_SMMU_BYPASS_ALLOW are 'DISALLOW'', AXI_AP_VIRTUAL_CLIENT is 'NO_SET_VIRTUAL_CLIENT ' and AXI_AP_VIRTUAL_CLIENT_WRITE_ACCESS=1 (lock)

8.7.2.3.3 Support Coherency Debug via CoreSight AXI-AP

ACE Lite Background

CoreSight supports ACE-Lite protocol for AXI interface at AXI-AP.

AxSNOOP and AxBAR Optimization

As CoreSight does not contain a cache of its own, AxSNOOP attribute can be ignored and assumed tied to 0b000. Support for native barrier transactions is absent in DBB, and so AxBAR can also be ignored and assumed tied to 0b0.

AxDOMAIN Optimization

When AxSNOOP[3:0]==0b000 and AxBAR[0]==0b0,

- ^AxDOMAIN[1:0]==0 implies Non Snooping Access and
- ^AxDOMAIN[1:0]==1 implies Coherent Access.

AXI-DBB socket does not support the 2-bit signal, AxDOMAIN[1:0] on the read/write address channels, but rather supports a 1-bit signal, COH_IO_ACC[0] on the read/write address channels.

- COH_IO_ACC==0 generates a Non-Snooping Access whereas
- COH_IO_ACC==1 generates a Coherent Access.

Assigning COH_IO_ACC = ^AxDOMAIN[1:0] is reasonable, but only when SMMU is bypassed. When a transaction does not bypass SMMU, coherency/snoop is determined by the page table attribute in the SMMU.

8.7.2.3.4 Debugging Clusters with DCLS Implementation

For safety reasons, some of the auxiliary Cortex-R5 clusters (e.g., BPMP, SCE, RCE, DCE) have used the Dual-Core Lock-Step Implementation to detect faults using redundancy and comparison. However, as the debug logic external to the Cortex-R5 cores is not replicated, these debug signals are monitored on a Check Not Asserted (CNA) Bus. Assertion of any signal on the CNA bus can be propagated as a fault indication. This can prevent debug because as soon as any debug transaction occurs, it could potentially lead to a fault and thus change the system environment. Also, there could be certain issues that can only be replicated when DCLS is enabled, so the ability to debug while DCLS is enabled is necessary. To support debugging while DCLS is enabled, configurable bits are provided that can disable CNA bus monitoring thus preventing assertion of signals that are monitored on the CNA bus from leading to an error/reset/fault assertion.

Subsequent subsections describe the guidelines to debug clusters with DCLS enabled for the Cortex-R5 cores.

BPMP

Programming the following registers prevents the debug signals from generating any comparator error/fault or trigger a CNA event/error/fault:

1. Write "1" to CPU_CNA_ERR field in the BPMP_MISC_ERR_RESET_MASK2 register (**Note:** Software usually leaves this field to its reset value (1), so programming this during debug might not be necessary.)
2. Write "0" to ERR63_ENABLE field in the BPMP_ERROR_COLLATOR_ERRSLICE1_MISSIONERR_ENABLE_0 register
3. Write "0" to ERR63_ENABLE field in the BPMP_ERROR_COLLATOR_ERRSLICE1_LATENTERR_ENABLE_0 register

APS_PROC_CFG_DCLS_ENABLE=1 allows the Cortex-R5s to execute in lock-step despite disabling the debug signal monitoring. This allows for debugging BPMP without disabling lock-step execution of the Cortex-R5-cores.

SCE

Program the following fields with '0' to prevent debug transactions to SCE from generating errors/faults:

1. ERR77 field in the SCE_EC_REGS_ERRSLICE2_MISSIONERR_ENABLE register in SCE_ERR_COLLATOR amap space
2. ERR77 field in the SCE_EC_REGS_ERRSLICE2_LATENTERR_ENABLE register in SCE_ERR_COLLATOR amap space
3. ERR77 field in the SCE_MISC_AUTOLOCK_EC_ERRSLICE2_MISSIONERR_ENABLE register in SCE_MISC amap space
4. ERR77 field in the SCE_MISC_AUTOLOCK_EC_ERRSLICE2_LATENTERR_ENABLE register in SCE_MISC amap space

APS_PROC_CFG_DCLS_ENABLE=1 allows the Cortex-R5s to execute in lock-step despite disabling the debug signal monitoring. This allows for debugging SCE without disabling lock-step execution of the Cortex-R5 cores.

RCE

The programming steps are similar to SCE. Replace “SCE” with “RCE” in the programming sequence mentioned in the SCE subsection above.

DCE

The programming steps are similar to SCE. Replace “SCE” with “DCE” in the programming sequence mentioned in the SCE subsection above.

FSI

Software programs the following before enabling FSI debug to prevent FSI-debug enablement from generating errors or faults:

1. NV_ADDRESS_MAP_FSI_MISC_GEN_CFG_BASE + CHECK_NOT_ASSERTED.CPU_CNA = 0 (reset default)
2. NV_ADDRESS_MAP_FSI_MISC_GEN_CFG_BASE + CHECK_NOT_ASSERTED.CPU_CHSM (reset default)

The reset values ensure that during bring-up, faults are disabled by default out of reset.

In FSI-R52, individual debug enables can be enabled/disabled from generating faults by programming:

1. FSI_HSM_SOC_ERROR_EN_<> [FSI_CPU_CNA_*_UE mod 32] = 0
2. FSI_HSM_INT_EN_<> [FSI_CPU_CNA_*_UE mod 32] = 0

Also program the following before issuing any debug transaction to FSI:

1. FSI_R5_EC_ERRSLICE1_MISSIONERR_ENABLE_0[ERR44]=DISABLE
2. FSI_R5_EC_ERRSLICE1_LATENTERR_ENABLE_0[ERR44]=DISABLE

Reset Guidelines

Upon reset, debug bus monitoring is disabled. This ensures that DCLS_ENABLE=0 upon reset. Thus, even if the error-enable fields in the error-collator registers are '1' upon reset, that is not an issue as DCLS_ENABLE=0 suppresses all interrupts/errors/resets/faults from being generated due to assertions on the debug bus.

8.7.2.3.5 Unified UART

In the SoC, depending on platform requirements, the following UARTs are available for printing debug logs:

- UART3/UARTC
  - This UART is in AO-Cluster and hence can be used even in low power states like SC7.
- UART6/UARTF
  - This UART is pinmuxed over the DPAUX0 pins.
- UART8/UARTH
  - This UART is pinmuxed over USB2-Port0 and USB2-Port1 and can be available over either of the two at a time.

Refer to the Software Application Note on how multiple processors can print UART logs over a single UART without getting the eventual output log entangled.

8.7.2.3.6 ACCESS_BLOCK and ACCESS_TIMEOUT on Debug-APB

To not hang the debugger itself while making accesses on the Debug APB, CoreSight has an internal NV module to:

- Timeout the Debug APB accesses for each of the debug-APB clients on no response.
- Block the accesses that are guaranteed to result in timeout due to a non-responsive debug APB client. Either of these could be non-responsive due to power down, clock gating, or under reset.

Both the timeout and access block features can be separately disabled by using bits, TIMEOUT_EN and BLOCK_ACCESS_EN, in the CORESIGHT_CFG_CTL_0 register.

In case of Timeout, users have an option between choosing between a long timeout of 0xfffff counts(default) versus a short timeout of 0x200 clock cycles of DBGAPB_CLK. This can be done via the CORESIGHT_CFG_CTL_0.

In the event of either a timeout or access block, deadcodes are returned on the prdata bus to immediately identify what caused a particular access to timeout or get blocked. Corresponding pready signal from initiator is also asserted (1'b1).

On receiving the plsverr, the debugger clears the sticky status bit in the DAP (this is STICKYERR bit in CTRL/STAT register of JTAG-DP/SW-DP, depending on which external debug interface, JTAG or SWD, is being used) and then query the respective CORESIGHT_CFG_<module>_ALIVE_STATUS register to check the cause of timeout/blocking.

If the transaction was terminated due to the timeout generated by DFD TIMEOUT logic, then corresponding .*_DBGAPB_TIMEOUT bit in respective CORESIGHT_CFG_.*_ALIVE_STATUS_0 is set

to ‘1’. Only a reset or writing (MMIO write-access to this register[field]) a ‘1’ to this bit clears the bit to ‘0’.

Note: MMIO writes of ‘0’ to any field in CORESIGHT_CFG_.*_ALIVE_STATUS_0 do not change the value of that field. Also, apart from .*_DBGAPB_TIMEOUT, all other fields in CORESIGHT_CFG_.*_ALIVE_STATUS_0 should be read-only (‘RO’).

The table below lists the accesses that are blocked by CoreSight. Note that for accesses that are not blocked, they result in a timeout over the debug APB if the CPUs are not able to respond before the timeout.

	Yes indicates Accesses Blocked by CoreSight		
	Powergated?	**Under Reset?**	**Clock gated?**
Cortex-R5 | BPMP	No	Yes	No
Cortex-R5 | SCE	No	Yes	No
Cortex-R5 | RCE	No	Yes	No
Cortex-R5 | DCE	No	Yes	No
CVPVA	Yes	No	No
PVA{0}_R5	Yes	Yes	No
PVA{0}_VPU{0,1}	Yes	Yes	No
Cortex-R5 | SPE	Yes	Yes	No
Cortex-A9 | APE	Yes	No	Yes
HSSTP	No	Yes	No
FSI	Yes	Yes	Yes

- The Cortex-R5s have Processor Integration Layers (PIL) around them that allow access to the debug ROM table even if Cortex-R5s are clock-gated, and therefore cannot be called as non-responsive.
- For processors with multiple resets, only nsysporresetn and presetdbg are relevant to derive a non-responsive status. The former puts both processor and non-processor logic with debug registers under reset, and the latter impacts the pipe-stages that transport debug signals.
- APE uses the same reset as DFD (csitel2rstn, which is an L2 warm reset). Accesses to APE do not use its reset status for blocking accesses. However, as APE’s debug-APB interface is on CSITE-CLK, and as there is a possibility of CSITE_CLK being off, accesses to APE are blocked when CSITE_CE=0 (i.e., when CSITE_CLK is OFF).
- For CVPVA, PVA0, and PVA1, the CoreSight accesses are blocked under following circumstances:

a. CoreSight accesses to CVPVA, PVA0, and PVA1 are blocked when one of the following conditions becomes true
   i. CVPVA is rail gated
   ii. Clock gating not applicable as components use DBGAPB and CSITE clocks (same as CoreSight)
b. CoreSight accesses to entire PVA<p> (includes Cortex-R5 PIL and VPS clusters in corresponding PVA) are blocked when one of the following conditions becomes true (<p> ∈ {0,1})
   i. PVA<p> is power gated
   ii. PVA<p> debug is under reset (PRESETDBGn associated with corresponding Cortex-R5 cluster in the PVA cluster is asserted)
   iii. Clock gating not applicable as CoreSight components use DBGAPB and CSITE clocks (same as CoreSight)
c. CoreSight accesses to a Cortex-R5 cluster inside a PVA<p> cluster are blocked when one of the following conditions becomes true
   i. PVA<p> Cortex-R5 cluster is under reset (nSYSPORESET associated with the Cortex-R5 cluster is asserted)
   ii. Power gating not applicable because power gating only occurs at entire PVA-level and not individually for sub-clusters of PVA like Cortex-R5 cluster and VPS-clusters.
   iii. Clock gating not applicable as CoreSight components use DBGAPB or CSITE clocks (same as CoreSight)
d. CoreSight accesses to a VPS<p,v> cluster inside a PVA<p> cluster are blocked when one of the following conditions becomes true (<p,v> ∈ {00,01,10,11})
   i. VPS<p,v> debug is under reset (VPUPRESETDBGn associated with the VPS<p,v> cluster is asserted)
   ii. Power gating not applicable because power gating only occurs at entire PVA-level and not individually for sub-clusters of PVA like Cortex-R5 cluster and VPS-clusters.
   iii. Clock gating not applicable as CoreSight components use DBGAPB or CSITE clocks (same as CoreSight)

CV Rail Clamp	PVA<p> Power Clamp	PVA<p>_ R5PIL PRESETD BGn	PVA<p>_ R5PIL SYSPORE SETn	PVA<p>_VP S<v> VPUPRESET DBGn	CVPVA	PVA<p>	PVA<p>_R 5PIL	PVA<p>_V PS<v>
1	x	x	x	x	Blocked	Blocked	Blocked	Blocked
0	1	x	x	x	ALLOW	Blocked	Blocked	Blocked
0	0	0	x	x	ALLOW	Blocked	Blocked	Blocked
0	0	1	0	0	ALLOW	ALLOW	Blocked	Blocked
0	0	1	1	0	ALLOW	ALLOW	ALLOW	Blocked

CV Rail Clamp	PVA<p> Power Clamp	PVA<p>_ R5PIL PRESETD BGn	PVA<p>_ R5PIL SYSPORE SETn	PVA<p>_VP S<v> VPUPRESET DBGn	CVPVA	PVA<p>	PVA<p>_R 5PIL	PVA<p>_V PS<v>
0	0	1	0	1	ALLOW	ALLOW	Blocked	ALLOW
0	0	1	1	1	ALLOW	ALLOW	ALLOW	ALLOW

where <p,v> ∈ {00,01,10,11}

- For CCPLEX, CCPLEX logic in SOC-NPG domain on system L2warm reset has its clocks enabled by default and guaranteed to respond (either with data, RAZ/WI or pslverr) to all transactions. Access blocking is not done for transactions directed towards CCPLEX.
- For HSSTP, the HSSTP CAR reset is used for ACCESS_BLOCK for entire HSSTP-Collator subsystem. If a transaction is directed to a register under the internal HSSTP reset (e.g., TPIU, which is a subset of HSSTP-Collator subsystem), ACCESS_BLOCK may not always be able to intercept the transaction, and ACCESS_TIMEOUT intervenes. Additionally, when any of the APBICs is in reset, it returns a pslverr, as indicated in Debug APB Interface definition in CoreSight SAS.
- For FSI, access to a core and its related debug components (PMU, CTI, ETM) is blocked in the following cases:
  a. Access to entire FSI is blocked if
     i. FSI Rail Clamp is ‘1’ or
     ii. FSI MV APB Async Bridge’s CSYSREQ is ‘0’
     iii. FSI_CSITE_RSTN=’0’ or
     iv. FSI_CSITE_CE=0
  b. Access to entire FSI Cluster-0 is blocked if
     i. FSI Cluster-0’s nSYSPORESET or nPRESETDBG is ‘0’
  c. Access to entire FSI Cluster-1 is blocked if FSI Cluster-1’s nTOPRESET, or nPRESETDBG is ‘0’.
  d. Access to an R52 core<x> DBG, PMU and ETM (not CTI) is blocked if corresponding nCPUPORESET<x> is ‘0’.

Interface for CoreSight Deadcodes and Timeouts

CoreSight needs the following sidebands to use in the timeout/access block logic for CoreSight for Cortex-R5:

- Cortex-R5’s sysporreset
- Cortex-R5’s global clock enable
- Cortex-R5’s main powerdown clamp for the cluster

When blocking is disabled and timeout enabled, the deadcode obtained from Cortex-R5s is 0xdead1002 as opposed to the expected deadcode 0xdead000A (timeout). This is because, when

PG-ed the ports are clamped in such a way that 0xdead1002 is returned with PSLVERR, a read transaction is issued over Debug APB. So, the access does complete without a timeout and the deadcode is natively returned from the client.

List of Timeout and Access Block Deadcodes

Refer to the field description for CORESIGHT_CFG_CTL_0[TIMEOUT_EN] and CORESIGHT_CFG_CTL_0[BLOCK_ACCESS_EN] for a list of corresponding deadcodes.

Unit	TIMEOUT_EN deadcode	BLOCK_ACCESS_EN deadcode	Comments
CCPLEX	0xdead0002	N/A	
APE	0xdead0004	0xdead0003	
LA	0xdead0006	0xdead0005	
BPMP	0xdead0008	0xdead0007	
SPE	0xdead000A	0xdead0009	
SCE	0xdead000C	0xdead000B	
RCE	0xdead000E	0xdead000D	
DCE	0xdead0010	0xdead0011	
-	-	-	
HSSTP	0xdead0014	0xdead0015	
FSI	0xdead0016	0xdead0017	
All remaining apertures on Major APBIC that do not use DBGAPB-CLK and are not already covered by ACCESS-TIMEOUT (e.g., Major Funnel, ETF, Replicator, ETR, STM, CTI-TOP0, CTI-TOP1)	0xdead0018	-	
All remaining apertures on Minor APBIC that do not use DBGAPB-CLK and are not already covered by ACCESS-TIMEOUT (e.g., Minor Funnel)	0xdead0020	-	

Unit	TIMEOUT_EN deadcode	BLOCK_ACCESS_EN deadcode	Comments
PVA	0xdead0041 – PVA Collator RomTable/Funnel Debug APB Timed Out 0xdead0043 – PVA0- R5PIL Debug APB Timed Out 0xdead0045 – PVA0-VPS0 Debug APB Timed Out 0xdead0047 – PVA0-VPS1 Debug APB Timed Out 0xdead0049 – PVA1- R5PIL Debug APB Timed Out 0xdead004B – PVA1-VPS0 Debug APB Timed Out 0xdead004D – PVA1-VPS1 Debug APB Timed Out 0xdead004F – PVA0 Rom Table Timed Out 0xdead0051 – PVA1 Rom Table Timed Out	0xdead0040 – PVA Collator RomTable/Funnel is nonresponsive 0xdead0042 – PVA0- R5PIL is nonresponsive 0xdead0044 – PVA0-VPS0 is nonresponsive 0xdead0046 – PVA0-VPS1 is nonresponsive 0xdead0048 – PVA1- R5PIL is nonresponsive 0xdead004A – PVA1-VPS0 is nonresponsive 0xdead004C – PVA1-VPS1 is nonresponsive 0xdead004E – PVA0 Rom Table is nonresponsive 0xdead0050 – PVA1 Rom Table is nonresponsive	

Note:

- When an SCR blocks an access, a pslverr is returned with deadcode: 0xdead1007. Also, if an access is made to an unmapped offset in an slvgen based APB-target, deadcode: 0xdead1001 is returned.
- CCPLEX may return a pslverr with deadcode 0xDEADCCAF when ACDI detects timeout on debug APB bus within CCPLEX. It also sets a bit that prevents further accesses to same target until the EDACR register is read. CCPLEX returns a pslverr with deadcode 0xDEADCCA0 when SCF is in reset.

8.7.2.3.7 EDBGRQ, DBGACK, and Freezing Timers During Debug

CORESIGHT_CFG_EXTERNAL_DEBUG_REQ_0[EDBGRQ_.*]

EDBGRQ bits corresponding to the following processors exist in CORESIGHT_CFG_EXTERNAL_DEBUG_REQ_0:

- APE, SPE
- SCE, RCE, DCE
- PVA0-R5, PVA0-VPU0, PVA0-VPU1
- FSI Cluster-1 Cores [0-3], FSI Cluster-0 Core0

If the processor's debug is enabled, asserting the corresponding EDBGRQ causes the processor to enter debug state.

EDBGRQ is Not Cleared by DBGACK

CORESIGHT_CFG_EXTERNAL_DEBUG_REQ_0 contains custom EDBGRQ bits corresponding to different processors which, if set, can cause the corresponding processor to enter debug mode if the processor's debug has been allowed by its corresponding debug authentication signals. Assertion of a DBGACK bit (observable in CORESIGHT_CFG_DEBUG_ACK_0) does not automatically clear the corresponding EDBGRQ bit (in CORESIGHT_CFG_EXTERNAL_DEBUG_REQ_0). Hence, if EDBGRQ is used to put a processor in debug mode, it should be cleared before trying to get the processor out of debug mode, else if EDBGRQ is left asserted, the processor does not exit debug mode.

Typically, tools are expected to use the CTI registers (accessible over the debug-APB bus) to trigger debug-request to a particular processor rather than using the EDBGRQ bits in CORESIGHT_CFG_EXTERNAL_DEBUG_REQ_0.

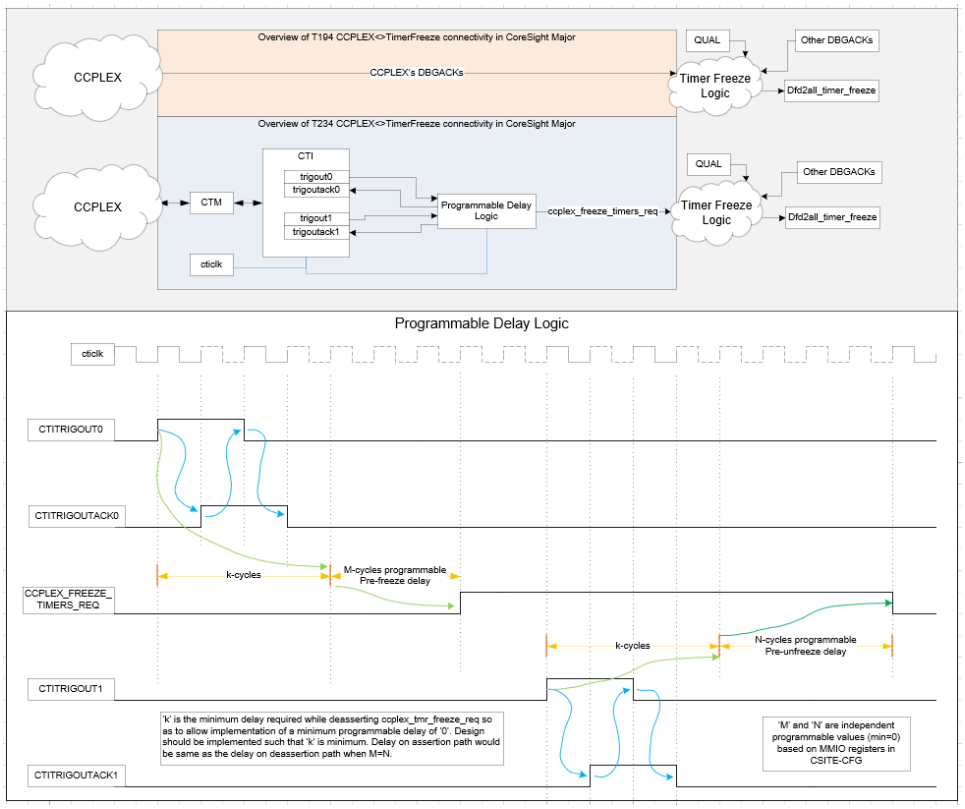
Generating Timer Freeze Request During CCPLEX Debug

As the DSU wrapper that encapsulates the CCPLEX CPUs does not expose EDBGRQ or DBGACK, the cross-trigger network is used to propagate the triggers that put CCPLEX CPUs in and out of debug state to a timer freeze request generation logic.

An assertion on CTI1's CTITRIGOUT0 is used to assert the timer freeze request from ccplex (ccplex_freeze_timers_req) and assertion on CTI1's CTITRIGOUT1 is used to de-assert ccplex_freeze_timers_req. This request is then merged with qualified timer-freeze requests from the rest of the processors in the SoC to generate a final timer freeze request that is propagated to all the timers.

The following figure describes how the CTI1 triggers assert/deassert the timer freeze request.

Figure 8.73 CTI1 Triggers Assert/Deassert Timer Freeze Request



When ctitrigout0 and ctitrigout1 are asserted simultaneously, ctitrigout0 is given priority.

‘M’ and ‘N’ are a 32-bit value indicating number of additional cticlk-cycle delays to be added on assertion and deassertion paths.

‘M’ is derived from MMIO register CORESIGHT_CFG_TMR_FREEZE_DLY_0[31:0]

‘N’ is derived from MMIO register CORESIGHT_CFG_TMR_UNFREEZE_DLY_0[31:0]

CORESIGHT_CFG_TMR_{FREEZE,UNFREEZE}_DLY_0 are protected by CORESIGHT_CFG_SCR_TMR_FREEZE_DLY_0.

‘k’ value is k = 3.

DBGACK Aggregation

CoreSight collects the DBGACKs from all the CPUs and updates them in its status registers.

CORESIGHT_CFG_DEBUG_ACK_0[0] – DBGACK CCPLEX (common for all CCPLEX cores)

CORESIGHT_CFG_DEBUG_ACK_0[4:1] – DBGACK FSI R52 cores [3:0]

CORESIGHT_CFG_DEBUG_ACK_0[5] – DBGACK FSI R5

CORESIGHT_CFG_DEBUG_ACK_0[16] – DBGACK BPMP

CORESIGHT_CFG_DEBUG_ACK_0[17] – DBGACK APE

CORESIGHT_CFG_DEBUG_ACK_0[18] – DBGACK SPE

CORESIGHT_CFG_DEBUG_ACK_0[19] – DBGACK SCE

CORESIGHT_CFG_DEBUG_ACK_0[20] – DBGACK RCE

CORESIGHT_CFG_DEBUG_ACK_0[26:21] – DBGACK PVA

CORESIGHT_CFG_DEBUG_ACK_0[27] – DBGACK DCE

Unused CORESIGHT_CFG_DEBUG_ACK_0 bits are tied to '0'.

CORESIGHT_CFG_DEBUG_ACK_QUAL_0 in the CoreSight configuration aperture has a qualifying bit corresponding to each assigned DBGACK bit in CORESIGHT_CFG_DEBUG_ACK_0. When the bit in CORESIGHT_CFG_DEBUG_ACK_QUAL_0 is set, the corresponding DBGACK in CORESIGHT_CFG_DEBUG_ACK_0 participates in the timer freeze, else the effect of that DBGACK is ignored. The enable bit offset in CORESIGHT_CFG_DEBUG_ACK_QUAL_0 must match the corresponding CPU DBGACK in CORESIGHT_CFG_DEBUG_ACK_0.

The non-ccplex halt equation then becomes:

soc_freeze_timers_req=

| (CORESIGHT_CFG_DEBUG_ACK_0[m:0] &

CORESIGHT_CFG_DEBUG_ACK_QUAL_0[m:0])

The global timer halt equation:

dfd2all_freeze_timers_req = soc_freeze_timers_req | ccplex_freeze_timers_req

Each of the timers internally adds a halt on the debug (HDBG) bit to allow it to halt/continue ticking when the dfd2all_freeze_timers_req is asserted. This register bit in the timer AMAP page can then follow the security attributes for the timer AMAP or add its own security attribute via the Security Control Register.

8.7.2.3.8 Logging External Debug Attempts

This feature logs accesses made by an external debugger connected via JTAG or SWD to access the system via CoreSight. It does not log attempts made by external debugger if those attempts are not even able to generate a transaction from DP to AP.

Additionally, it also logs the number of times TMS=0.

Note: The counting above is done on DBGAPB clock. DBGAPB clock is always expected to be running (although at lower frequencies ~9.6MHz or ~32kHz in mission mode to save energy). If DBGAPB clock is turned off, the above counters are stalled. Similarly, TMS is usually used by logic on TCK. Hence, if TMS and TCK are toggled such that TMS manages to remain '1' during positive edge of DBGAPB clock, then TMS=0 will not get logged. Logging TCK is not feasible as that is left floating unlike TMS, which has a weak pullup to have a steady value when unused.

Algorithm

Let,

invalid_attempt = (('dapenables' input of DAPBUS interconnect == '1') && (global_jtag_enable==0))

valid_attempt = (('dapenables' input of DAPBUS interconnect == '1') && (global_jtag_enable==1))

If 'invalid_attempt' transitions from 0->1 on DBGAPB_CLK increment CORESIGHT_CFG_CSITE_INVALID_ATTEMPT_LOG (that is, if 'invalid_attempt' sampled on previous clock cycle is '0' and on current clock cycle is '1', increment CORESIGHT_CFG_CSITE_INVALID_ATTEMPT_LOG).

If 'valid_attempt' transitions from 0->1 on DBGAPB_CLK increment CORESIGHT_CFG_CSITE_VALID_ATTEMPT_LOG.

If 'TMS' value transitions from 1->0 on DBGAPB_CLK increment CORESIGHT_CFG_CSITE_TMS_LOG.

Given <p> ∈ { INVALID_ATTEMPT,VALID_ATTEMPT,TMS },

- CORESIGHT_CFG_CSITE_<p>_<k> is conceptual 64 bit representation of two physical MMIO 32-bit registers CORESIGHT_CFG_CSITE_<p>_<k>_LOW[31:0] and CORESIGHT_CFG_CSITE_<p>_<k>_HIGH[31:0].
- Given <k> ∈ { LOG , SNAP }
  - CORESIGHT_CFG_CSITE_<p>_<k> [63:0] = { CORESIGHT_CFG_CSITE_<p>_<k>_HIGH[31:0], CORESIGHT_CFG_CSITE_<p>_<k>_LOW[31:0]}
- If CORESIGHT_CFG_CSITE_<p>_LOG not equal to CORESIGHT_CFG_CSITE_<p>_SNAP, assert csite_<p>_trig =1; else csite_<p>_trig =0;

- If software writes ‘1’ to CORESIGHT_CFG_EXT_DBG_LOG_CTL[SNAP_CSITE_<p>], value of CORESIGHT_CFG_CSITE_<p>_LOG is copied into CORESIGHT_CFG_CSITE_<p>_SNAP (which results in deassertion of csite_<p>_trig to ‘0’ too). SNAP_CSITE_<p> clears to ‘0’ on the next clock cycle after which it was set to ‘1’;
- Value of csite_<p>_trig is reflected in CORESIGHT_CFG_EXT_DBG_LOG_STATUS [CSITE_<p>_TRIG] respectively.
- Below are on L1warm reset (csite_l1sysreset_rstn) and their reset value is ‘0’:
  - CORESIGHT_CFG_CSITE_<p>_((LOG)|(SNAP))_((HIGH)|(LOW))
  - CORESIGHT_CFG_EXT_DBG_LOG_CTL[SNAP_CSITE_<p>]
  - csite_<p>_trig
  - CORESIGHT_CFG_EXT_DBG_LOG_STATUS [CSITE_<p>_TRIG]
  - CORESIGHT_CFG_EXT_DBG_TRIG_MASK [CSITE_<p>]
- The following registers are RO for software:
  - CORESIGHT_CFG_CSITE_<p>_((LOG)|(SNAP))_((HIGH)|(LOW))
  - CORESIGHT_CFG_EXT_DBG_LOG_STATUS [CSITE_<p>_TRIG]
- The following are RW for software:
  - CORESIGHT_CFG_EXT_DBG_LOG_CTL[SNAP_CSITE_<p>]
    - This bit can be set to ‘1’ by software and hardware auto-clears it on subsequent cycle. Priority is given to hardware clearing action than software write action.
  - CORESIGHT_CFG_EXT_DBG_TRIG_MASK
- CORESIGHT_CFG_SCR_EXT_DBG_LOG_STATUS protects:
  - CORESIGHT_CFG_CSITE_<p>_((LOG)|(SNAP))_((HIGH)|(LOW))
  - CORESIGHT_CFG_EXT_DBG_LOG_STATUS [CSITE_<p>_TRIG]
- CORESIGHT_CFG_SCR_EXT_DBG_LOG_CTL protects:
  - CORESIGHT_CFG_EXT_DBG_LOG_CTL
- CORESIGHT_CFG_SCR_EXT_DBG_TRIG_MASK protects:
  - CORESIGHT_CFG_EXT_DBG_TRIG_MASK
- As always, SCRs are on csite_sysreset_rstn (which is an l2warm reset).

The above algorithm is implemented on logic running on DBGAPB_CLK.

Typical Usage Model

The LOG counters start updating out of reset. However, out of reset, the MASK is ‘0’ and hence, no interrupt is generated. Programming the corresponding MASK bit as ‘1’ can enable interrupt generation. If any activity prior to this needs to be ignored, write ‘1’ to corresponding CORESIGHT_CFG_EXT_DBG_LOG_CTL[SNAP_CSITE_<p>] bit. That way CORESIGHT_CFG_CSITE_<p>_SNAP is updated with CORESIGHT_CFG_CSITE_<p>_LOG and hence, counts prior to this do not trigger an interrupt.

After programming the MASK bit, if value in CORESIGHT_CFG_CSITE_<p>_LOG exceeds value in CORESIGHT_CFG_CSITE_<p>_SNAP, an interrupt is generated. The processor handling the interrupt must program the CORESIGHT_CFG_EXT_DBG_LOG_CTL[SNAP_CSITE_<p>] bit to '1'. This causes CORESIGHT_CFG_CSITE_<p>_SNAP to be updated with CORESIGHT_CFG_CSITE_<p>_LOG value and thus clear the interrupt.

Finally, CORESIGHT_CFG_CSITE_<p>_LOG and CORESIGHT_CFG_CSITE_<p>_SNAP are compared for equality and not greater/less-than. Hence, overflow is not an issue and even if it is a concern, the 64-bit value overflows after more than 15 years even if DBGAPB is run at its maximum frequency of 136MHz.

Software is responsible to log such debug attempts to non-volatile media. Counters are reset on an L1warm reset. Hence, software is responsible for logging the information to non-volatile media if required for use later.

Boot ROM does not cause a chip reset request to be generated or turtle mode to be entered based on these sidebands from DFD. This is because invalid attempts may occur. To differentiate between innocent invalid attempts and intentional invalid attempts, software/OEM uses BCT bits to determine when to start checking for such attempts and what actions to take.

8.7.2.4 Watchdog for Debug and Recovery

Typical Watchdog Timers (WDTs) in the SoC have five expiration events. The 1st and 2nd expirations trigger IRQ and FIQ for the processor with which they are associated. The 3rd expiration is routed to the system LIC and can map to CCPLEX's (or any other processor e.g., SCE's) IRQ. For subsequent sections, it is assumed that LIC maps the 3rd expiration to CCPLEX. The 4th expiration typically triggers a Level-2 reset, while 5th expiration triggers a Level-1 reset.

Watchdogs can be used for debug or recovery. "Debug" here implies trying to understand what went wrong. "Recovery" here implies trying to get the processor to restart work, ignoring what went wrong.

The OS can either use a scheduled task to clear a watchdog timer periodically, or it could use the 1st expiration of the watchdog timer as interrupt with ISR to clear the watchdog. Reaching 2nd expiration is an indication that software was unable to clear the watchdog in an expected manner and should be an indication of system not behaving in a normal functional manner.

For debug purposes, upon 2nd expiration, the processor can store all its context in DRAM and wait for 3rd expiration. The 3rd expiration triggers an interrupt to CCPLEX, which can check which WDT's 3rd expiration triggered and store system (e.g., other BTCMs, SYSRAM, any other system context as required/accessible) context in DRAM. It can then wait for the 4th expiration from the tripping WDT. For debug purposes, when RAM dump is enabled, the Level-2 reset request from WDT/CSITESW/HSM causes the CCPLEX caches to be flushed, DRAM to be put in self-refresh, and then L2 reset is asserted. Post L2 reset, system boots until MB2 and DRAM contents can be scavenged by a tool on the host machine via talking to MB2 via XUSB.

For recovery purposes, upon 2nd expiration, the processor, if capable may reset itself. If unable to do so, for AUXP, it could rely on the CCPLEX to reset it upon 3rd expiration. Upon 4th expiration, RAM dump is disabled if the system intends to recover rather than debug. The system should be able to boot to kernel upon L2 reset. Kernel can extract and process the RAS related contents and eventually trigger a L1 reset (via SWMAINRST) causing the chip to reboot.

L1 reset resets most of the SoC. Boot ROM may choose to boot the chip or request PMIC to trigger an external reset to the chip based on the BR-PMIC structure loaded in SCRATCH by MB1.

There are various hardware and software components, and various exceptions/caveats can come into play. The subsequent sections provide information on what is present in hardware and recommendations for software on how to leverage the hardware features.

8.7.2.4.1 Various WDTs in the SoC

The table below gives an overview of various WDTs in the system and any caveats pertaining to them.

#	WDT (AMAP Aperture)	Owner/Use Case	Comments
1	TOP WDT0 (WDT0)	CCPLEX	
2	TOP WDT1 (WDT1)		
3	TOP WDT2 (WDT2)	APE (Cortex-A9)	
4	APE WDT (APE_TKE_WDT)	Do not use	4th and 5th expiration triggers are not connected to anything.
5	BPMP WDT (BPMP_TKE_WDT)	BPMP	
6	SPE WDT (AON_TKE_WDT)	SPE Probability of SPE WDT 4th/5th expiration as the cause of reset being low because SC7RF issues a CSITE-SW reset if wake from SC7 is due to SPE WDT's 3rd expiration. SC7RF stores the cause, i.e., SPE WDT expiration, in a scratch register for software to know if CSITE-SW reset was triggered as result of the above wake event.	1st expiration (drives IRQ) and 2nd expiration (drives FIQ) can potentially trigger wake (through aggregated IRQ and FIQ which are programmable wake events), although typically these are intended to be handled by SPE-FW. 3rd expiration is an AO-NPG WAKE event. 4th expiration is also an AO-NPG wake. However, this should not be used as a wake event.
7	AOWDT (RTC)	SPE AOWDT is a system resource and ownership may be shared or taken over by any other entity like BPMP/SCE/ CCPLEX based on software architecture or OEM/Platform specific needs when the chip is not in SC7 mode.	This should only be used for recovery. Always runs only on 32-kHz source. 2nd expiration triggers a WAKE event. 4th expiration does not trigger RAMDUMP even if RAMDUMP_EN=1. Thus, its 4th expiration always triggers L2 reset.
8	SCE WDT (SCE_TKE_WDT)	SCE	
9	RCE WDT (RCE_TKE_WDT)	RCE	

#	WDT (AMAP Aperture)	Owner/Use Case	Comments
10	DCE WDT (DCE_TKE_WDT)	DCE	
11	PVA0 WDT (PVA0_TKE_WDT)	PVA0 (Cortex-R5)	4th and 5th expiration triggers are not connected to anything.
12	-	-	-
13	FSI WDT (Total: 5 WDTs)	FSI	For each WDT: 1st and 2nd expirations routed to both, R52 and Cortex-R5. 1st expiration can be used for IRQ and 2nd expiration for FIQ for processor handling the WDT. 3rd expiration: refer to the *FSI TOP CTI Mappings* section. 4th expiration supports L2 reset (with RAMDUMP) and 5th expiration supports L1 reset.

The PMC_IMPL_RST_STATUS_0 can be used to determine the source of a reset.

WDT 5th expirations, for WDTs that can trigger L1 reset, can also trigger an external PMIC reset to the chip based on PMC_IMPL_RST_REQ_CONFIG_0. However, the platform PMIC must support this capability to accept a side band input for reset request.

8.7.2.5 Global Freezing for WDTs

8.7.2.5.1 Overview

There are two bits in the chip that can freeze all WDTs (subject to certain conditions described below).

PMC_MISC_WDT_FREEZE_REQ_0[FREEZE_REQ] is an MMIO register[bit] added to PMC, which can be used by software for freezing WDTs. This register is on L1cold reset with reset value of '0' ('UNFREEZE') and protected by an SCR.

There is also a creg bit (DC_MISC_CTL.wdt_disable). This is only accessible to MCE and MTS code owned by NVIDIA and hence, not usable by Arm software or other entities that can access MMIO registers. MCE/MTS's ability to drive the creg bit is qualified by the CPU DFD disable bit. The signal from creg is routed only to PMC, which is in AO power-domain. The signal should have a debug clamp and the clamped value for the creg signal should be '0'.

The PMC bit and the creg bit are OR'ed in PMC and the OR'ed output (dfd2all_wdt_freeze) drives the wdt-freeze input to all the WDTs. Additionally, all WDTs also use locally available debug-enable signal from DFD to qualify the wdt-freeze request. The wdt-freeze request signal is ignored if the local debug-enable is '0'. Each of the WDTs also have a software programmable bit

(DisallowWDTFreeze) that when set can cause the individual WDT to ignore wdt-freeze signal request. Refer to the Timers chapter for details.

#	WDT (AMAP Aperture)	Signal Driving WDT's DBGEN
1	TOP WDT0 (WDT0)	global_dbgen
2	TOP WDT1 (WDT1)	global_dbgen
3	TOP WDT2 (WDT2)	global_dbgen
4	APE WDT (APE_TKE_WDT)	ape_dbgen
5	BPMP WDT (BPMP_TKE_WDT)	bpmp_dbgen
6	SPE WDT (AON_TKE_WDT)	spe_dbgen
7	AOWDT (RTC)	spe_dbgen
8	SCE WDT (SCE_TKE_WDT)	sce_dbgen
9	RCE WDT (RCE_TKE_WDT)	rce_dbgen
10	DCE WDT (DCE_TKE_WDT)	dce_dbgen
11	PVA0 WDT (PVA0_TKE_WDT)	pva0_dbgen
-	-	-
13	FSI TKE WDT	dfd2fsi_cluster0_core0_dbgen & dfd2fsi_cluster1_core0_dbgen & dfd2fsi_cluster1_core1_dbgen & dfd2fsi_cluster1_core2_dbgen & dfd2fsi_cluster1_core3_dbgen

The local debug-enable should use a debug clamp while crossing from the SoC to AO and clamped value should be '0'.

Note: The signal from PMC to freeze all the WDTs is called dfd2all_freeze_wdt_req. Do not confuse it with the signal dfd2all_freeze_timers_req that is driven from DFD. dfd2all_freeze_timers_req can freeze timers (and WDTs if WDT's source is not TSC) whereas dfd2all_freeze_wdt_req can only freeze WDTs. The mask bit in TKE and related units, DisallowWDTFreeze, used to mask dfd2all_freeze_wdt_req is protected using the TKE's sticky safety protect bit – StickyEnabled. However, the StickyEnabled does not and must not protect (prevent writes) to the bit – HDBG that masks dfd2all_freeze_timers_req.

Most of the individual debug-enables are derived from a few common debug-enable signals. The status of most such debug enables can be inferred from CORESIGHT_CFG_DND_STATUS_0 and CORESIGHT_CFG_CUSTOMER_DFD_ALIVE_STATUS_0 (FSI related are in CORESIGHT_CFG_FSI_DBG_ALIVE_STATUS_0) and can help determine whether a specific WDT will be frozen after programming PMC_MISC_WDT_FREEZE_REQ_0[FREEZE_REQ] or DC_MISC_CTL[WDT_DISABLE].

8.7.2.5.2 Boot Impact

Boot/SC7RF software that programs the SCRs for *TKECR_0[HDBG] must allow AXI-AP access to modify the HDBG bit.

8.7.2.5.3 Reset and Timer Source for WDTs

All WDTs except SPE WDT (AON_TKE_WDT) and AOWDT (RTC) use L2warm reset.

SPE WDT (AON_TKE_WDT) uses L2cold reset. AOWDT (RTC) uses L1cold reset.

AOWDT is used to recover from hangs occurring during L2 reset. AOWDT expiration period must be long enough such that even if L2 reset occurs before AOWDT 3rd expiration and, hence, software that is expected to pet AOWDT is unable to do so, due to the large period, post L2 reset, chip can boot up to a point where Boot ROM executes and software entities that can handle AOWDT are loaded and are able to pet or restart the AOWDT. That is, the AOWDT's period between two expirations should be longer than entire duration of L2 reset (including preceding actions like cache flush, putting DRAM in self refresh, power sequencing, etc.) and subsequent Boot-SW execution (including time for steps like loading, authenticating, decrypting and executing sary binaries, system initialization, etc.). Thus, Boot ROM must not be required to pet AOWDT even if it is ticking post L2 reset. (Typically, for AOWDT, timeout is recommended to be configured as 10 seconds.)

Similarly, Boot ROM must not be required to pet SPE WDT post SC7 exit even though it is ticking across the warm reset. Duration of SPE-WDT must also be large enough to accommodate such boot post SC7 exit until software expects to pet or handle SPE-WDT expirations is able to do so without causing SPE-WDT to reach its next expiration before the software is able to service the prior expiration if required.The timer to be selected for AOWDT and SPE-WDT must be the always running TSC. Configuration options exist for selecting other timer sources, but software must refrain from selecting any other timer. This is because timers other than TSC stop counting on L2 reset assertion, while the TSC is on L1 reset.

8.7.2.5.4 AO-WDT

AO WDT is a system resource, not a cluster specific resource. AO-WDT serves as an in-chip alternative for external PMIC WDT. The external PMIC WDT can be disabled and the more convenient AO-WDT can be used if required.

AO WDT uses the 32-kHz clock and remains active in power state when OSC is stopped and/or OSC has issues and/or other WDT are rail gated.

For SC7 state, the SPE WDT can be used, as it also present in AON cluster. The AO WDT works during cases when OSC is shut off. AO WDT provides a new time-based SC7 wake signal, possibly easier to use than the RTC based alarms (RTC alarms are not periodic).

As AO-WDT is a system resource, its expirations are mapped as follows:

1st expiration: IRQ to SPE
2nd expiration:

- If system is in SC7 state, this triggers a WAKE.
- If system is not in SC7 state, this triggers interrupt (depending on wake tier config) to SPE VIC (Tier-0), BPMP VIC (Tier-1), or CCPLEX GIC (Tier-2).

3rd expiration: Does not trigger wake. Remote interrupt to LIC like other WDTs.
4th expiration: Do not enable.
5th expiration:

- If L1BAOWDTRST_WDT_OUT_RST_EN==1, toggles externally visible signal that can act as side-band to PMIC to request a chip reset.
- If L1BAOWDTRST_WDT_OUT_RST_EN==0, generates an L1 reset request.

8.7.2.5.5 Typical WDT Expiration Behavior

Unless specified otherwise, typical WDT expirations are as follows:

- 1st expiration – IRQ to processor owning the WDT
- 2nd expiration – FIQ to processor owning the WDT
- 3rd expiration – Remote interrupt to LIC
- 4th expiration – Triggers L2 reset request. If RAMDUMP is enabled, CCPLEX caches are flushed and DRAM is put in self-refresh. Then L2 reset occurs.
- 5th expiration – Triggers L1 reset request. If RAMDUMP_EN==1 && RAMDUMP_WDT_L1_DISABLE=0xA, the reset request is masked. Else, L1 reset occurs.

Note: HSM and CSITESW reset triggers an L2 reset with support for RAMDUMP too.

8.7.2.5.6 WDT Programming Guidelines

The previous subsections describe WDT expiration related features in hardware. This section provides guidance on how the features in hardware can be used by software for system Debug and Recovery. Software may choose alternate implementations based on specific circumstances.

- Upon L0/L1 reset, all WDTs are disabled.

ROMs

1. Boot ROM does not enable any WDT except for its local WDTs (i.e., BPMP WDT) when applicable (i.e., when its respective WDT enable fuses are burnt).
   a. Boot ROM enables BPMP WDT's 5th expiration based on fuse.
   b. During bringup, do not burn these fuses.
2. When Boot ROM enables its WDTs, they only enable the 5th expiration.
3. Boot ROM does not program the sticky bits in any WDT.

4. Boot ROM programs the WDT timeout such that expiration interval is at least 10s. If there is no expiration skipping, then as 5 expirations are encountered for reaching 5th expiration, each expiration timeout should be at least 2s. Thus, total duration to reach 5th expiration = 5 expirations x 2s/expiration = 10s. If skipping directly to 5th expiration, then timeout per expiration should be 10s, as only one expiration duration is encountered before actions associated with 5th expiration occur.
5. TSC does not run during Boot ROM phase. During BR, WDT must use the microsecond timer running off the OSC clock.
6. If it is a WDT/HSM/CSITESW L2 reset and RAMDUMP_EN=1, it is a L2-RAMDUMP Boot. 1. Boot ROM does not decide whether to regenerate or reuse the carveout keys based on SCRATCH_SECURE_RSV50_SCRATCH_0. This is delegated to MB1.

SC7RF

1. If it is a warm reset (SC7 exit) where the cause for the wake is SPE-WDT 3rd expiration and if RAMDUMP_EN=1, then Boot ROM does normal SC7 exit and passes execution to SC7RF. SC7RF logs the SPE WDT 3rd expiration wake event in an AO-SCRATCH register and issues a CSITE-SW L2 reset.

MB1 (with additional MCE guidelines)

1. MB1 programs PMC_MISC_DEBUG_AUTHENTICATION_0 and PMC_IMPL_RAMDUMP_CTL_STATUS2_0[RAMDUMP_EN] based on BCT.
2. Post-L2 reset, if ((PMC_IMPL_RAMDUMP_CTL_STATUS2_0[RAMDUMP_EN]==1) && (PMC_IMPL_DEBUG_AUTHENTICATION_0[SPIDEN]==0 || FUSE_DEBUG_AUTHENTICATION_0[SPIDEN_DIS]==1)), MB1 issues an L1 reset
3. MB1 decides whether to regenerate or reuse the MSS carveout keys based on BCT and NV-policy.

If (Preproduction mode)

{

Do not regenerate keys for any carveout. Instead, reuse/redistribute existing keys.

}

else

{

if ((ECID_Check==MATCH) & ((PMC_IMPL_RAMDUMP_CTL_STATUS2_0[RAMDUMP_EN]==1) &&

(PMC_IMPL_DEBUG_AUTHENTICATION_0[SPIDEN]==1 &

FUSE_DEBUG_AUTHENTICATION_0[2]==0)))

// FUSE_DEBUG_AUTHENTICATION_0[2] is fuse for spiden_dis

{

Regenerate and distribute MSS carveout keys (e.g., MTS, TZ, VPR, GSC) **based on BCT** (If BCT suggests don't regenerate the specific key(s), then don't regenerate specific key(s) and instead reuse/redistribute existing key(s).)

}

else

{

Regenerate and distribute all MSS carveout keys (e.g., MTS, TZ, VPR, GSC) independent of BCT

}

1. If ((RAMDUMP_EN==1) and (IST Boot**))
   a. {Do not set the sticky lock-bits (WDTCMDR_0[StickyStart]) in WDTs}
2. If (PMC_IMPL_RAMDUMP_CTL_STATUS2_0 [RAMDUMP_EN] ==1)
   a. { PMC_MISC_WDT_CFG_0 [RAMDUMP_WDT_L1_DISABLE] = 0xA }
3. If WDT/HSM/CSITESW-L2 Boot with RAMDUMP_EN=0, it is a L2-RAS Boot. Continue normal boot flow on path to boot to kernel. RAS-FW then stores RAS contents into Flash and issues an L1 reset.
4. If WDT/HSM/CSITESW-L2 Boot w/ RAMDUMP_EN==1 and (PMC_IMPL_DEBUG_AUTHENTICATION_0[SPIDEN]==1 && FUSE_DEBUG_AUTHENTICATION_0[SPIDEN_DIS]==0), it is an L2-RAMDUMP Boot.
5. If L2-RAMDUMP Boot, and DRAM is in self-refresh,
   a. MB1 disables ECC check based on BCT.
      i. IF (RAMDUMP-Boot && DRAM in self-refresh && BCT:ECC_CHECK_DISABLE_DURING_RAMDUMP), THEN, program ECC_CHECK_DISABLE=ON
   b. MB1 takes DRAM out of self-refresh
6. MB1 loads CPU-MB2. (**Note:** If possible, do not load SPE-FW during L2 boot with RAMDUMP enabled. That way, if L2 was post a SPE 3rd expiration in SC7, then, MB2 could potentially scavenge SPE related contents too.)

CPU-MB2

1. RAMDUMP logic to extract OEM carveouts from DRAM is part of MB2 binary.
2. If RAMDUMP boot is enabled, MB2 waits for USB host commands and dumps the OEM-DRAM carveout if RAMDUMP boot is enabled.
3. Commands from host-side to MB2:
   a. find cause of L2 reset
      i. For example, which WDT/HSM/CSITESW triggered L2-RAMDUMP and if CSITESW, then also contents of SCRATCH_SECURE_RSV106_SCRATCH_0 / SCRATCH_SPE_WDT_3RD_EXPIRY_DURING_SC7
      ii. SCRATCH assigned above must be cleared during L0/L1 reset but not during L2 reset
   b. query DRAM size
   c. dump DRAM contents from PA: X to Y
   d. dump ETF RAM contents
   e. continue boot
4. On receiving continue boot, MB2 provides reset of the boot.
5. RAS contents are extracted by later boot-stages by Secure Arm software.

BPMP-IST (This executes on the IST-Boot path):

```
{
 Do all the stuff that BPMP-IST needs to do before setting pre-ist-mode bit [No change]
 If (RAMDUMP_EN==1)
 {
 Disable 4th expiration of armed WDTs
 If (PMC_IMPL_RAMDUMP_CTL_STATUS_0[RAMDUMP_CSTATE]!=IDLE), spin in loop.
 }
 PMC_MISC_WDT_CFG_0[IST_L1L2_DISABLE] = 0xA
 Set the pre-ist-mode bit. [No change]
 Do all the stuff that BPMP-IST needs to do after setting pre-ist-mode bit [No change]
 }
```

Note: IST Boot refers to the Boot that occurs upon an L1 reset after which MB1 loads BPMP-IST instead of MB2.

Additional Guidelines on Handling WDT Expirations:

1st expiration:

- Upon 1st expiration of a WDT, IRQ for processor owning the WDT should restart corresponding WDT. (e.g., set WDTCMDR_0[StartCounter]=1)

2nd expiration:

- Upon 2nd expiration of the WDT, check if debug is enabled for the particular processor.

- If system debug has been enabled, store the cluster's state (e.g., CPU registers, local cache's, BTCMs, etc.) used by the processor in DRAM and wait for 3rd expiration.
- If system is in mission mode, software can attempt to recover the subsystem if security permits. For example, the processor can change its reset vector to the starting location of the firmware and request itself to be reset.

3rd expiration:

- This is routed as a remote interrupt to the LIC.
- If system debug has been enabled, CCPLEX can request the other processors to save their state in DRAM. It can also save its own state and any system state (e.g., SYSRAM) that can help debug into DRAM. Then it can wait for system to be reset via L2 reset.
- If the 3rd expiration occurred in SC7 state, then there are no contents from other processors to be stored. SC7RF logs the event in SCRATCH and issue CSITESW L2 reset (which also supports DFD-RAMDUMP).
- FSI 3rd expiration is not routed to LIC directly. However, as it is routed to R52 GIC and Cortex-R5 VIC, either of them can scavenge the state of the other, i.e., if FSI-R52 WDTs were tripping, FSI-R5 can scavenge R52's state it can access; or, if FSI-R5 WDT was tripping, FSI-R52 can scavenge Cortex-R5's state (as much as it has access to; for security reasons, all state may not be accessible; in which case any information helpful for debug may be logged). Additionally, FSI WDT 3rd expiration is also routed to FSI_TOP_CTI, from there it can trigger an interrupt to the processor that services WDT 3rd expiration LIC interrupts for rest of the SoC via cross-trigger network for it to scavenge and store and debug pertinent information.
- If system is in mission mode, CCPLEX may attempt to recover the cluster that reached 3rd expiration by resetting the cluster depending on the particular platforms security settings.

4th expiration:

- This triggers a L2 reset. See above on boot post L2 reset.

5th expiration:

- This triggers a L1 reset. Based on PMC_IMPL_RST_REQ_CONFIG_0, it may result in request to PMIC to trigger an external reset in which case, system post boot sees it as a L0 reset

Additional Guidelines on Programming WDTs (from Debug and Recovery Perspective)

The following table provides a broad recommendation on how WDTs should be configured across the chip.

WDT	Debug					Recovery				
expiration	1st	2nd	3rd	4th	5th	1st	2nd	3rd	4th	5th
TOP_WDT0	EN	EN	DIS	EN	DIS	EN	EN	DIS	EN	EN
TOP_WDT1	DIS	DIS	DIS	DIS	DIS	DIS	DIS	DIS	DIS	DIS
TOP_WDT2	EN	EN	EN	EN	DIS	EN	EN	EN	EN	EN

WDT	Debug					Recovery				
AOWDT	EN	EN	EN	DIS	DIS	EN	EN	EN	DIS	EN
SPE_WDT	EN	EN	EN	EN	DIS	EN	EN	EN	EN	EN
-	-	-	-	-	-	-	-	-	-	-
BPMP_WDT	EN	EN	EN	EN	DIS	EN	EN	EN	EN	EN
SCE_WDT	EN	EN	EN	EN	DIS	EN	EN	EN	EN	EN
RCE_WDT	EN	EN	EN	EN	DIS	EN	EN	EN	EN	EN
DCE_WDT	EN	EN	EN	EN	DIS	EN	EN	EN	EN	EN
PVA0_WDT	EN	EN	EN	DIS	DIS	EN	EN	EN	DIS	DIS
PVA1_WDT	EN	EN	EN	DIS	DIS	EN	EN	EN	DIS	DIS
APE_WDT	DIS	DIS	DIS	DIS	DIS	DIS	DIS	DIS	DIS	DIS
FSI_WDT	EN	EN	EN	EN	DIS	EN	EN	EN	EN	EN

Note: The gray cells must always be programmed as indicated.

Note: The WDT 3rd expiration routes an interrupt to Legacy Interrupt Controller (LIC). LIC can be programmed to route this interrupt to any processor (e.g., CCPLEX, BPMP, etc.). Also, the local interrupt controller for the processor (e.g., VIC, GIC, etc.) may choose to trigger IRQ/FIQ to the processor based on its configuration. The above table assumes the 3rd expiration events are routed by LIC to CCPLEX and hence, CCPLEX itself does not need to enable 3rd expiration for its own WDT. This applies for any other processor. (If BPMP were to be responsible for handling WDT 3rd expiries from other processors, the 3rd expiration for WDT owned by CCPLEX would be enabled and that for BPMP would be disabled).

In the above table, APE_WDT is shown as unused because APE is expected to use TOP_WDT2. TOP_WDT1 is current unused as shown above. If there is a viable use for it, it may be programmed based on the use case.

FSI WDT 3rd expiry is not routed to LIC. Instead, it is routed to FSI_R52_GIC, FSI_R5_VIC and FSI_TOP_CTI.

8.7.2.6 Debugging Low Power States

8.7.2.6.1 SC7 Debug Mode

In True SC7 state, most of the SoC and the entire CCPLEX and GPU are powered off. However, SPE processor and other logic in the Always ON (AON) cluster is powered and active. There could be

issues in the AON cluster that may need to be debugged using DFD components. However, DFD partition is powered by the VDD_CORE power rail, which is typically shut down in True SC7 state.

To debug components in the AON cluster using DFD components in VDD_CORE domain, a new state was created – SC7 Debug Mode.

In this SC7 Debug state, the system requirements are as follows:

- VDD_CORE rail must remain powered ON throughout SC7 transition.
  - **Note:** Throughout SC7 transition implies: during SC7 debug mode entry, during SC7 debug mode execution, and during SC7 debug mode exit.
  - The voltage on the VDD_CORE rail, when it is ON must be sufficient for SoC operation.
- DFD partition is not reset at any time throughout SC7 transition.
- The following signals are not clamped and transmit their values in normal functional manner.
  - From/to DFD to/from AON cluster.
  - From Fuse to DFD cluster.
  - **Note:** Signals to the AON cluster from the rest of the SoC (including clusters like CCPLEX, GPU, etc.) should be clamped as per their regular SC7 clamp values. This ensures that for the logic in the AON partition, the SoC appears similar to how it would during a True SC7 state.
- To support SC7 Debug Mode in Open Box Debug environment (i.e. over ArmJTAG), there are additional requirements:
  - JTAG pads/IOs are functional and can perform JTAG transactions.
  - Signals from/to JTAG pins to/from DFD partition are not clamped and transmit their values in normal functional manner.
- To support SC7 Debug Mode in Closed Box Debug environment (i.e., over SWD-DFD), there are additional requirements:
  - USB2 pads/IOs and padmacro are functional and can operate USB2 pins in SWD mode.
  - Signals from/to USB2 pins to/from DFD partition are not clamped and transmit their values in normal manner.

SC7 Debug Flow Details

- Program PMC_IMPL_SC7_DEBUG_CTRL_0[GLITCH_FREE_CLKEN]=ENABLE while enabling SC7 Debug mode to avoid clock glitch issues during SC7 Debug exit.
- Program PMC_IMPL_SC7_DEBUG_CTRL_0[DFD_CLAMP_OVERRIDE]=1 to avoid DFD signals from being clamped during SC7 Debug mode.
- Retain PMC_IMPL_SC7_DEBUG_CTRL_0[MAIN_CLAMP_DEBUG]=0 to ensure non-debug related signals are clamped during SC7 Debug mode.
- Program PMC_IMPL_SC7_DEBUG_CTRL_0[DBGRST_OVR]=1 to avoid DFD from being reset during SC7 Debug exit.
- Retain PMC_IMPL_SC7_DEBUG_CTRL_0[RESET_DEBUG]=0 to ensure warm reset is asserted during SC7 Debug exit.

- Program PMC_IMPL_SC7_DEBUG_CTRL_0[DPD_ENABLE_DEBUG]=1 to ensure pads do not enter Deep-Power-Down (DPD) state during SC7 Debug mode.
- Program CLK_RST_CONTROLLER_AON_CPU_PGRST_CTRL_0[AON_PG_EMULATE]=1
  - Skipping this programming step causes debug-logic in AO-PG to be held in reset when AO-PG is power-gated in SC7 Debug mode.
  - **Note:** If ACCESS BLOCK/TIMEOUT features in DFD are not enabled, a debug-APB access to a cluster held in reset will hang.
- Program PMC_IMPL_CNTRL_0[PWRREQ_OE]=1 to ensure VDD_CORE remains powered on during SC7 Debug.

- Clocks
  - OSC may be turned off in SC7 mode. Hence, 32 kHz is available as a source for DBGAPB and CSITE, and LA clock muxes.

8.7.3 DFD Registers

Refer to Reading Register Tables in the Introduction section for the register table protocol as well as recommendations for accessing registers.

8.7.3.1 CoreSight_CFG Registers

For base address of this block of registers, please refer to CSITE in the System Address Map.

CORESIGHT_CFG_CTL_0

CoreSight Control Register
This register can be used to configure various debug features of CoreSight.

Offset: 0x20000
Read/Write: R/W
Parity Protection: N
Reset: 0x00010101 (0bxxxx,xxxx,xxxx,xxx1,xxxx,xx01,xxxx,xxx1)

Bit	Reset	Description
16	0x1	**RTCK_SPEED**: RTCK delay (Synchronized or Direct). This is used to select between the early and late rtck from ARM7 RTCK is delayed by one synchronizer when this bit is enabled and set to 2 otherwise. The additional synchronizer was added to support enabling RTCK on legacy SoC FPGA. **0** = SLOW **1** = FAST
9	0x0	**TIMEOUT_VAL**: Choose a small timeout versus a long timeout. The timeouts are fixed and not changeable The short timeout is to reduce the duration of system verification simulations. **1** = SHORT_TIMEOUT **0** = LONG_TIMEOUT
8	0x1	**TIMEOUT_EN**: Enable TimeOut for an Access on the Debug APB Bus to CCPLEX, APE, iLA. Each Master port emanating from CoreSight Debug APB to external world has a timeout. This timeout is triggered on each psel generated. If pready is not received within the duration configured by the TIMEOUT_VAL register, the timer associated with the timeout expires resulting in a deadcode on the corresponding prdata of the Debug APB bus with the following values yet completing the access gracefully with a pready of 1. 0xdead0002 - CCPLEX Cluster timed out. 0xdead0004 - APE Debug APB timed out. 0xdead0006 - LA Debug APB timed out. 0xdead0008 - BPMP Debug APB timed out. 0xdead000A - SPE Debug APB timed out. 0xdead000C - SCE Debug APB timed out. 0xdead000E - RCE Debug APB timed out. 0xdead0010 - DCE Debug APB timed out.0xdead0041 - PVA Collator RomTable / Funnel Debug APB Timed Out 0xdead0043 - PVA0-R5PIL Debug APB Timed Out 0xdead0045 - PVA0-VPS0 Debug APB Timed Out 0xdead0047 - PVA0-VPS1 Debug APB Timed Out 0xdead0049 - PVA1-R5PIL Debug APB Timed Out 0xdead004B - PVA1-VPS0 Debug APB Timed Out 0xdead004D - PVA1-VPS1 Debug APB Timed Out 0xdead004F - PVA0 Rom Table Timed Out 0xdead0051 - PVA1 Rom Table Timed Out On receiving the deadcode, the respective ALIVE STATUS registers can be queried to determine the exact cause of no-response **1** = ENABLE **0** = DISABLE

Bit	Reset	Description
0	0x1	**BLOCK_ACCESS_EN**: Block accesses to non-responsive CPUs over the Debug APB The accesses complete gracefully with a deadcode value as below and pready asserted as 1 0xdead0003 - APE is non responsive. 0xdead0005 - LA is non responsive. 0xdead0007 - BPMP is non responsive. 0xdead0009 - SPE is non responsive. 0xdead000B - SCE is non responsive. 0xdead000D - RCE is non responsive. 0xdead0011 - DCE is non responsive.0xdead0040 - PVA Collator RomTable / Funnel is non responsive 0xdead0042 - PVA0- R5PIL is non responsive 0xdead0044 - PVA0-VPS0 is non responsive 0xdead0046 - PVA0-VPS1 is non responsive 0xdead0048 - PVA1- R5PIL is non responsive 0xdead004A - PVA1-VPS0 is non responsive 0xdead004C - PVA1-VPS1 is non responsive 0xdead004E - PVA0 Rom Table is non responsive 0xdead0050 - PVA1 Rom Table is non responsive On receiving the deadcode, the respective ALIVE STATUS registers can be queried to determine the exact cause of no-response. Following are the cases when the accesses are blocked CCPLEX : CCPLEX rail gated OR NONCPU of CCPLEX power gated CCPLEX has its PRESETDBG asserted Note: Accesses not gated for clock gating of CPU clock as the ROM tables live on the Debug APB clk APE : APE powergated together with its ROM table APE clock gated Note : Accesses not gated for APE under reset as ROM table logic can still work via the CoreSight reset LA : LA clk gated LA under reset Note : Accesses not gated for Power gating as CoreSight and LA on same Power domain. **1** = ENABLE **0** = DISABLE

CORESIGHT_CFG_EXTERNAL_DEBUG_REQ_0

CoreSight External Debug Request Register

Each ARM CPU implements an edbreq active high level input. This input is one of the many Halting Debug Events that can make CPU enter into Debug State. A high value on edbreq signals the CPU to enter into debug state. Clearing these bits does NOT restart the processor, but the request needs to be cleared before the debugger can restart the processor.

Offset: 0x20004
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0bxxx0,0000,0000,0000,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
28	-	-
27	0x0	**EDBGRQ_DCE**: External Debug Request for DCE **1** = ENABLE **0** = DISABLE
26	0x0	**EDBGRQ_PVA1_VPU1**: External Debug Request for PVA1_VPU1 **1** = ENABLE **0** = DISABLE
25	0x0	**EDBGRQ_PVA1_VPU0**: External Debug Request for PVA1_VPU0 **1** = ENABLE **0** = DISABLE
24	0x0	**EDBGRQ_PVA1_R5**: External Debug Request for PVA1_R5 **1** = ENABLE **0** = DISABLE
23	0x0	**EDBGRQ_PVA0_VPU1**: External Debug Request for PVA0_VPU1 **1** = ENABLE **0** = DISABLE
22	0x0	**EDBGRQ_PVA0_VPU0**: External Debug Request for PVA0_VPU0 **1** = ENABLE **0** = DISABLE
21	0x0	**EDBGRQ_PVA0_R5**: External Debug Request for PVA0_R5 **1** = ENABLE **0** = DISABLE
20	0x0	**EDBGRQ_RCE**: External Debug Request for RCE **1** = ENABLE **0** = DISABLE

Bit	Reset	Description
19	0x0	**EDBGRQ_SCE**: External Debug Request for SCE **1** = ENABLE **0** = DISABLE
18	0x0	**EDBGRQ_SPE**: External Debug Request for SPE **1** = ENABLE **0** = DISABLE
17	0x0	**EDBGRQ_APE**: External Debug Request for APE **1** = ENABLE **0** = DISABLE
16	0x0	**EDBGRQ_BPMP**: External Debug Request for BPMP **1** = ENABLE **0** = DISABLE
4	0x0	**EDBGRQ_FSI_R5**: External Debug Request for FSI Cortex-R5 **1** = ENABLE **0** = DISABLE
3	0x0	**EDBGRQ_FSI_R52_CORE3**: External Debug Request for FSI R52 Core3 **1** = ENABLE **0** = DISABLE
2	0x0	**EDBGRQ_FSI_R52_CORE2**: External Debug Request for FSI R52 Core2 **1** = ENABLE **0** = DISABLE
1	0x0	**EDBGRQ_FSI_R52_CORE1**: External Debug Request for FSI R52 Core1 **1** = ENABLE **0** = DISABLE
0	0x0	**EDBGRQ_FSI_R52_CORE0**: External Debug Request for FSI R52 Core0 **1** = ENABLE **0** = DISABLE

CORESIGHT_CFG_DEBUG_ACK_0

CoreSight Debug Acknowledge Status Register

Each CPU asserts an active high Debug Acknowledge signal when in Debug State The Debug State may have been entered by any one of the Debug Entry mechanisms Halting Debug Event like: EDBGREQ, Halt Req via DRCR Halt req bit Software Debug Event for example, BKPT, Breakpoint Debug event, etc. The ack bits are set for each of these mechanisms.

Offset: 0x20008
Read/Write: RO
Parity Protection: N
Reset: 0x00000000 (0bxxx0,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
28	-	-
27	0x0	**DBGACK_DCE**: Debug Acknowledge from DCE **1** = ENABLE **0** = DISABLE
26	0x0	**DBGACK_PVA1_VPU1**: Debug Acknowledge from PVA1_VPU1 **1** = ENABLE **0** = DISABLE
25	0x0	**DBGACK_PVA1_VPU0**: Debug Acknowledge from PVA1_VPU0 **1** = ENABLE **0** = DISABLE
24	0x0	**DBGACK_PVA1_R5**: Debug Acknowledge from PVA1_R5 **1** = ENABLE **0** = DISABLE
23	0x0	**DBGACK_PVA0_VPU1**: Debug Acknowledge from PVA0_VPU1 **1** = ENABLE **0** = DISABLE
22	0x0	**DBGACK_PVA0_VPU0**: Debug Acknowledge from PVA0_VPU0 **1** = ENABLE **0** = DISABLE
21	0x0	**DBGACK_PVA0_R5**: Debug Acknowledge from PVA0_R5 **1** = ENABLE **0** = DISABLE
20	0x0	**DBGACK_RCE**: Debug Acknowledge from RCE **1** = ENABLE **0** = DISABLE

Bit	Reset	Description
19	0x0	**DBGACK_SCE**: Debug Acknowledge from SCE **1** = ENABLE **0** = DISABLE
18	0x0	**DBGACK_SPE**: Debug Acknowledge from SPE **1** = ENABLE **0** = DISABLE
17	0x0	**DBGACK_APE**: Debug Acknowledge from APE **1** = ENABLE **0** = DISABLE
16	0x0	**DBGACK_BPMP**: Debug Acknowledge from BPMP **1** = ENABLE **0** = DISABLE
5	0x0	**DBGACK_FSI_R5**: Debug Acknowledge from FSI Cortex-R5 **1** = ENABLE **0** = DISABLE
4	0x0	**DBGACK_FSI_R52_CORE3**: Debug Acknowledge from FSI R52 CORE3. **1** = ENABLE **0** = DISABLE
3	0x0	**DBGACK_FSI_R52_CORE2**: Debug Acknowledge from FSI R52 CORE2. **1** = ENABLE **0** = DISABLE
2	0x0	**DBGACK_FSI_R52_CORE1**: Debug Acknowledge from FSI R52 CORE1. **1** = ENABLE **0** = DISABLE
1	0x0	**DBGACK_FSI_R52_CORE0**: Debug Acknowledge from FSI R52 CORE0. **1** = ENABLE **0** = DISABLE
0	0x0	**DBGACK_CLUSTER0_CPU0**: Debug Acknowledge from CLUSTER0_CPU0 - Aggregated DBGACK from CCPLEX. **1** = ENABLE **0** = DISABLE

CORESIGHT_CFG_APE_PROCESSOR_ALIVE_STATUS_0

For additional clusters
CoreSight Service Processor Alive Status Register.

Respective RESET fields in ALIVE_STATUS regs are set when particular reset is asserted This register provides the CPU Clk gated (in CAR), clamped (from PMC), and reset status (from CAR) for main CPU Also provides Debug APB timeout status.

Offset: 0x20010
Read/Write: See table below
Parity Protection: N
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
3	RW	X	**APE_DBGAPB_TIMEOUT**: Debug APB Bus timed out for APE **1** = TIMEOUT **0** = NORMAL
2	RO	X	**APE_CLAMPED**: clamp status of APE **1** = CLAMPED **0** = UNCLAMPED
1	RO	X	**APE_CLK_GATED**: clk gated status of APE **1** = CLKGATED **0** = CLKUNGATED
0	RO	X	**APE_RESET**: reset status of APE **1** = RESET_ASSERTED **0** = RESET_DEASSERTED

CORESIGHT_CFG_ARM_JTAG_MODES_0

CoreSight ARM_JTAG Mode Register

Offset: 0x20018
Read/Write: R/W
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01)

Bit	Reset	Description
1:0	0x1	**ARM_JTAG_MODE**: Only Coresight DAP is in the ARM JTAG chain (only 0x1 value is valid). **1** = CORESIGHT_ONLY

CORESIGHT_CFG_ARM_JTAG_STATUS_0

CoreSight JTAG Status Register

Offset: 0x2001c
Read/Write: RO
Parity Protection: N
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
0	X	**JTAGTOP**: indicates JTAG-DP TAP controller is in one of the following states Test-Logic-Reset Run-Test/Idle Select-DR-Scan Select-IR-Scan **1** = JTAG_IN_USE **0** = JTAG_NOT_IN_USE

CORESIGHT_CFG_DEBUG_ACK_QUAL_0

CoreSight Debug Acknowledge Qualification Register

When the corresponding bit is set, the corresponding DBGACK in CORESIGHT_CFG_DEBUG_ACK_0 participates in the timer freeze else the effect of that DBGACK is ignored.

Offset: 0x20020

Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0bxxx0,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
28	-	-
27	0x0	**DBGACK_DCE**: Debug Acknowledge qualifier for DCE **1** = ENABLE **0** = DISABLE

Bit	Reset	Description
26	0x0	**DBGACK_PVA1_VPU1**: Debug Acknowledge qualifier for PVA1_VPU1 **1** = ENABLE **0** = DISABLE
25	0x0	**DBGACK_PVA1_VPU0**: Debug Acknowledge qualifier for PVA1_VPU0 **1** = ENABLE **0** = DISABLE
24	0x0	**DBGACK_PVA1_R5**: Debug Acknowledge qualifier for PVA1_R5 **1** = ENABLE **0** = DISABLE
23	0x0	**DBGACK_PVA0_VPU1**: Debug Acknowledge qualifier for PVA0_VPU1 **1** = ENABLE **0** = DISABLE
22	0x0	**DBGACK_PVA0_VPU0**: Debug Acknowledge qualifier for PVA0_VPU0 **1** = ENABLE **0** = DISABLE
21	0x0	**DBGACK_PVA0_R5**: Debug Acknowledge qualifier for PVA0_R5 **1** = ENABLE **0** = DISABLE
20	0x0	**DBGACK_RCE**: Debug Acknowledge qualifier for RCE **1** = ENABLE **0** = DISABLE
19	0x0	**DBGACK_SCE**: Debug Acknowledge qualifier for SCE **1** = ENABLE **0** = DISABLE
18	0x0	**DBGACK_SPE**: Debug Acknowledge qualifier SPE **1** = ENABLE **0** = DISABLE
17	0x0	**DBGACK_APE**: Debug Acknowledge qualifier for APE **1** = ENABLE **0** = DISABLE

Bit	Reset	Description
16	0x0	**DBGACK_BPMP**: Debug Acknowledge qualifier for BPMP **1** = ENABLE **0** = DISABLE
5	0x0	**DBGACK_FSI_R5**: Debug Acknowledge from FSI Cortex-R5 **1** = ENABLE **0** = DISABLE
4	0x0	**DBGACK_FSI_R52_CORE3**: Debug Acknowledge from FSI R52 CORE3. **1** = ENABLE **0** = DISABLE
3	0x0	**DBGACK_FSI_R52_CORE2**: Debug Acknowledge from FSI R52 CORE2. **1** = ENABLE **0** = DISABLE
2	0x0	**DBGACK_FSI_R52_CORE1**: Debug Acknowledge from FSI R52 CORE1. **1** = ENABLE **0** = DISABLE
1	0x0	**DBGACK_FSI_R52_CORE0**: Debug Acknowledge from FSI R52 CORE0. **1** = ENABLE **0** = DISABLE
0	0x0	**DBGACK_CLUSTER0_CPU0**: Debug Acknowledge qualifier for CLUSTER0_CPU0 - Aggregated qualifier for CCPLEX DBGACK. **1** = ENABLE **0** = DISABLE

CORESIGHT_CFG_CHIPID_0

CoreSight Chip ID Register

This R/W register can be used to ensure that any CPU, APBAP and AXIAP are able to make an access to the system. Debugger is expected to first write and then read this register with a known CHIPID value - after halting the CPU or connecting to an Access Port (APB-AP or AXI-AP) When the read and write data is the same, then the corresponding chipid can be printed. This indicates that the connection with CPU/Access Port is good to use.

CHIPIDs are:

1. `cluster 0 cpu 0 - chipid = 0xbcbc0000`

2. cluster 0 cpu 1 - chipid = 0xbcbc0001
3. cluster 1 cpu 0 - chipid = 0xbcbc0010
4. cluster 1 cpu 1 - chipid = 0xbcbc0011
5. cluster 2 cpu 0 - chipid = 0xbcbc0020
6. cluster 2 cpu 1 - chipid = 0xbcbc0021
7. cluster 3 cpu 0 - chipid = 0xbcbc0030
8. cluster 3 cpu 1 - chipid = 0xbcbc0031
9. bpmp - chipid = 0xbbbbbbbb (b= bpmp)
10. ape   - chipid = 0xaaaaaaaa (a = ape)
11. spe   - chipid = 0x5e5e5e5e (5e = spe, see 5 as "S")
12. sce - chipid = 0x5c5c5c5c (5c = sce, see 5 as "S" and c for camera which is the non-automotive use case)
13. axiap - coresight axiap. chipid = 0xcacacaca (c= coresight a=axiap)
14. dapb - corsight debug apb (0xcdcdcdcd) (c=coresight d=debug apb)

Offset: 0x20024
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CHIPID_VAL**

CORESIGHT_CFG_CORESIGHT_SOC400_CLK_OVR_ON_0

Software accessible IP master override bit for all SLCGs in the IP

Offset: 0x2002c
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**OVERRIDE_BIT**

CORESIGHT_CFG_SW_RESET_REQ_0

DFD software based L2 reset request sideband from CoreSight to PMC.

Offset: 0x20038
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**L2_RST_REQ**: **1** = ASSERT **0** = DEASSERT

CORESIGHT_CFG_BPMP_R5_PROCESSOR_ALIVE_STATUS_0

BPMP Cortex-R5 Processor Alive Status Register

This register provides the BPMP Clk gated (in CAR), clamped (from PMC), and reset status (from CAR). Also provides Debug APB timeout status. The reset signals and clken are of the same name in RTL as the fields in this register Polarity of Reset signals (*RSTN*, *SRSTN*) is active high (status signals).

Offset: 0x2003c
Read/Write: See table below
Parity Protection: N
Reset: 0x0000XXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,0xxx,xxxx,xxxx)

Bit	R/W	Reset	Description
14	RO	X	**COUNTER_TOGGLE**
13	RO	X	**NWFIPIPESTOPPED**
12	RO	X	**NCLKSTOPPED**
11	RO	0x0	**ATCERROR**: This bit is tied to 0 in BPMP.
10	RO	X	**ROM_SLEEP_EN**: indicates "bpmp_cpu_lockstep_error_pulse" OR "rom_sleep_en" status.
9	RO	X	**ACLKEN_TOGGLE**
8	RO	X	**CAR2SOC_PREBOOTL2WARM_RSTN_DEBUG**
7	RO	X	**CAR2BPMP_BOOTL2WARM_RSTN_DEBUG**
6	RO	X	**BPMP_PRESETDBGN_DBGAPB_CLK_SRSTN**
5	RO	X	**DBGRESETN_BPMP_CLK_SRSTN**
4	RO	X	**NRESET_BPMP_CLK_SRSTN**

Bit	R/W	Reset	Description
3	RW	X	**BPMP_DBGAPB_TIMEOUT**: Debug APB Bus timed out for BPMP **1** = TIMEOUT **0** = NORMAL
2	RO	X	**BPMP_CLAMPED**: clamp status of BPMP **1** = CLAMPED **0** = UNCLAMPED
1	RO	X	**BPMP_ACLKEN**
0	RO	X	**NSYSPORRESET_BPMP_CLK_SRSTN**

CORESIGHT_CFG_SPE_R5_PROCESSOR_ALIVE_STATUS_0

SPE Cortex-R5 Processor Alive Status Register

This register provides the SPE Clk gated (in CAR), clamped (from PMC), and reset status (from CAR). Also provides Debug APB timeout status. The reset signals and clken are of the same name in RTL as the fields in this register Polarity of Reset signals (*SRSTN*) is active high (status signals).

Offset: 0x20040
Read/Write: See table below
Parity Protection: N
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
6	RO	X	**AO_PRESETDBGN_DBGAPB_CLK_SRSTN**
5	RO	X	**DBGRESETN_AON_CPU_CLK_SRSTN**
4	RO	X	**NRESET_AON_CPU_CLK_SRSTN**
3	RW	X	**SPE_DBGAPB_TIMEOUT**: Debug APB Bus timed out for SPE **1** = TIMEOUT **0** = NORMAL
2	RO	X	**SPE_CLAMPED**: clamp status of SPE **1** = CLAMPED **0** = UNCLAMPED
1	RO	X	**AON_ACLKEN**
0	RO	X	**NSYSPORRESET_AON_CPU_CLK_SRSTN**

CORESIGHT_CFG_SCE_R5_PROCESSOR_ALIVE_STATUS_0

SCE Cortex-R5 Processor Alive Status Register

This register provides the SCE Clk gated (in CAR), clamped (from PMC), and reset status (from CAR). Also provides Debug APB timeout status. The reset signals and clken are of the same name in RTL as the fields in this register Polarity of Reset signals (*SRSTN*) is active high (status signals).

Offset: 0x20044
Read/Write: See table below
Parity Protection: N
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
6	RO	X	**SCE_PRESETDBGN_DBGAPB_CLK_SRSTN**
5	RO	X	**DBGRESETN_SCE_CLK_SRSTN**
4	RO	X	**NRESET_SCE_CLK_SRSTN**
3	RW	X	**SCE_DBGAPB_TIMEOUT**: Debug APB Bus timed out for SCE **1** = TIMEOUT **0** = NORMAL
2	RO	X	**SCE_CLAMPED**: clamp status of SCE **1** = CLAMPED **0** = UNCLAMPED
1	RO	X	**SCE_ACLKEN**
0	RO	X	**NSYSPORRESET_SCE_CLK_SRSTN**

CORESIGHT_CFG_RCE_R5_PROCESSOR_ALIVE_STATUS_0

RCE Cortex-R5 Processor Alive Status Register

This register provides the RCE Clk gated (in CAR), clamped (from PMC), and reset status (from CAR). Also provides Debug APB timeout status. The reset signals and clken are of the same name in RTL as the fields in this register Polarity of Reset signals (*SRSTN*) is active high (status signals).

Offset: 0x20048
Read/Write: See table below
Parity Protection: N
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
6	RO	X	**RCE_PRESETDBGN_DBGAPB_CLK_SRSTN**
5	RO	X	**DBGRESETN_RCE_CLK_SRSTN**
4	RO	X	**NRESET_RCE_CLK_SRSTN**
3	RW	X	**RCE_DBGAPB_TIMEOUT**: Debug APB Bus timed out for RCE **1** = TIMEOUT **0** = NORMAL
2	RO	X	**RCE_CLAMPED**: clamp status of RCE **1** = CLAMPED **0** = UNCLAMPED
1	RO	X	**RCE_ACLKEN**
0	RO	X	**NSYSPORRESET_RCE_CLK_SRSTN**

CORESIGHT_CFG_DCE_R5_PROCESSOR_ALIVE_STATUS_0

DCE Cortex-R5 Processor Alive Status Register This register provides the DCE Clk gated (in CAR), clamped (from PMC), and reset status (from CAR). Also provides Debug APB timeout status. The reset signals and clken are of the same name in RTL as the fields in this register Polarity of Reset signals (*SRSTN*) is active high (status signals).

Offset: 0x2004c
Read/Write: See table below
Parity Protection: N
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
6	RO	X	**DCE_PRESETDBGN_DBGAPB_CLK_SRSTN**
5	RO	X	**DBGRESETN_DCE_CLK_SRSTN**
4	RO	X	**NRESET_DCE_CLK_SRSTN**
3	RW	X	**DCE_DBGAPB_TIMEOUT**: Debug APB Bus timed out for DCE **1** = TIMEOUT **0** = NORMAL
2	RO	X	**DCE_CLAMPED**: clamp status of DCE **1** = CLAMPED **0** = UNCLAMPED

Bit	R/W	Reset	Description
1	RO	X	**DCE_ACLKEN**
0	RO	X	**NSYSPORRESET_DCE_CLK_SRSTN**

CORESIGHT_CFG_PVA0_R5_PROCESSOR_ALIVE_STATUS_0

PVA0 Cortex-R5 Processor Alive Status Register

This register provides the PVA Clk gated (in CAR), clamped (from PMC), and reset status (from CAR). Also provides Debug APB timeout status. The reset signals and clken are of the same name in RTL as the fields in this register Polarity of Reset signals (*RSTN*, *SRSTN*) is active high (status signals).

Offset: 0x20050
Read/Write: See table below
Parity Protection: N
Reset: 0x00000XXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
9	RO	X	**PVAA_CLAMPED**
8	RO	X	**PVA0_VPS1_RESET_RSTN**
7	RO	X	**PVA0_VPS0_RESET_RSTN**
6	RO	X	**PVA0_PRESETDBGN_VPS1_RSTN**
5	RO	X	**PVA0_PRESETDBGN_VPS0_RSTN**
4	RO	X	**PVA0_PRESETDBGN_DBGAPB_CLK_SRSTN**
3	RO	X	**DBGRESETN_PVA0_CPU_CLK_SRSTN**
2	RO	X	**NRESET_PVA0_CPU_CLK_SRSTN**
1	RW	X	**PVA0_DBGAPB_TIMEOUT**: Debug APB Bus timed out for PVA **1** = TIMEOUT **0** = NORMAL
0	RO	X	**NSYSPORESET_PVA0_CPU_CLK_SRSTN**

CORESIGHT_CFG_PVA1_R5_PROCESSOR_ALIVE_STATUS_0

PVA1 Cortex-R5 Processor Alive Status Register

This register provides the PVA Clk gated (in CAR), clamped (from PMC), and reset status (from CAR). Also provides Debug APB timeout status. The reset signals and clken are of the same name in RTL as the fields in this register Polarity of Reset signals (*RSTN*, *SRSTN*) is active high (status signals).

Offset: 0x20054
Read/Write: See table below
Parity Protection: N
Reset: 0x00000XXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
9	RO	X	**PVAB_CLAMPED**
8	RO	X	**PVA1_VPS1_RESET_RSTN**
7	RO	X	**PVA1_VPS0_RESET_RSTN**
6	RO	X	**PVA1_PRESETDBGN_VPS1_RSTN**
5	RO	X	**PVA1_PRESETDBGN_VPS0_RSTN**
4	RO	X	**PVA1_PRESETDBGN_DBGAPB_CLK_SRSTN**
3	RO	X	**DBGRESETN_PVA1_CPU_CLK_SRSTN**
2	RO	X	**NRESET_PVA1_CPU_CLK_SRSTN**
1	RW	X	**PVA1_DBGAPB_TIMEOUT**: Debug APB Bus timed out for PVA **1** = TIMEOUT **0** = NORMAL
0	RO	X	**NSYSPORESET_PVA1_CPU_CLK_SRSTN**

CORESIGHT_CFG_CVPVA_PROCESSOR_ALIVE_STATUS_0

PVA1 Cortex-R5 Processor Alive Status Register

This register provides the PVA Clk gated (in CAR), clamped (from PMC), and reset status (from CAR). Also provides Debug APB timeout status. The reset signals and clken are of the same name in RTL as the fields in this register Polarity of Reset signals (*RSTN*, *SRSTN*) is active high (status signals).

Offset: 0x20058
Read/Write: RO
Parity Protection: N
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
0	X	**CVRG_CLAMPED**

CORESIGHT_CFG_SYSTEM_CTL_0

Register to force PADDRDBG31 to 1 for the top level APBIC master ports.

Offset: 0x20064
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**FORCE_PADDRDBG31**: This allows an internal access to come up as an external debugger access. **1** = ENABLE **0** = DISABLE

CORESIGHT_CFG_CSITE_MC_WR_CTRL_0

Offset: 0x20068
Read/Write: See table below
Parity Protection: N
Reset: 0x000X0X00 (0bxxx0,0000,0000,00xx,0000,0x00,0000,0000)

Bit	R/W	Reset	Description
28:26	RW	0x0	**USER_SIZE**
25:23	RW	0x0	**RSDV**
22	RW	0x0	**SP_AWARE**
21:19	RW	0x0	**SUB_PART_ADD**: Sub partition Address { sp_aware , user_adr1[8:6], user_adr[5]}
18	RW	0x0	**USER_ADR_5**
17:16	RO	X	**GSC_AL**: Hardcoded to 0, General Security Carveout Access { GSC_AL[1:0]}
15:11	RW	0x0	**GSC_AID**: General Security Carveout Aperture ID GSC_AID[4:0]

Bit	R/W	Reset	Description
10	RO	X	**VPR**: Hardcoded to 0, VPR setting (sc2mc_vpr_wr) 1'b1: Write should not be allowed outside the VPR aperture 1'b0: Writes should be allowed anywhere in the memory (including VPR region).
9	RW	0x0	**WSB_NS**: AWPROT
8	RW	0x0	**COH**
7:0	RW	0x0	**CSITE_SID**: CoreSight AXIAP StreamID for write transactions

CORESIGHT_CFG_CSITE_MC_RD_CTRL_0

Offset: 0x2006c
Read/Write: See table below
Parity Protection: N
Reset: 0x000X0X00 (0bxxx0,0000,0000,00xx,0000,0x00,0000,0000)

Bit	R/W	Reset	Description
28:26	RW	0x0	**USER_SIZE**
25:23	RW	0x0	**RSDV**
22	RW	0x0	**SP_AWARE**
21:19	RW	0x0	**SUB_PART_ADD**: Sub partition Address { sp_aware , user_adr1[8:6], user_adr[5]}
18	RW	0x0	**USER_ADR_5**
17:16	RO	X	**GSC_AL**: Hardcoded to 0, General Security Carveout Access { GSC_AL[1:0]}
15:11	RW	0x0	**GSC_AID**: General Security Carveout Aperture ID GSC_AID[4:0]
10	RO	X	**VPR**: Hardcoded to 0, VPR setting (sc2mc_vpr_wr) 1'b1: Write should not be allowed outside the VPR aperture 1'b0: Writes should be allowed anywhere in the memory (including VPR region).
9	RW	0x0	**RSB_NS**: ARPROT
8	RW	0x0	**COH**

Bit	R/W	Reset	Description
7:0	RW	0x0	**CSITE_SID**: CoreSight AXIAP StreamID for read transactions

CORESIGHT_CFG_ETR_MC_WR_CTRL_0

Offset: 0x20070
Read/Write: See table below
Parity Protection: N
Reset: 0x000X0X00 (0bxxx0,0000,0000,00xx,0000,0x00,0000,0000)

Bit	R/W	Reset	Description
28:26	RW	0x0	**USER_SIZE**
25:23	RW	0x0	**RSDV**
22	RW	0x0	**SP_AWARE**
21:19	RW	0x0	**SUB_PART_ADD**: Sub partition Address { sp_aware , user_adr1[8:6], user_adr[5]}
18	RW	0x0	**USER_ADR_5**
17:16	RO	X	**GSC_AL**: Hardcoded to 0, General Security Carveout Access { GSC_AL[1:0]}
15:11	RW	0x0	**GSC_AID**: General Security Carveout Aperture ID GSC_AID[4:0]
10	RO	X	**VPR**: Hardcoded to 0, VPR setting (sc2mc_vpr_wr) 1'b1: Write should not be allowed outside the VPR aperture 1'b0: Writes should be allowed anywhere in the memory (including VPR region).
9	RW	0x0	**WSB_NS**: AWPROT
8	RW	0x0	**COH**
7:0	RW	0x0	**ETR_SID**: ETR StreamID for write transactions

CORESIGHT_CFG_ETR_MC_RD_CTRL_0

Offset: 0x20074
Read/Write: See table below
Parity Protection: N
Reset: 0x000X0X00 (0bxxx0,0000,0000,00xx,0000,0x00,0000,0000)

Bit	R/W	Reset	Description
28:26	RW	0x0	**USER_SIZE**
25:23	RW	0x0	**RSDV**
22	RW	0x0	**SP_AWARE**
21:19	RW	0x0	**SUB_PART_ADD**: Sub partition Address { sp_aware , user_adr1[8:6], user_adr[5]}
18	RW	0x0	**USER_ADR_5**
17:16	RO	X	**GSC_AL**: Hardcoded to 0, General Security Carveout Access { GSC_AL[1:0]}
15:11	RW	0x0	**GSC_AID**: General Security Carveout Aperture ID GSC_AID[4:0]
10	RO	X	**VPR**: Hardcoded to 0, VPR setting (sc2mc_vpr_wr) 1'b1: Write should not be allowed outside the VPR aperture 1'b0: Writes should be allowed anywhere in the memory (including VPR region).
9	RW	0x0	**RSB_NS**: ARPROT
8	RW	0x0	**COH**
7:0	RW	0x0	**ETR_SID**: ETR StreamID for read transactions

CORESIGHT_CFG_CSITE_CBB_WR_CTRL_0

Offset: 0x20080
Read/Write: See table below
Parity Protection: N
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,x00x,xxxx,xx00)

Bit	R/W	Reset	Description
16:11	RO	X	**MSTR_ID**
10:9	RW	0x0	**VQC**
8:2	RO	X	**GRPSEC**
1:0	RO	0x0	**FALCONSEC**

CORESIGHT_CFG_CSITE_CBB_RD_CTRL_0

Offset: 0x20084
Read/Write: See table below
Parity Protection: N
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,x00x,xxxx,xxxx)

Bit	R/W	Reset	Description
16:11	RO	X	**MSTR_ID**
10:9	RW	0x0	**VQC**
8:2	RO	X	**GRPSEC**
1:0	RO	X	**FALCONSEC**

CORESIGHT_CFG_TPIU_CFG_0

Offset: 0x20098
Read/Write: R/W
Parity Protection: N
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	0x0	**TPCTL**

CORESIGHT_CFG_FSI_CLUSTER1_ALIVE_STATUS_0

Offset: 0x200f0
Read/Write: RO
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
30	X	**FSI_MV_ATB_CSYSREQ**
29	X	**FSI_MV_APB_CSYSREQ**
28	X	**FSI_MV_ATB_CSYSACK**
27	X	**FSI_MV_APB_CSYSACK**
26:23	X	**RESERVED**
22	X	**CPUHALT0**
21	X	**CPUHALT1**
20	X	**CPUHALT2**

Bit	Reset	Description
19	X	**CPUHALT3**
18	X	**nCPUPORESETDCLS0**
17	X	**nCPUPORESETDCLS1**
16	X	**nCPUPORESETDCLS2**
15	X	**nCPUPORESETDCLS3**
14	X	**nCORERESETDCLS0**
13	X	**nCORERESETDCLS1**
12	X	**nCORERESETDCLS2**
11	X	**nCORERESETDCLS3**
10	X	**nCPUPORESET0**
9	X	**nCPUPORESET1**
8	X	**nCPUPORESET2**
7	X	**nCPUPORESET3**
6	X	**nCORERESET0**
5	X	**nCORERESET1**
4	X	**nCORERESET2**
3	X	**nCORERESET3**
2	X	**nPRESETDBG**
1	X	**nTOPRESET**
0	X	**RSVD**

CORESIGHT_CFG_FSI_DBG_ALIVE_STATUS_0

Offset: 0x200f4
Read/Write: See table below
Parity Protection: N
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31	RW	X	**FSI_DBGAPB_TIMEOUT**: Debug APB Bus timed out for FSI **1** = TIMEOUT **0** = NORMAL

Bit	R/W	Reset	Description
30:29	RO	X	**FSI_POWER_CLAMP**
28:26	RO	X	**RESERVED**
25	RO	X	**FSI_CSITE_CE**
24	RO	X	**FSI_CHSM_CPU_CE**
23	RO	X	**FSI_FABRIC_CLK_TOGGLE**
22	RO	X	**FSI_CSITE_RSTN**
21	RO	X	**FSI_R5_nRESET**
20	RO	X	**FSI_R5_DBGRESETn**
19	RO	X	**FSI_R5_PRESETDBGn**
18	RO	X	**FSI_R5_nSYSPORESET**
17	RO	X	**dfd2fsi_cluster0_core0_dbgen**
16	RO	X	**dfd2fsi_cluster0_core0_niden**
15	RO	X	**dfd2fsi_cluster1_core0_dbgen**
14	RO	X	**dfd2fsi_cluster1_core0_niden**
13	RO	X	**dfd2fsi_cluster1_core0_hiden**
12	RO	X	**dfd2fsi_cluster1_core0_hniden**
11	RO	X	**dfd2fsi_cluster1_core1_dbgen**
10	RO	X	**dfd2fsi_cluster1_core1_niden**
9	RO	X	**dfd2fsi_cluster1_core1_hiden**
8	RO	X	**dfd2fsi_cluster1_core1_hniden**
7	RO	X	**dfd2fsi_cluster1_core2_dbgen**
6	RO	X	**dfd2fsi_cluster1_core2_niden**
5	RO	X	**dfd2fsi_cluster1_core2_hiden**
4	RO	X	**dfd2fsi_cluster1_core2_hniden**
3	RO	X	**dfd2fsi_cluster1_core3_dbgen**
2	RO	X	**dfd2fsi_cluster1_core3_niden**
1	RO	X	**dfd2fsi_cluster1_core3_hiden**
0	RO	X	**dfd2fsi_cluster1_core3_hniden**

CORESIGHT_CFG_HSSTP_LANE_CFG_0

Offset: 0x201b8
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	0x0	**HSSTP_TO_UPHY_LANE_MAP**: 0 = No lanes selected for output 1 = UPHY Lane-0 mapped to HSSTP Lane-0 2 = UPHY Lane-1 mapped to HSSTP Lane-0 3 = UPHY Lane-0 mapped to HSSTP Lane-0; UPHY Lane-1 mapped to HSSTP Lane-1 4 = UPHY Lane-4 mapped to HSSTP Lane-0 5 = UPHY Lane-5 mapped to HSSTP Lane-0

CORESIGHT_CFG_HSSTP_SPEED_CFG_0

Offset: 0x201bc
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	0x0	**RATE_ID**: Sets rate of transfer over UPHY 000 - 2.5Gbps 001 - 5Gbps 011 - 10Gbps

CORESIGHT_CFG_HSSTP_ACTIVATE_CFG_0

Offset: 0x201c0
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**UPHY_LANE_CTL_VALID**: Indicates to HSIO that contents of CORESIGHT_CFG_NVUL* are valid
1	0x0	**UPHY_LANE_MAP_VALID**: Indicates to CAR and UPHY that a valid HSSTP_TO_UPHY_LANE_MAP is present in CORESIGHT_CFG_HSSTP_LANE_CFG.
0	0x0	**RATE_ID_VALID**: Indicates to CAR that a valid RATE_ID is present in CORESIGHT_CFG_HSSTP_SPEED_CFG[RATE_ID] and parallel frequency should be updated accordingly

CORESIGHT_CFG_HSSTP_STREAM_CTL_0

Offset: 0x201c4
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**AITS_SET**: If 1, positive edge on DLY_TRIG0 sets ALLOW_IO_TRACE_STREAMING. If 0, it does not affect ALLOW_IO_TRACE_STREAMING.
1	0x0	**AITS_CLR**: If 1, positive edge on DLY_TRIG0 clears ALLOW_IO_TRACE_STREAMING. If 0, it does not affect ALLOW_IO_TRACE_STREAMING.
0	0x0	**ALLOW_IO_TRACE_STREAMING**: Allows stalling between TPIU and upstream device by masking atready, atvalid, afready

CORESIGHT_CFG_HSSTP_INTERRUPT_0

Offset: 0x201c8
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:1	0x0	**MSG_0**: Software programmable message for communication between Debugger and BPMP
0	0x0	**INT_0**: RW Interrupt to LIC, CTI1, BPMP VIC

CORESIGHT_CFG_HSSTP_ALIVE_STATUS_0

Offset: 0x201cc
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: CORESIGHT_CFG_SCR_HSSTP_ALIVE_STATUS_0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	R/W	Reset	Description
1	RO	0x0	**CAR_HSSTP_RST**
0	RW	0x0	**HSSTP_DBGAPB_TIMEOUT**

8.7.3.2 Scratch Registers

For base address of this block of registers, please refer to SCRATCH in the System Address Map.

These scratch registers are implemented in SRAM.
Hence, the reset values below may not always hold true such that they all need initialization.

SCRATCH_SECURE_RSV67_SCRATCH_0

Offset: 0x27c,0x280
Read/Write: R/W
Parity Protection: N
SCR Protection: SCRATCH_SCR_RSV67_SCR_0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**

8.8 System Registers

8.8.1 Miscellaneous Registers

This register provides the chip ID and revision information. Chip revision information can be used when software needs to identify the revision when behavior is different between revisions. Refer to the datasheet or errata document to see if this applies.

MISCREG_HIDREV_0
Offset: 0x4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00012347 (0bxxxx,xxx0,0000,0001,0010,0011,0100,0111)

Bit	Reset	Description
24:20	0x0	**PRE_SI_PLATFORM**: Set to 0 on silicon platform; set to pre-silicon platform type on pre-silicon platforms. **0** = SILICON **1-9** = reserved for pre-silicon use

Bit	Reset	Description
19:16	0x1	**MINORREV**: The encoding of this field is as below. A,B,C,D denote major revs, while 01, 02, 03 denote minor revision. 0000:Reserved 0001:A01 0010:A02 0011:A03 0100:Reserved 0101:B01 0110:B02 0111:B03 1000:Reserved 1001:C01 1010:C02 1011:C03 1100:Reserved 1101:D01 1110:D02 1111:D03
15:8	0x23	**CHIPID**: Upper two digits of the SoC ID.
7:4	0x4	**MAJORREV**: This field provides the last digit of the Chip ID.
3:0	0x7	**HIDFAM**: Chip ID family register. **0-6** = reserved for other NVIDIA products **7** = APXX_TXX (NVIDIA SOC)

8.9 Always-On Cluster (AON) and SPE

8.9.1 Overview

The AON cluster provides a processor (the SPE) and a set of peripherals in a small power island that may remain on when the majority of the System-on-Chip (SoC) is powered off. This cluster supports some I/O functions, and also provides hardware to wake the SoC when a number of different events occur.

Use cases include, but are not limited to:

- low power sensor management during standby
- running specific communications stacks in standby, including the CAN bus
- wake management from sleep states

The cluster consists of an ARM Cortex®-R5F processor core with a tightly coupled RAM, supporting peripherals (such as timers and an interrupt controller), various I/O controller peripherals, a DMA

engine, an inter-connect fabric, and the logic to manage the cluster power states and wake sensing feature.

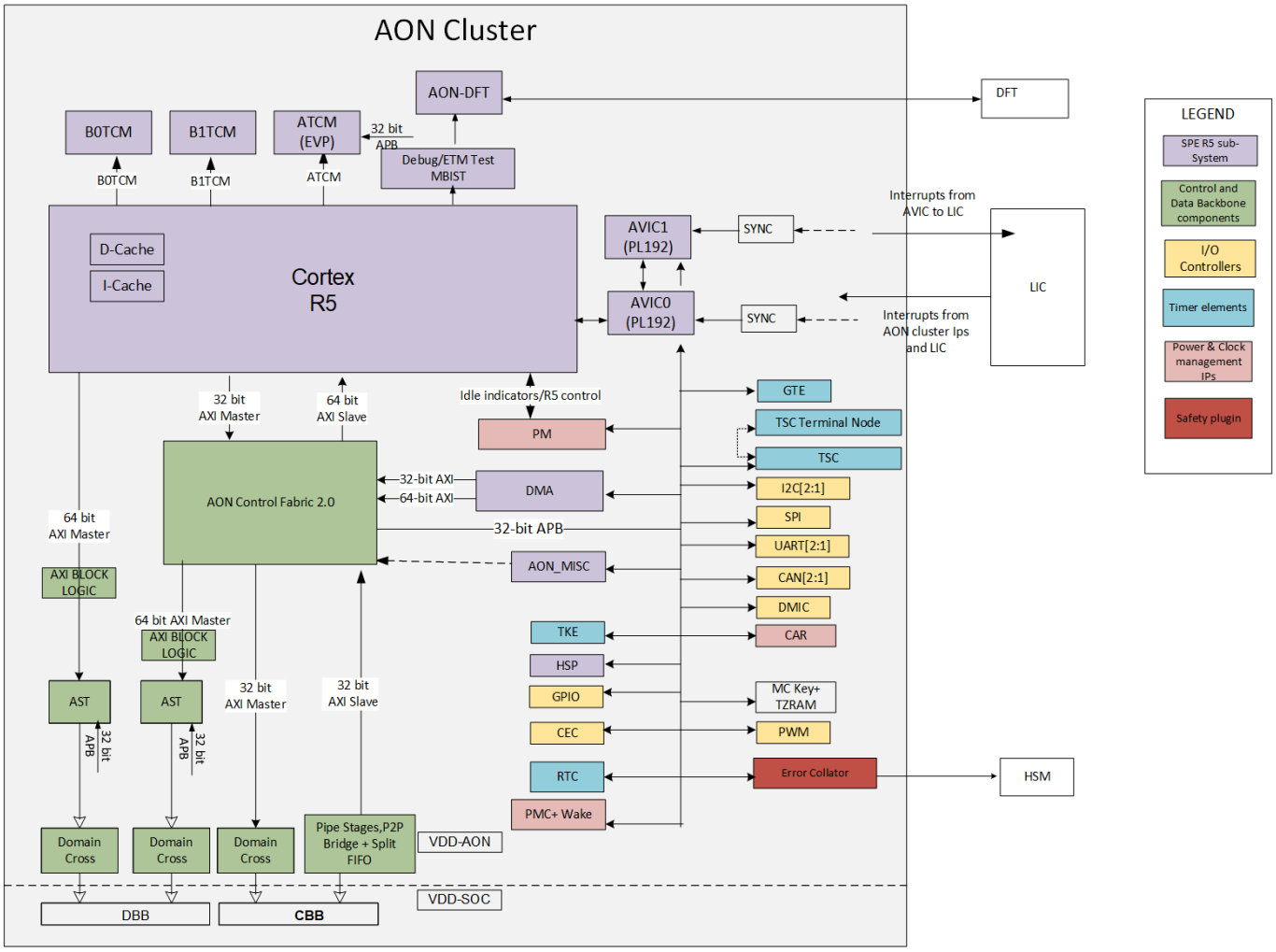
The Cortex-R5F processor in the AON cluster is also referred to as the SPE, an abbreviation for its mobile role as the Sensor Processing Engine. For brevity, the Cortex-R5F core used in AON is referred to as Cortex-R5 or just R5, but does include the Floating Point and Double Precision Floating Point capabilities.

While the Cortex-R5 may be used for the use cases outlined above, it is actually a generic CPU able to run arbitrary software. Nothing in the hardware design limits usage to just those applications.

8.9.1.1 AON Cluster Diagram

The following figure shows the block diagram of the AON Cluster.

Figure 8.74 AON Cluster Block Diagram



8.9.2 AON Cluster Functional Description

8.9.2.1 Cortex-R5 Processor

The Cortex-R5 processor supports the following features:

- ARMv7-R ISA
- Integrated instruction and data caches
- TCM interface for local SRAM
- Vectored interrupt support
- 64-bit AXI master interface for DRAM requests

- 32-bit AXI master interface for MMIO requests
- 32-bit AHB master interface for AVIC access
- AXI slave interface for DMA access to the local SRAM

8.9.2.2 Processor Configuration

The AON Cluster Cortex-R5 is implemented with the following features:

- 32 KB instruction cache
- 32 KB data cache
- 256 KB of SRAM attached to TCMB as two banks of 128 KB SRAM
- TCMA base address set to 0
- Init low vectors
- Processor version r1p3
- Floating-Point Unit extensions including double-precision capability

8.9.2.3 Interrupt Controller

The Cortex-R5 does not have an integrated interrupt controller. A pair of ARM PL192 vectored interrupt controllers on the local bus serve as the interrupt controller for the AON Cortex-R5. The SoC implements PL192 version r0p0.

8.9.2.4 Processor and Interrupt Controller Documentation

Detailed documentation for the Cortex-R5 Processor and the PL192 Interrupt Controller are readily available from ARM, and may be downloaded from the ARM web site. A good starting point is these documents:

- *Cortex -R5 and Cortex-R5F Technical Reference Manual*, ARM Document DDI 0460C
- *ARM PrimeCell Vectored Interrupt Controller (PL192) Technical Reference Manual*, ARM Document DDI 0273A

This TRM assumes access to these documents, and does not reproduce information from them.

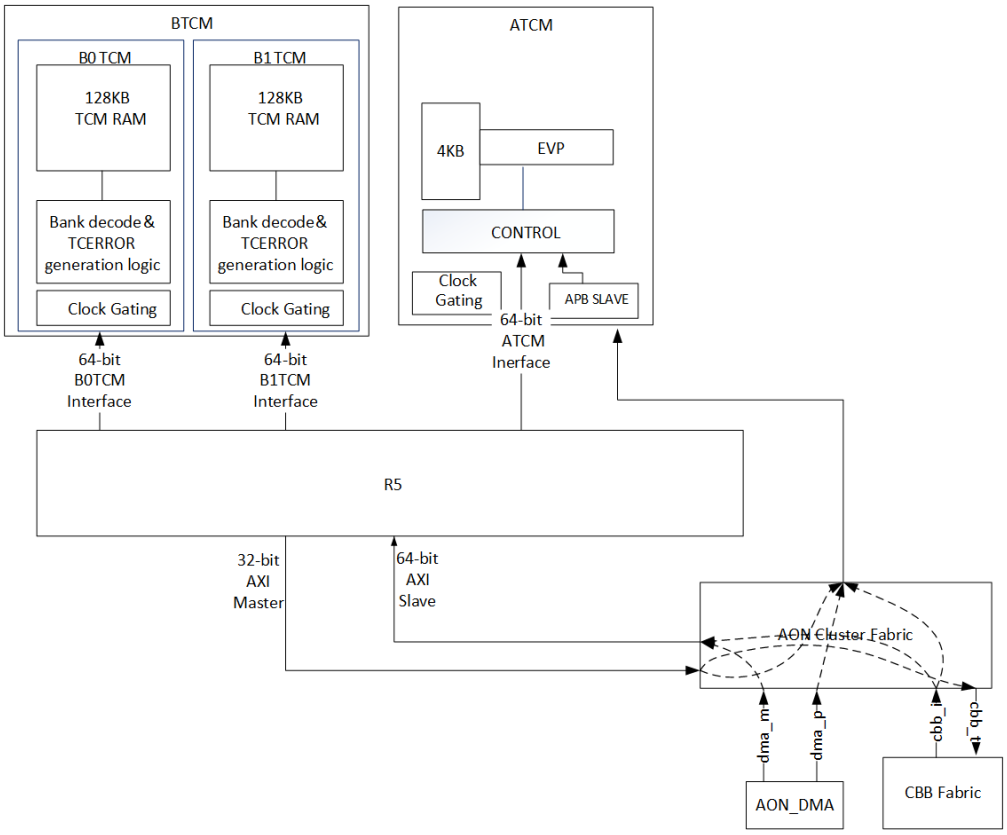
8.9.2.5 Timers

The Cortex-R5 does not have any integrated timers. Timers for the AON Cluster are provided by a separate timer block in the AON Cluster.

8.9.2.6 TCM Interface

Cortex-R5 provides two tightly coupled memory ports. These ports, named ATCM and BTCM are meant to provide low latency access to the Cortex-R5 processor. The ATCM hosts the EVP interface (see below) and the BTCM hosts a 256 KB RAM.

Figure 8.75 Cortex-R5 TCM interface



The following pin and build configurations define the ATCM and BTCM configurations.

8.9.2.7 ATCM Interface

The Cortex-R5 ATCM interface hosts the Exception Vector Table module. The EVP is wrapped in the ATCM Wrapper that provides the ATCM interface to Cortex-R5.

8.9.2.7.1 Exception Vector Pointers (EVP)

When Cortex-R5 takes an exception (including the reset exception), processor execution is forced to an address that corresponds to the type of exception. These addresses are called the *exception vectors*. By default, the exception vectors are eight consecutive word-aligned memory addresses,

starting at an exception base address (which is reset default to 0x0000_0000 for the AON Cortex-R5 since VINITHI is tied to 0).

The EVP module is implemented through a set of 16 registers starting at address 0x0. The table below shows the default organization of the EVP coming out of reset. The programmability of EVP allows the processor to jump to an arbitrary address space in the entire memory space that is accessible to the AON Cortex-R5.

Table 8.86 EVP Register Table

Offset	Data	Description
0x00	LDR, PC, [PC+0x18]	Reset vector. According to the ARM Architecture manual, the PC reads an address equals current instruction + 8.
0x04	LDR, PC, [PC+0x18]	Undefined exception
0x08	LDR, PC, [PC+0x18]	SWI (also known as SVC) exception
0x0C	LDR, PC, [PC+0x18]	Prefetch abort
0x10	LDR, PC, [PC+0x18]	Data abort
0x14	LDR, PC, [PC+0x18]	Reserved
0x18	LDR, PC, [PC+0x18]	IRQ
0x1C	LDR, PC, [PC+0x18]	FIQ
0x20	NV_ADDRESS_MAP_AON_BTCM	TCMB Base Address
0x24	NV_ADDRESS_MAP_AON_BTCM + 4	
0x28	NV_ADDRESS_MAP_AON_BTCM + 8	
0x2C	NV_ADDRESS_MAP_AON_BTCM + 12	
0x30	NV_ADDRESS_MAP_AON_BTCM + 16	
0x34	NV_ADDRESS_MAP_AON_BTCM + 20	
0x38	NV_ADDRESS_MAP_AON_BTCM + 24	
0x3C	NV_ADDRESS_MAP_AON_BTCM + 30	

The EVP table above assumes that the TCMB starts at address NV_ADDRESS_MAP_AON_BTCM. The redirection of exception vectors enables BPMP to execute an ISR handler in the entire memory space accessible by it. The following steps are performed to handle EVP requests:

1. The Cortex-R5 issues a read for exception vector address (@0x0000_00XX).
2. EVP decodes exception vector aperture, and generates read-response.
   a. Returns LD PC, [PC + 0x18] instruction as return data.
3. This results in PC getting loaded with pointer to exception vector.

Notes:

1. Cortex-R5 additionally supports vectored exception handling specifically for IRQ interrupts. This is supported in conjunction with the PL192 VIC interrupt controller. When an IRQ exception is triggered by the VIC, it can optionally send out a pointer to the interrupt handler to service that particular interrupt. In case of vectored interrupts through VIC, the EVP table is not referenced by the Cortex-R5.
2. EVP in earlier chips supported vectorization of interrupts. This is deprecated since the required vectorization support is already provided through VIC interface.
3. EVP reset is decoupled from the Cortex-R5 CPU reset. This enables the Cortex-R5 to be independently reset without affecting the content of EVP registers.

8.9.2.8 Error Checking and Signaling

When access is not allowed by the ATCM interface, it signals an error. When bit[0] of the Auxiliary Control Register (ACTLR.ATCMECEN) is set to 1, the error triggers an undefined exception on Cortex-R5 accesses or AXI slave error on AXI slave requests by asserting the ATCERROR.

ATCERROR is asserted when any of the following happens:

- ATCADDR address does not decode into the address space implemented within the ATCM wrapper.
- Whenever write access is made from the ARM ATCM interface to the ATCM.

ATCLATEERROR is not used to trigger an error since the ATCLATEERROR response is discarded by the AXI slave interface.

8.9.2.9 BTCM Interface

BTCM hosts the Cortex-R5 tightly coupled 256 KB RAM. BTCM is split internally into two ports, B0TCM and B1TCM. The address range for the BTCM is defined in the NV_ADDRESS_MAP_AON_BTCM define in the global AMAP.

The BTCM has two ports for two reasons:

1. One can be used by the processor and the other by the AON DMA. Cortex-R5 can work on one bank of the BTCM while AON DMA can page in data/code to/from the other.
2. If the code and data space can be separated between B0TCM and B1TCM, then instruction and data accesses from Cortex-R5 can be done independently.

8.9.2.10 BTCM RAM Configuration

The 128 KB of RAM on each BTCM channel is constructed of eight banks of 16 KB RAM. The BTCM RAM banks are logically configured into eight 32 KB power-gating groups. Since the BTCM interface is operating in interleaved mode one bank from B0TCM and one bank from B1TCM are in each power-gating group. Bank 0 starts at the bottom 32 KB of the BTCM address space.

8.9.2.11 Error Checking and Signaling

Accesses to the BTCM signal an error when the particular access isn't allowed by BTCM interface. When bit 1 and bit 2 of the Auxiliary Control Register (ACTLR.B0TCMECEN and ACTLR.B1TCMECEN) are set to 1, the error triggers an undefined exception on Cortex-R5 accesses or AXI slave error on AXI slave requests by asserting the B0TCERROR or B1TCERROR.

An error is asserted when any of the following occurs:

- The SD, SLP, or DSLP port for a TCM RAM bank is asserted and the TCM RAM bank is accessed.

8.9.2.12 Interrupts

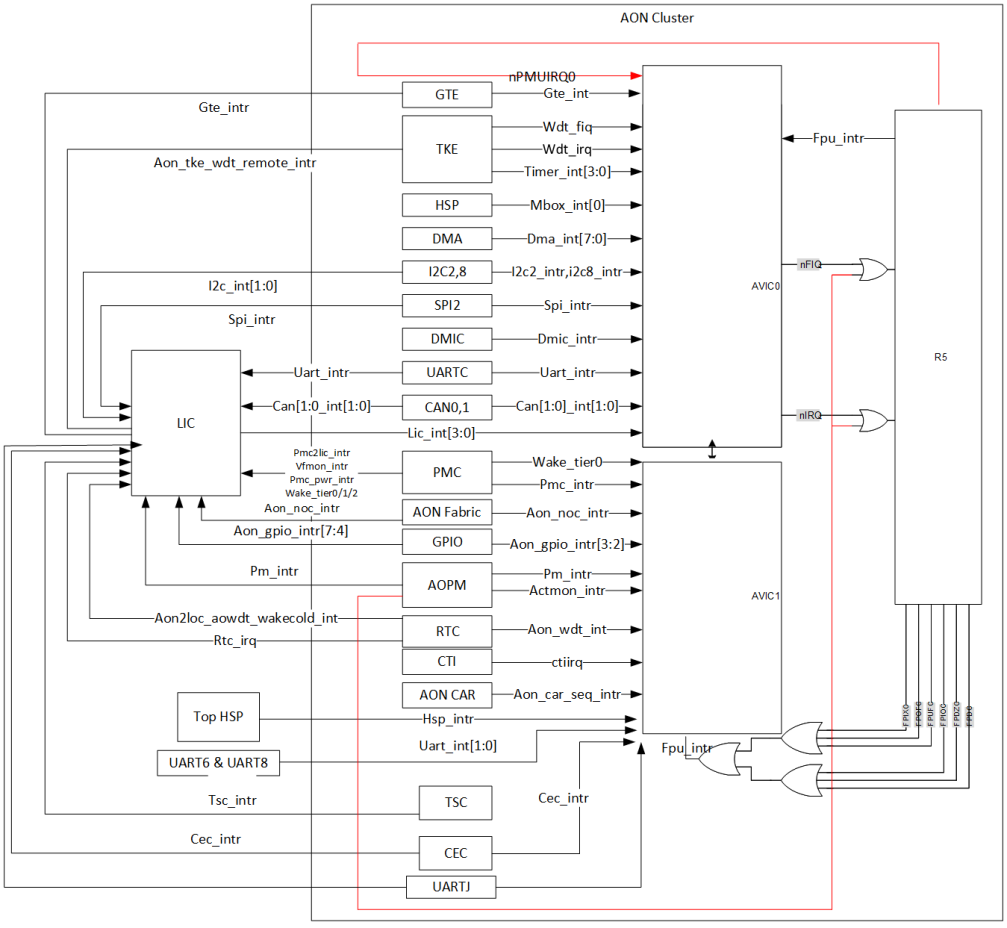
The interrupt sources from/to the SPE (Cortex-R5) in the AON block are comprised of the following components:

1. {Vector/Scalar} interrupts are driven through a pair of daisy-chained ARM Vectored Interrupt Controllers (AVIC)
   - Specific peripherals such as TKE, DMA, etc., (which are exclusively owned by the SPE) have their interrupts connected to AVIC.
   - Other I/O peripherals which can be owned by the SPE and other system masters, such as the CCPLEX, send their interrupts to AVIC as well as to top level LIC, depending on the usage model software can mask the interrupts in unused path.
2. Interrupts from the Legacy Interrupt Controller (LIC) that feed into the AVIC.
   - There are four such interrupts coming from LIC. There are a few peripherals and interrupt sources connected to LIC can be managed by the SPE by properly enabling the routing to AON cluster channel in LIC and finally routing the aggregated interrupt to AVIC.
3. AON Interrupts to other Clusters
   - There are a few peripherals that are controlled by other clusters though they are instanced in the AON cluster (e.g., CAN, UART). For example, specific instances of the UART have usage models where they are controlled by the Safety Cluster Engine (SCE) or the Boot and Power Management Processor (BPMP). These peripherals interrupt are connected to the AON cluster and to the other clusters apart from LIC; they are fanned out to LIC and other respective clusters.
4. Interrupts to AON from SoC Domain Peripherals

- UART6 and UART8 from the SoC domain send interrupts to AON cluster. These interfaces could be repurposed as Debug UART in closed system debug; interrupts from these UART instances are routed to AVIC so that the Debug UART driver running in SPE (which various debug routines interact) is getting the correct interrupt.

5. Secure Interrupts Routing
   - Secure interrupts are generally routed to LIC and subsequently handled by the TrustZone Interrupt handler. Secure Interrupt is also routed as a fault to the HSM via error loggers in the AON cluster NoC.
   - There are cases where the secure interrupt could be overloaded with normal functional scenarios (e.g., APB slave indicates a security access violation, such as SCR, or other violations such as an address hole via SLAVE_ERR). If the secure interrupt become overloaded, the AON Fabric asserts the Secure Interrupt to be resolved by the TrustZone software.

Figure 8.76 Interrupt Structure



Like any other APB peripherals in AON, AVIC access and configuration spaces are accessed via MMIO path. The AVIC port of Cortex-R5 interfaces to the PL192 and enables the following functions in hardware to minimize software overhead of servicing the interrupt.

1. Determine from the interrupt controller which interrupt source is requesting service.
2. Determine where the service routine for that interrupt source is loaded.
3. Masks that interrupt source, before re-enabling processor interrupts to permit another interrupt to be taken.

Interrupt connections to PL192 include:

1. Local interrupts originating from within the AON cluster are connected to PL192 interrupt inputs; these are vectored (i.e., mapped to IRQ for low interrupt latency).
2. Four external interrupts are connected from the LIC. Each interrupt can be an aggregation of one or more interrupts.
3. Interrupts from selective SoC peripherals like UARTs and Doorbell units.

AON Fabric

The AON cluster uses a single FlexNoC IP for routing access to DRAM via DBB and MMIO requests (via CBB to SoC peripherals and peripherals in AON).

8.9.2.12.1 AON Cluster Fabric

The AON cluster fabric uses various Software entities to access various IP's register space. The AON Cluster fabric supports:

1. MMIO requests from various AON initiators to SoC peripherals (via CBB), and to peripherals in AON.
2. MMIO requests from various CBB initiators to IPs in the AON cluster.
3. AON DMA accesses to DRAM, AON TCMs, and various MMIO requests to IPs in AON and SoC domains.

AON Cluster Fabric Topology

The AON cluster fabric supports the initiators mentioned below as per the protocol and data widths. Each of the below initiator is connected to an initiator node inside the cluster fabric.

Initiator	Protocol	Data-Width	Power Domain of the initiator node	Comments
SPE R5's Periphery port	AXIv3	32-bit	AOPG	SPE R5 accesses various IP's (both in SOC and AON) register configuration space through this port
AON DMA Periphery port	AXIv3	32-bit	AOPG	AON DMA accesses various IP's especially IO controller's register configuration space through this port
AON DMA Memory port	AXIv4	64-bit	AOPG	AON DMA accesses memory and AON TCM through this port

Initiator	Protocol	Data-Width	Power Domain of the initiator node	Comments
CBB in SOC domain	AXIv4	32-bit	AOPG (MV crossing from SOC to AOPG domain)	Intermediate initiator on AON Cluster fabric, through which various SoC initiators can access AON IP's register configuration space. Note that this initiator port does not have physical path to Cortex-R5 AXI Target.

AON Cluster Fabric supports the targets mentioned in the below table as per the protocol and data widths. Each of the below target is connected to a target node inside the Cluster Fabric.

Target	Protocol	Data-Width	Power domain of the target node	Comments
AST1	AXIv3	64-bit	AONPG	AON DMA memory accesses go through this target. AST0/1 has certain security critical configuration along with address translation setup for Cortex-R5 and DMA's memory accesses. ASTs are placed in AONPG domain to retain all such configuration for any latency critical use cases where SPE needs early access to DRAM/MC during the warm boot.
CBB Intermediate target	AXIv4	32-bit	AOPG	AON Initiator's MMIO requests to SoC domain go through this target.
R5 AXI Target	AXIv3	32-bit	AOPG	AON TCM can be accessed by AON DMA through this target.
AXI2APB Target node	APBv4	32-bit	AOPG	All PG-able and Non-PG-able APB targets are connected to this node. All PG-able APB targets connect to this node via a MDMM, placed in PG domain. All NPG-able APB targets connect to this node via a MDMM, placed in NPG domain.
Cluster Fabric internal APB Targets	APBv4	32-bit	AOPG	Cluster fabric internally hosts APB targets for configuring firewalls, error loggers, error collator, and error monitors.

Target nodes connect to the Initiator nodes via a Junction Box which has the arbiter and routing logic. In addition, cluster fabric implements common firewall and error logger/notifier/collator logic.

Error Generation and Logging

Error Monitor

Each Initiator node in the fabric contains an Error Monitor Module. This module can raise an interrupt on the following errors:

- SLVERR/DECODE error
- Power down error
- Firewall error
- Timeout error
- Unsupported access error

Error Notifier and Interrupt

The AON Cluster fabric implements one Error Notifier module which collates interrupts from various Error Monitor blocks in the initiator nodes and generates a single interrupt signal. This interrupt is routed to both LIC and SPE-AVIC. In addition, this interrupt is routed as a fault directly to the HSM.

Error Collator

The AON cluster fabric implements error collators in each partition where the AON Fabric logic exists. Faults from the diagnostics implemented in the fabric logic sitting in each partition are routed to the respective error collators. In AON, since the fabric logic is implemented in both AOPG and AONPG partitions, four faults from AON fabric are routed to HSM (two from each EC i.e., a corrected and uncorrected fault from each error collator).

8.9.2.12.2 Always-On Power Manager (AO-PM or PM)

The block called AO-PM contains registers and state machines which enable the Cortex-R5 and other resources to enter low power states when the Cortex-R5 is idle. Additionally, AO-PM also contains some general configuration registers related to the cluster.

The AO-PM unit implements the SPE Power States and coordinates with PMC and AO-CAR to manage the SPE and System Power State transitions. It provides a programming interface to the SPE to manage the cluster's power states. The functionalities of this module include the following:

1. SPE Power State Control: Making power state transition decisions according to enabled power state configurations and triggering entry-exit sequences to/from SPE power states.
2. SPE Power State exit management: Monitoring and responding to the processor complex and DMA engine's activity requests and wake events that trigger SPE Power State exits.
3. Clock source management: Sending control signals to AO CAR state machines for controlling the AO PLL, switching clock sources for the processor complex for the target SPE power state.
4. Power-gating: Handshake with PMC to support retention mode for the processor caches, TCMs RAMs, and the power gate-able portions of the AO cluster. Handshakes with NOC-PM to make sure there are no pending transactions before power gating AOPG partition and ensures NOC is ready to accept transactions post PG reset deassertion.

5. Power Security: Supports a secure configuration of the mapping of the clocking and power resources per AO power state.
6. PMC handshake: Handshake with PMC to support interaction of SPE power states with SC7.
7. RTC Alarm: Implements an alarm for the SPE to be woken up from Dormant state when the timers are power-gated.

This module is not directly responsible for the SoC power states which are controlled by PMC.

8.9.2.12.3 Use Cases for the AO-PM

1. Cortex-R5 and TCM Idle based power management
2. Dynamic Clock Gating of TCM clocks
3. Dynamic Power Gating of Cortex-R5 cluster, if supported
4. Cortex-R5 Clock burst policy (For AON_CPU_NIC_CLK and AON_APB_CLK) on interrupts
5. SRAM low power mode controls
6. Baseline support for SC7
7. Boot support

8.9.2.12.4 SPE Power State Management

At the core of the AO-PM Module is a Power State Transition FSM that controls transitions from SPE ACTIVE to any of the "target" low power states. These states can be SPE_IDLE, SPE_STANDBY, or SPE_DORMANT, either Shallow or Deep.

Before the SPE executes WFI/WFE, it programs the power configuration registers in AO-PM to a target state, such that whenever the AO-PM detects WFI/WFE along with nCLKSTOPPED, it can readily transition to one of the targeted states. Primarily, two sets of registers must be programmed:

1. TGT_PWR_STATE register (Programmed by SPE-FW).
2. POWER_CFG_MASK registers (Set up by Power Security software). This set overrides the TGT_PWR_STATE registers.

Entry into any of the low-power states is possible by two conditions:

1. Functional Trigger from the Cortex-R5: Either nWFIPIPESTOPPED or nWFEPIPESTOPPED is asserted, along with nCLKSTOPPED asserted.
2. Debug trigger: A debug register (DBG_FEATURES_0_FORCE_START) write can also trigger a specific state machine. This is provided for ASIC debug purposes only.

> AOPM does not enter into the low power states if the global clock override is set.

Exit conditions from low power states are:

- !WFI/WFE/nCLKSTOPPED
- DMA active (Wake on a transaction on AXI-S port of the Cortex-R5)
- IRQ or FIQ
- LP_EXIT trigger (register write)
- DEBUG TIMEOUT (if stuck in a state longer than timer value)
- RTC alarm
- Wake Interrupt (Tier 0)
- SC7 wake request goes high from PMC (For Tier 1 or Tier 2 wake events)

Some interrupts generated from the Cortex-R5-clusters are not routed to the local VIC but go to the LIC. These are not capable of automatically causing the PM to exit an LP state. The low power states lower the cluster clock frequencies and potentially slow down the interrupt service routine running on the external target processor. Such interrupts and their effects are analyzed below.

1. Watchdog_remote interrupt – This is the WDT 3rd expiry interrupt from the local TKE's WDT. The handler for the remote WDT wakes up the PM by issuing a software triggered exit through the LP_EXIT register. For AON, when the expiry occurs in SC7, the 3rd expiry of SPE-WDT causes an SC7 exit. This results in the AOPM coming out to an ACTIVE state and then re-entering a SHALLOW state. In the SHALLOW state, the handler for the WDT accesses the LP_EXIT register and causes a wake to ACTIVE. In order to prevent the PM from going back to a low power state, the WFI is masked by the handler inside the PM. In general, the 1st or 2nd expiry of the WDT causes an exit from the LP state – hence the 3rd expiry is not expected to occur in an LP state. However, if the 3rd expiry occurs in an LP state (or if the 1st and 2nd expiries are masked) then the handler uses the knobs described above to force the PM to exit to the low power state.
2. HSP_SI_[1|2|3|4] interrupts – An outbound full interrupt is raised when the local processor posts a message to another processor. The local processor may execute WFI after posting the message. The PM does not need to stay in the ACTIVE state. When the target processor consumes the interrupt and clears it, the local processor does not need to move to the ACTIVE state. If the target processor requires further communication with the local processor, it posts another message that causes the HSP_SI_0 interrupt to assert and wake the PM from the ACTIVE state. An outbound empty interrupt is raised when the target processor consumes the message. The interrupt requires the target processor, not the local processor, to wake to the ACTIVE state.
3. APB bridge error and security interrupts – The effect of these interrupts not causing an LP (Low Power) state exit is to slow down the interrupt handler routine.

SPE-IDLE Power State

SPE-Idle is a group of idle states characterized by the following steady state conditions:

- Cortex-R5 in Standby: nWFIPIPESTOPPED or nWFEPIPESTOPPED, and nCLKSTOPPED are asserted.
- The states of the L1 caches, ASTs and external paths from the SPE cluster to SoC domain are subject to ULP mode. These are idle and not being used by the Cortex-R5 actively.

- Second level clock gating handles the SPE and TCM clocks. I/O controllers may be clock-gated by software.
- DMA is idle.
- I/O controllers in slave mode may be active and using local buffering but those in master mode are idle.
- Clocks can be switched to alternative sources including OSC. Switchover to 32 kHz is allowed only when ALLOW_DEEP_I bit of AOPM_POWER_CFG_MASK_0 is set and the system is in SC7 state.

The I/O controller configurations and their clock controls can be in various states depending on use cases.

Entry Configuration

- The TCM is clock gated on assertion of nCLKSTOPPED signal. AO-PM detects this signal to gate and ungate the TCM clocks.
- The AO PLL may be configured by SPE-FW to be placed in a low power mode like DISABLE/IDDQ if the I/O controllers are under S-managed clock-gated state or if their clocks are configured to be supplied from non-PLL sources like OSC and 32 kHz.
- When AO PLL is placed in a low power mode, the default clock source is switched to OSC. The switching of the clock source is handled by AO-CAR.
- AON Rail retention is not enabled in this state. The TCM does not expect to be in retention mode in this shallower idle state.
- BPMP and PMC control whether clocks can be switched to 32 kHz in this state when the system is in SC7.

SPE-Standby Power State

The SPE-Standby state is entered when the SPE subsystem has been resident in the SPE-Idle state for a programmable interval of time. SPE-Standby is similar to SPE-Idle except that the TCM and cache SRAMs are placed in a low power mode (RET). The TCMs or the caches cannot be accessed when placed in RET mode. When the RAMs are placed in RET mode, the AO rail can be externally lowered to the level of SRAM retention voltage (expected to be higher than minimum functional logic voltage)

TCM and Cache RAMs are put in RET mode by the AOPM Standby sequencer when the RAM_RET bit in the AOPM_TGT_PWR_STATE_0 register is set. Other cluster RAMs are placed in RET mode under software control. Clocks and AO PLL are configured the same way as for SPE-Idle. Before the TCM is placed in retention, access to the TCM controller are blocked by the AO-PM hardware. The AO-PM also resolves race conditions between entering this low power state and an exit trigger like an interrupt showing up during entry. The L1 caches can be placed in retention mode along with the TCMs. The interrupts to the Cortex-R5 are explicitly blocked when caches are placed in retention. Accesses through the AXI-S slave port are blocked by AO-PM hardware too.

Exit conditions for this state are identical to SPE-Idle state. The Deep Standby state entry is allowed only if the system is in the SC7 state and the ALLOW_DEEP_S bit in the AOPM_POWER_CFG_MASK_0 register is set.

SPE-Dormant Power State

The SPE wakes up on a CAN wake event to process the CAN message. The hardware architecture supports a generic partially power-gated state for the AO cluster that can also be entered more aggressively when the sensor functions are enabled in the ULP mode. This power state may also trivially be engaged when sensor functions are statically disabled in the system by configuration or by user intervention.

TCMs, Caches and other RAMs in AOPG cluster can be placed in RET mode. SPE's reset vector must point to the TCM when the processor is taken out of power-gating since the SoC domain is powered off during this time. When all the SRAMs in the VDD_RTC power domain are placed in RET mode, the VDD_RTC rail can be lowered to SRAM_VRET. AOPG cluster can be power gated completely.

In addition to the low power states of the resources supported in SPE-Idle and SPE-Standby state, the OSC pad can be turned off in this state if the system is in SC7. This requires all clocks to be switched to 32 kHz either in software or in hardware or gated off prior to OSC is turned off. AOPM can enter its Deep Dormant state, only when SoC is in the SC7 state. The PM handshakes with the SC7 FSM in the PMC to ensure that the SoC is in SC7 before the PM enters the deep dormant state. There is a software knob to bypass the second clock switch (the one that switches to 32K and turns off OSC). If the BYPASS_TWO_STEP bit is set, DEEP_DORMANT is entered after PG. (There is no retention wait counter for dormant state.)

The RAM_RET and RAIL_RET options are as discussed in the STBY section. The only difference is that even if no RAM_RET is requested, hardware will do RAM_RET, but it will still not override what software has requested.

Exit from the SPE-Dormant state is triggered by interrupts (in the shallow state) from sources that are not power-gated and by wake events from sources that are power-gated, including the SC7 Tier 0 wake events. Thus, exit from the SPE-Dormant is automatically included in the SC7 exit sequence.

Exit Handling Due to SC7 Exit

When AOPM FSMs are in any non-active state, an SC7 exit of the PMC results in an exit condition of AOPM FSMs, and this exit in non-maskable. That is, unlike for other exit causes who have a mask/unmask bit in POWER_CFG_MASK register, the SC7 exit from the PMC results in AOPM exiting from LP states.

While an SC7 entry of the PMC gates the entry into deep states only, an SC7 exit causes an exit whether AOPM FSMs are in their shallow or deep flavors. An SC7 exit also gates off further deep state re-entry, unless explicitly reprogrammed to do so by the BPMP before the next SC7 entry.

Example: If in a deep state and an SC7 exit happens in the PMC:

- Some deep state -> (SC7 wake) -> Shallow state -> Transitions for exit -> Active -> (LP condition) -> Go up to Shallow only

If in a shallow state and an SC7 exit happens in the PMC:

- Some shallow state -> (SC7 wake) -> Transition for exit -> Active -> (LP condition) -> go up to Shallow only

Clock Burst on Interrupt

The processor may be busy executing very low horsepower tasks but not be able to suspend all tasks and execute a WFI instruction. The processor may switch its clocks to a slow source like OSC clock. This slow clock affects the latency to a subsequent interrupt. The added latency of processing the interrupt signal sequence and the ISR at a slow clock prevents the processor from switching to the slow clock when it is not completely idle but very lazy. AOPM supports switching the processor subsystem clocks to a faster source automatically when an interrupt is triggered. The interrupt specific clock source is used in both active and low power idle states of the processor complex.

When the processor is active, the SPE-FW requests the AO-PM to switch aon_cpu_clk, aon_nic_clk and aon_apb_clk to pre-programmed values in the AO-CAR. SPE-FW writes to AOPM_PWR_STATE register STATE = ACTIVE_IRQFIQ to raise this request. AO-PM sends a request to AO-CAR using the TGT_ACTIVE_IRQFIQ command index, waits for the AO-CAR to switch the clocks and acknowledge, after which it updates AOPM_PWR_STATUS to STATE = ACTIVE_IRQFIQ to reflect that the software request is completed. SPE-FW switches the clocks back to the TGT_ACTIVE state after servicing the pending interrupts. When SPE-FW switches the clocks back to TGT_ACTIVE state, the AOPM_PWR_STATUS gets updated to STATE = ACTIVE.

When the processor is idle and the AO-PM has placed the subsystem in one of the low power states described above, if the exit condition is an interrupt, the AO-PM hardware requests the AOCAR to auto-switch to the faster clock sources if BURST mode is enabled.

This feature leverages the index-based request/acknowledge protocol between AOPM and the AO-CAR block. BURST_CLK_EN field in AOPM_BURST_CLK_0 register must be set to enable this feature in ACTIVE state and BURST_LP_EN field of the same register must be set to enable this feature in non-ACTIVE or low power states. If the interrupt to Cortex-R5 is masked inside the PM module using the LP_EXIT_MASK register, then clock bursting is bypassed by the PM hardware upon a wake from nCLKSTOPPED or nWFIPIPESTOPPED de-assertion event.

8.9.2.12.5 AO I/O DPD

AO-PM does not control any DPD control of the AO I/O pads. This is handled inside the PMC.

8.9.2.12.6 SRAM Low Power Mode Management

AO-PM implements a handshake with the PMC in the Standby/Dormant states in order to put the following RAMs in retention mode.

This interface constitutes a request from the AO-PM and an acknowledgement coming from the PMC. For the RAMs to be put in retention, the request line is asserted HIGH and low indicates removal of Retention mode.

In the case of NO_ACK from PMC, PM FSM times out and an interrupt can be raised on NO_RET_ACK.

The following RAMs are controlled by PMC:

1. B0TCM and B1TCM_Bank0
2. B0TCM and B1TCM_Bank1
3. B0TCM and B1TCM_Bank2
4. B0TCM and B1TCM_Bank3
5. B0TCM and B1TCM_Bank4
6. B0TCM and B1TCM_Bank5
7. B0TCM and B1TCM_Bank6
8. B0TCM and B1TCM_Bank7
9. Cache0
   - LCache0
   - DCache0
10. CAN0
11. CAN1

In addition to the hardware handshake, shutdown and retention controls of these Cluster RAMs can be directly controlled by software and those registers lie in the PMC module.

Note: CAN RAMs are not controlled by AO-PM hardware sequencer. It is mandatory for SPE-SW to put CAN RAMs in low power mode (RET or SLEEP) if AO-PM is configured to enter the deep-dormant state, as it is illegal to have RAM in functional mode and Logic in a PG-ed state.

If RAM_RET == RAIL_RET == 0, then Standby FSM becomes identical to Idle FSM, barring the fact that different CAR target request ids are sent from PM to CAR.

8.9.2.13 AXI Block Support

AOPM provides support via configuration registers to block AXI requests. The following register in AO-PM supports this.

Register: AOPM_AON_BLOCK_0

Fields:

- AOPM_AON_BLOCK_0_AON2SNIC_REQ: This bit blocks AXI requests on the control fabric to the SOC. Since AON can be operating when SOC is not powered on, it is used to block the requests going towards SOC.
- AOPM_AON_BLOCK_0_AON2MC_REQ: This bit in AO-PM blocks AXI requests to MC. Since AON can be operating when MC is not powered on, it is used to stop the requests going towards SOC.

Additionally, there is a status register to read back status from the AXIBLOCK FSM via a block status [31:0] sideband input to AO-PM.

8.9.2.14 Interrupts

AO-PM has three interrupts going out.

Table 8.87 PM Interrupts

	Interrupt Sources
Pm2vic_intr	▪ RTC_ALARM ▪ NO_CAR_ACK ▪ NO_OSC_ACK ▪ NO_RET_ACK ▪ NO_NOC_IDLE_ACK ▪ NO_NOC_IDLE
Pm2lic_intr	▪ RTC_ALARM ▪ NO_CAR_ACK ▪ NO_OSC_ACK ▪ NO_RET_ACK ▪ NO_NOC_IDLE_ACK ▪ NO_NOC_IDLE
Pm_actmon_intr	▪ PM Unit Actmon

RTC_ALARM = When the RTC second/millisecond time matches the programmed value in the configuration registers, it generates an RTC alarm. The RTC generates an interrupt or generate an exit from low power states. The interrupt handler for the RTA Alarm Wake reprograms the AOPM_RTC_ALARM_TIMER and/or AOPM_RTC_ALARM_MSEC_TIMER registers before clearing the interrupt so that the interrupt does not fire spuriously. BPMP_PM (bpmp_pm2lic_int) and BPMP-PM (bpmp_pm2vic_int) are generated from the same sources, but have a different set of status (pre-mask), Mask, Set, and Clear registers for VIC and for LIC interrupts. Note that SPE should not

be interrupted for NO_DVS_ACK. The interrupt to NO_DVS_ACK can be selectively masked off for SPE but enabled for LIC (for BPMP).

NO_CAR_ACK = When CAR is requested to change the clock source or assert/de-assert PG resets, but the PM FSM times out waiting for an ACK, this interrupt is raised. The PM FSM cannot assume a good clock or a complete reset sequence, and the BPMP may need to come in and reset the PM and the system to a known good state of clocks.

NO_OSC_ACK = When PMC has been requested to turn on/off the OSC ACK is obtained, instead the PM FSM times out waiting for an ACK from the PMC.

NO_RET_ACK = When PMC has been requested to put RAMs in/out retention mode, but the PM does not get an ACK, the PM FSM times out waiting for an ACK from the PMC. In this case, PM FSM cannot assume any good state of cluster RAMs (partial sequencing of RAM low power mode is also a possible scenario). Upon narrowing down that the interrupt is due to NO_RET_ACK, software can check the corresponding RAM status bits and also the PG controller status register to infer if there is a hang in the PG controller in the PMC.

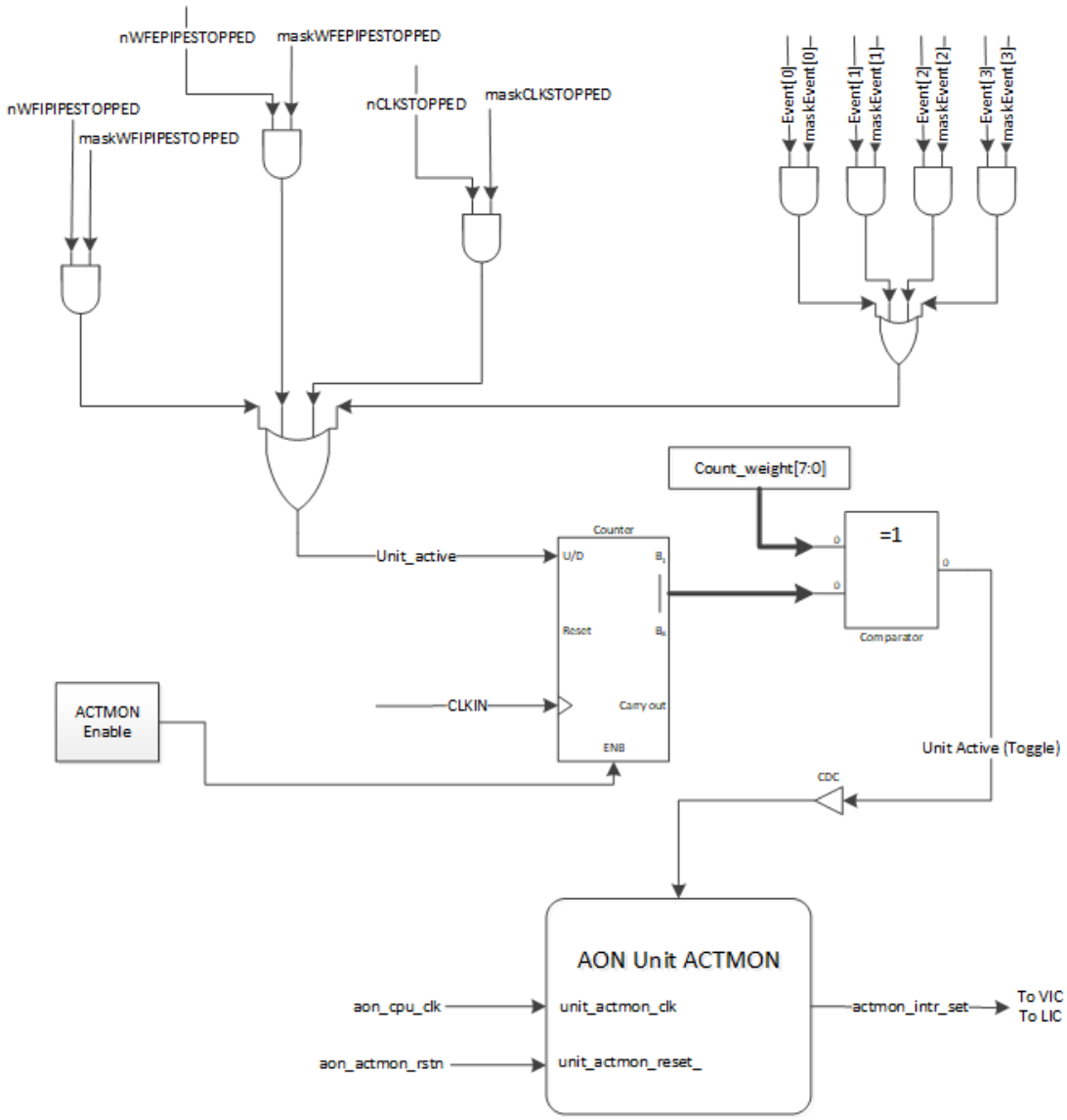
NO_NOC_IDLE_ACK = When NOC_PM has been requested to query power disconnect modules/T-NIUs that there are no pending transactions. If the PM does not get an ACK, the PM FSM timer times out.

NO_NOC_IDLE = When NOC_PM does not give an IDLE indication for the PM dormant state machine to move forward, the PM FSM times out.

8.9.2.15 Cortex-R5 Activity Monitor

The AO-PM has a unit ACTMON. The interrupt from the unit ACTMON is not merged with the PM interrupt. It is separately routed to the local VIC and to LIC. The PM uses this ACTMON to detect idle periods of the processor complex and monitors key events that the Cortex-R5 exposes on its event bus. The idle state of the Cortex-R5 is not exposed to the event bus. Therefore, the idle signals nCLKSTOPPED, nWFIPIPESTOPPED, and nWFEPIPESTOPPED must be separately exported to detect idle period statistics and use the information for scaling the performance of the Cortex-R5.

Figure 8.77 Cortex-R5 Unit ACTMON



Outputs from the Cortex-R5 are sampled at the processor clock frequency and a cumulative counter incremented for every cycle that the unit_active signal is asserted. The cumulative count is compared with a target count. When the cumulative count in processor clock rate reaches the target value, the ACTMON is notified.

The EVENTBUS signals hooked up to the ACTMON are indicative of the following events.

- EVENTBUS[5]: Indicates that the instruction is architecturally executed. Asserted on every cycle when an instruction was architecturally executed by the processor.
- EVENTBUS[16]: Stalls because instruction buffer cannot deliver an instruction. This can indicate an instruction cache miss. Asserted every cycle where the condition is present.

- EVENTBUS[19]: External memory request. Examples of this are cache refill, non-cacheable accesses, write-through writes, cache line evictions (write-back).
- EVENTBUS[20]: Stalls because of Load Store Unit (LSU) being busy. The event takes place each clock cycle where the condition is met. A high incidence of this event indicates the pipeline is often waiting for transactions to complete on the external bus.

8.9.2.16 Programming Model

8.9.2.16.1 AOPM Address Map

This address space is address decoded within PM into PM_IMPL (the functional PM part), the PM_ACTMON (of the unit ACTMON), and the rest is reserved.

Base Address Define	Value	Sub-Aperture Address Define	Value
NV_ADDRESS_MAP_AON_PM_BASE	0xC1F0000	NV_ADDRESS_MAP_AON_PM_IMPL_BASE	0xC1F0000
		NV_ADDRESS_MAP_AON_PM_IMPL_LIMIT	0xC1FFFFF
		NV_ADDRESS_MAP_AON_PM_IMPL_SIZE	0x10000
		NV_ADDRESS_MAP_AON_ACTMON_BASE	0xC200000
		NV_ADDRESS_MAP_AON_ACTMON_LIMIT	0xC20FFFF
		NV_ADDRESS_MAP_AON_ACTMON_SIZE	0x10000
		RESERVED >	0xC210000
NV_ADDRESS_MAP_AON_PM_LIMIT	0xC22FFFF		

8.9.2.16.2 AON Power States Programming Guidelines

SPE- IDLE Entry Sequence

Agent	Step
Cortex-R5 SW	Set up AO-PM with power state transition command by writing to AOPM_TGT_PWR _STATE register. STATE == IDLE TCM_ACTIVE (implicit for IDLE mode) Set up AO-CAR with power state transition info by writing to registers described in the Clocks Chapter to set up the clock source to switch to and the power state of the AOPLL. It could be either of these registers: ▪ TGT_IDLE_SHALLOW ▪ TGT_IDLE_DEEP
Cortex-R5 SW	Execute WFI.

Agent	Step
Clock gating Logic	All second level clock gating controls kick in at this stage except the TCM clocks, backbone clocks for DMA, NOC branches that may be actively used by the DMA during this time along with any active I/O clock(s). Software may additionally FLCG I/O clock sources or switch these to OSC or 32K. Software accordingly programs TGT_IDLE before executing WFI so that AO-PM or AO-CAR hardware does not turn the PLL off when any of the I/O controllers are actively using these clock sources.
AO-DMA	DMA goes idle and gets clock gated at the second level.
AXI-S	AXI-S port goes idle as DMA and I/O controllers go idle.
Cortex-R5	
AO-PM	Engage the second level clock gates on TCM clocks.
AO-PM	Switch Cortex-R5 subsystem clocks to a slower clock.
AO-CAR	Switch Cortex-R5 subsystem clocks to TGT_IDLE_SHALLOW sources if enabled. Place AO-PLL in TGT_IDLE_SHALLOW mode. Acknowledge AO-PM.
AO-PM	Update AOPM_PWR_STATUS register. Set CURRENT_STATE == IDLE_SHALLOW and other fields as per entry sequence.
AO-PM	Check for AOPM_POWER_CFG_MASK_TGT_DEEP_IDLE. If not DISABLE-d, enter DEEP_IDLE state.
AO-PM	Switch Cortex-R5 clocks to TGT_IDLE_DEEP. Place AO-PLL in low power TGT_IDLE_DEEP mode.
AO-CAR	Switch Cortex-R5 subsystem clocks to TGT_IDLE_SOURCE if enabled. Place AO-PLL in TGT_IDLE mode. Acknowledge AO-PM.
AO-PM	Update AOPM_PWR_STATUS register. Set CURRENT_STATE == IDLE_DEEP. Other fields as per entry sequence.

SPE-IDLE Exit Sequence

Once an exit condition is triggered:

Agent	Step
Cortex- R5	Detects interrupt from AVIC or detects request on AXI-S port to come out of Standby mode
AO-PM	Release TCM clock gates within three cycles of nCLKSTOPPED getting deasserted. Detect exit from the SPE-Idle state and capture the cause for wake up in a status register. If the wake up is due to an SC7 wake event: 1. Re-enable the DEEP_IDLE override in hardware. 2. Optionally raise an interrupt to SPE.
AO-PM	Take AOPLL out of low-power mode. Switch Cortex-R5 subsystem clocks back to the mode in which the clocks were running prior to entering this SPE-Idle state.

Agent	Step
AO-CAR	Place AOPLL in TGT_ACTIVE or TGT_ACTIVE_IRQFIQ mode. Switch Cortex-R5 clocks to TGT_ACTIVE or TGT_ACTIVE_IRFIQ source. Acknowledge AO-PM.
AO-PM	Update AOPM_PWR_STATUS register. Set CURRENT_STATE == ACTIVE or ACTIVE_IRQFIQ.

SPE-STANDBY Entry Sequence

Agent	Step
Cortex- R5 SW	Set up AO-PM with power state transition command by writing to: AOPM_TGT_PWR _STATE register. STATE == STANDBY RAM_RET or RAIL_RET enabled. ▪ RAM_RET = RAM ret mode ▪ RAIL_RET = AON rail retention Set RAM_RET and/or RAIL_RET. Set up AO-CAR with power state transition info for clock sources and AOPLL. ▪ TGT_STBY_SHALLOW ▪ TGT_STBY_DEEP
Cortex- R5 SW	Execute WFI.
Cortex- R5 HW	AXI-S port goes idle
AO-PM	Execute the entry steps for entering SPE-Idle. During these steps, AO-PM requests AO-CAR to use TGT_STBY_SHALLOW commands. SPE-Standby entry is stepped through SPE-Idle. Update AOPM_PWR_STATUS register. CURRENT_STATE = STBY_SHALLOW_IDLE Optionally, use the DMA_ACTIVE (MC-side) indicator from the SPE-DMA to infer idleness.
AO-PM	Block TCM accesses from AXI-S port upon expiry of the retention entry delay timer.
AO-PM	Block interrupts from reaching the Cortex-R5 from VIC.
AO-PM	Check if RAM_RET is enabled. Handshakes with PMC.
PMC	Completes RAM retention sequencing and ACKs back to PM.
AO-PM	Update AOPM_PWR_STATUS register with STATE == STBY_SHALLOW.
AO-PM	Check for AOPM_POWER_CFG_MASK_TGT_DEEP_STANDBY. If DISABLE-d, enter DEEP_STANDBY state.
AO-PM	Switch Cortex-R5 clocks to TGT_STBY source. Place AO-PLL in low power mode TGT_STBY.
AO-CAR	Switch Cortex-R5 subsystem clocks to TGT_STBY if enabled. Switch non Cortex-R5 clocks to OSC if they are sourced from AO-PLL. Place AO-PLL in TGT_STBY mode. Acknowledge AO-PM.

Agent	Step
AO-PM	Update AOPM_PWR_STATUS register. Set CURRENT_STATE == STANDBY_DEEP. Other fields as per entry sequence.

SPE-STANDBY Exit Sequence

Agent	Step
Cortex- R5	Detects interrupt from AVIC or detects request on AXI-S port to come out of Standby mode.
AO-PM	Release TCM clock gates within three cycles of nCLKSTOPPED getting deasserted. Detect exit from SPE-Standby state and capture the cause for wake-up in a status register. If the exit condition is due to an SC7 wake request from PMC 1. Re-enable DEEP_STANDBY override in hardware. 2. Optionally raise an interrupt to SPE.
	Before taking the rail out of retention, aon_cpu_clk, aon_nic_clk and aon_apb_clk clocks should be switched back to at least the OSC clock if they were running off 32 kHz. This is to not incur extra latency on the exit path by operating the logic that controls rail voltage at a very slow clock.
AO-PM	Switch aon_cpu_clk, aon_nic_clk and aon_apb_clk back to OSC and wait for acknowledgement from AO-CAR.
AO-CAR	Switch AOPLL and Cortex-R5 clocks to TGT_STBY_SHALLOW. Acknowledge AO-PM.
AO-PM	Handshake with PMC to take TCMs and caches out of retention mode
PMC	Brings RAMs out of retention modes
AO-PM	Switch Cortex-R5 clocks and AOPLL back to their ACTIVE state sources. This step must wait for RAIL_RET exit to complete. AO-PM implements the rail retention exit timer.
AO-CAR	Switch Cortex-R5 clocks and AOPLL to TGT_ACTIVE or TGT_ACTIVE_IRQFIQ mode. Acknowledge AO-PM.
AO-PM	Upon completion of RAIL_RET exit, RAM_RET exit requests, unblock TCM accesses on AXI-S path and unblock interrupts. Do not wait for clocks and AOPLL to switch to TGT_ACTIVE mode.
AO-PM	Upon completion of RAIL_RET exit, RAM_RET exit requests, unblock interrupt. Do not wait for clocks and AOPLL to switch to TGT_ACTIVE mode.
AO-PM	Update AOPM_PWR_STATUS register Set CURRENT_STATE == ACTIVE or ACTIVE_IRQFIQ.

SPE-DORMANT Entry Sequence

Agent	Step
Cortex-R5 SW	Prepare I/O controllers for power-gating mode. For each I/O controller: 1. Execute context save operations after the controller is in idle state. 2. Clock gate the controller. 3. Assert reset to the controller. 4. Switch I/O controller clocks to 32 kHz. Put the CAN message RAMs into SLEEP mode. Flush and disable the L1 caches and place them in SLEEP mode.
Cortex-R5 SW	Set up AO-PM with power state transition command by writing to AOPM_TGT_PWR _STATE register. STATE == DORMANT RAM_RET and/or RAIL_RET is enabled. Set up AO-CAR with power state transition info by writing to registers described in Clocks Chapter TGT_DORMANT_SHALLOW (should be no slower than OSC) TGT_DORMANT (expected to be 32 kHz) (Can optionally set up the Rail retention Control Mode in the PMC.)
Cortex-R5 SW	Execute WFI
Cortex-R5	
AO-PM	Switch Cortex-R5 subsystem clocks to TGT_DORMANT_SHALLOW. Place AO-PLL in low power mode.
AO-CAR	Switch Cortex-R5 subsystem clocks to TGT_DORMANT_SHALLOW if enabled. Place AO-PLL in TGT_DORMANT_SHALLOW mode. Acknowledge AO-PM.
AO-PM	Block accesses at the AXI-S interface. This prevents TCM accesses when the TCM is in retention. Block interrupts at the output of the AVIC. This prevents Cortex-R5 from being woken up when TCM and caches are held in retention.
AO-PM	Put RAMs in RET mode by implementing handshake with the PMC.
PMC	Upon receiving request from AO-PM, the PMC PG Controller puts the AOPG RAMs in RET mode and sends an ACK to AO-PM.
	Before switching the AO-PM clock itself to 32 kHz, the reset and PG sequence, OSC control sequence must be completed. AO-PM first waits for the PMC to go into SC7, executes the below sequences handshaking with PMC, and then switches the clocks to 32 kHz.
AO-PM	Wait for PMC to indicate that SC7 entry is completed.
PMC	De-assert SC7 commit indication once SC7 entry is completed and there is no pending wake event detected for SC7 exit. Wait for AO-PM to deassert AOPM acknowledgement before responding to a subsequent SC7 action.
	The rest of the Dormant state entry sequence holds off SC7 wake events.
AO-PM	Hold off PMC from exiting SC7.
AOPM	AO-PM sends a request to NOC PM to ensure there are no outstanding transactions in NoC.
AON-NOC	Handshakes with T-NIUs in NOC to ensure no outstanding transactions, and sends an ack back to the PM.
AON-NOC	Handshakes with power disconnect modules in NoC and sends a second ACK to the PM.

Agent	Step
AO-PM	Once idle is received from NoC, PM sequences the module resets and module clock enables in preparation for power-gating AOPG domain. The modules that are reset in Hardware are 1. Cortex-R5 resets (nSYSPORESET, PRESETDBG, DBGRESET and nRESET) 2. AON VIC reset 3. AON Fabric resets The module that can be reset only when the system is in SC7 is the 1. AON-NoC resets The AO-PM waits for the PMC to go into SC7 before resetting any module for power-gating.
AO-CAR	Assert requested resets and gate requested clocks and acknowledge AO-PM.
AO-PM	Place the Cortex-R5 in HALT mode until the PG exit sequence is completed
AO-PM	Trigger PG state machine to assert clamp enables and sleep enables for AOPG domain. Wait for completion of PG entry sequence.
AO-PM	Switch Cortex-R5 subsystem clocks to a clock source programmed in TGT_DORMANT_DEEP_CLKSRC and assert AOPLL low power mode.
AO-CAR	Switch Cortex-R5 subsystem clocks to TGT_DORMANT_CLKSRC if enabled and place AOPLL in low power mode defined by this configuration. Acknowledge AO-PM.
AO-PM	Request PMC to turn OSC off.
PMC	Conditionally turn OSC off if enabled by BPMP.
AO-PM	Update AOPM_PWR_STATUS register CURRENT_STATE == DORMANT_DEEP

SPE-DORMANT Exit Sequence

Agent	Step
AO-PM	Detects exit conditions: 1. Pending interrupt (in shallow state) from AVIC outputs 2. Pending wake events from the Wake Engine 3. SC7 indication to exit low power states 4. Software triggered exit from external NoC master access (valid only until AOPG is power gated)
AO-PM	Request for OSC pad to be turned on
PMC	Turn OSC pad on
AO-PM	Switch Cortex-R5 subsystem clocks back to TGT_DORMANT_SHALLOW (required to be OSC). This also switches AOPLL to the corresponding mode.
AO-PM	Requests PMC PG state machine to get RAMs out of RET mode and AOPG domain out of power gating. If the exit condition is due to an SC7 wake request from PMC, then re-enable DEEP_DORMANT override in hardware.

Agent	Step
AO-PM	Once AOPG domain is unpower gated, take the modules out of reset that are not under software reset control. This sends a request to the NOC to do power-ungating steps. Then indicates to the SC7 sequencer that AON exit actions are complete. Once the rail is exited from retention voltage, take the TCMs and SRAMs out of RET mode. Switch AOPLL back to TGT_ACTIVE and switch Cortex-R5 subsystem clocks back to TGT_ACTIVE sources. Wait for ACK from AO-CAR. Request CAR to deassert resets to all hardware-controlled modules. The modules whose resets are removed in hardware are: 1. Cortex-R5 resets (nSYSPORESET, PRESETDBG, DBGRESET and nRESET) 2. AON VIC reset 3. AON Fabric resets
AO-CAR	Switch AOPLL back to TGT_ACTIVE and switch Cortex-R5 subsystem clocks back to TGT_ACTIVE sources. Switch PWR-I2C clock back to TGT_ACTIVE source. Send acknowledgement to AO-PM. Take modules out of reset as requested by AO-PM and send acknowledgement back to AO-PM.
AO-PM	Once TCMs are out of RET mode, unblock the AXI-S.
AO-PM	Once the Cortex-R5 resets are removed and AO-CAR sends an acknowledgement back, and the SRAMs have also been taken out of retention mode, take the Cortex-R5 out of HALT mode.
AO-PM	Update AOPM_PWR_STATUS register CURRENT_STATE == ACTIVE

8.9.2.16.3 AO-CAR

The block called AO-CAR is the CAR parts that reside in the AO voltage domain. AO-CAR provides the clocks and resets to both the AON cluster and the other blocks that are inside the AO voltage domain.AO-CAR

8.9.2.16.4 DMA Controller

The AON cluster DMA controller is an eight-channel implementation of the central DMA controller. The DMA request from the AON Cluster (I2C[2:0], SPI, UART, and DMIC) are connected to the local DMA controller. Refer to the General Purpose DMA (GPC-DMA, AON, BPMP, and SCE DMA) chapter in this TRM for more details.

Table 8.88 DMA Request Mapping

Block	TX DMA Request Port (DMA_TX [31:0])	RX DMA Request Port (DMA_RX [31:0])
Reserved	0	0
I2C2 (AON_I2C2)	1	1
I2C8 (AON_I2C3)	2	2
SPI2 (AON_SPI1)	3	3

Block	TX DMA Request Port (DMA_TX [31:0])	RX DMA Request Port (DMA_RX [31:0])
UARTC (AON UARTC)	4	4
UARTJ (SBSA_UART)	5	5
DMIC	N/A	6
GTE	N/A	7
UART6 [from SoC for closed system debug]	8	8
UART8 [from SoC for closed system debug]	9	9

8.9.2.16.5 Address Translation

The AON Cluster is required to support sending requests to DRAM that indicate to the SMMU if the request requires translation to support a physical carve-out. The SMMU needs to handle requests differently based on if a translation is requested.

Since the Cortex-R5 is not capable of indicating if a request requires translation, an Address Space Translation Engine (AST) is used to define the address regions that require translation. The AST is also used to remap the Cortex-R5 and DMA generated address to prevent address space collision for I/O virtual addresses. The AON cluster includes two eight region AST modules. Refer to Address Space Translation (AST) for more information.

The AON Cluster is only allowed to bypass SMMU translation when accessing the AON Cluster-only carve-out. If a translation is not requested, the SMMU must bypass the address translation and ensure the address is inside the AON Cluster DRAM region. If a translation is requested, the SMMU performs the translation and ensures that the translated address is not in the AON Cluster region (or any other protected region).

8.9.2.16.6 Shared Mailboxes

The AON cluster uses shared MMIO mailbox registers to communicate between the Cortex-R5 other processors in the system. The mailbox implementation is defined in the Hardware Synchronization Primitives (HSP) chapter of this TRM. The HSP in the AON cluster uses the following configuration i.e., eight Shared Mail Box Instances (nSM) and four Shared Semaphore (nSS) instances. The number of Shared interrupts i.e., nSI is five.

nSM	nAS	nSS	nDB	nSI	Notes
8	0	4	0	5	SI[0] is sent to the AVIC. [SI[4:1] are sent to the LIC.

8.9.2.16.7 Timekeeping Engine (TKE)

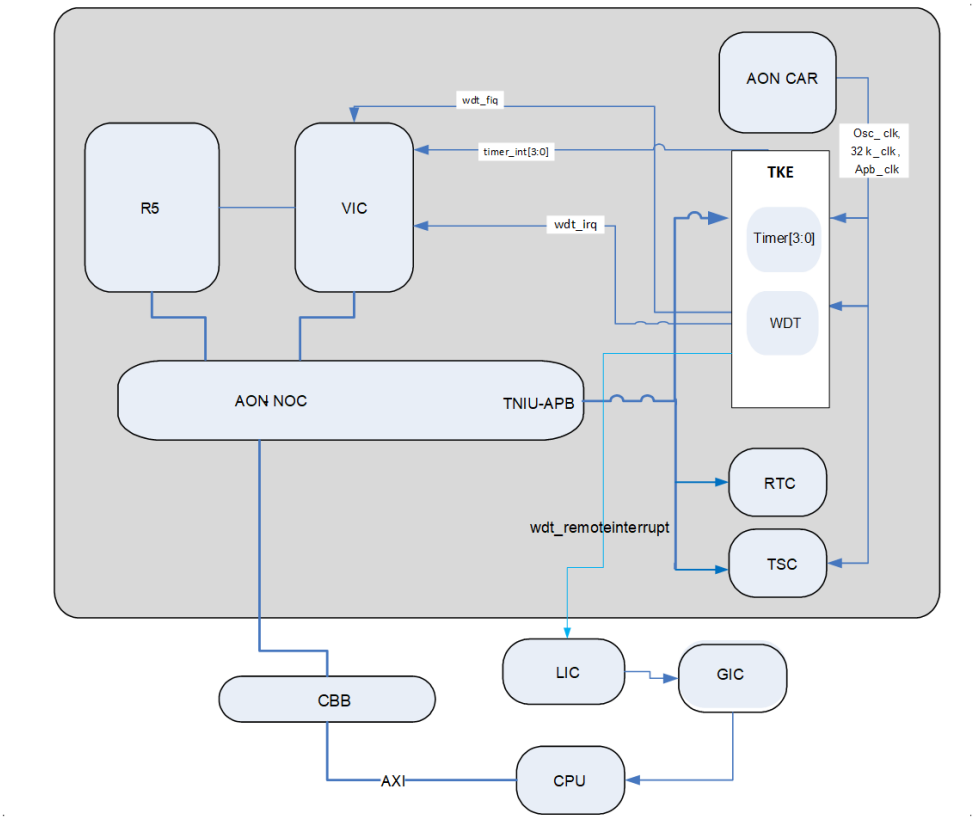
Cortex-R5 does not have built-in timers, the timers are integrated into the BPMP subsystem with an external Timekeeping Engine (TKE). The timekeeping unit provides the following functionalities:

1. Timers (NV_timers)
2. Watchdog (WDT)
3. Timestamp counter (TSC) read-back

Table 8.89 AON TKE Configuration

Parameter	Value
Number of NV_timers (nT)	4
Number of watchdogs (nW)	1
Interrupt routing	Disabled
Number of shared interrupts exposed	0
TSC input type (isSerial)	1 (Serial)

Figure 8.78 AON Timers and WDT Hookup



Watchdog Timer

A watchdog timer associated with the AON Cortex-R5 is instanced to monitor Cortex-R5 CPU's health. The watchdog timer has robust restart mechanisms like windowed watchdog and Question and Answer watchdog. AON cluster has a dedicated watchdog timer instanced in the cluster. This enables us to architect the functionality of this timer as per AON clusters specific requirements.

8.9.2.16.8 Timestamping (GTE)

The AON cluster implements a timestamp module (GTE) for logging the current TSC value when interrupts or GPIO events occur.

8.9.2.16.9 GTE and DMA

The AON DMA engine can be used to read from the GTE timestamp FIFO instead of reading the FIFO using the Cortex-R5. One use case is for the GTE to timestamp the GPIO changes to recover a waveform (for example handling IR receive). To use GTE and DMA in this manner, the following configuration must be used.

GTE:

- GTE_GPIO_TECTRL_0[AUTOADVADDR] = 0x3
- GTE_GPIO_TECTRL_0[OCCUPANCYTHRESHOLD] = 0x1
- GTE_GPIO_TECTRL_0[INTERRUPTENABLE] = 0x1

DMA:

- GPCDMA_FLV_8CH_NON_SAFE_CHANNEL_CH<X>_MMIO_SEQ_0[MMIO_BURST] = 0x7
- GPCDMA_FLV_8CH_NON_SAFE_CHANNEL_CH<X>_MMIO_SEQ_0[MMIO_ADDR_WRAP] = 0x4
- GPCDMA_FLV_8CH_NON_SAFE_CHANNEL_CH<X>_SRC_PTR_0[SRC_PTR] = GTE base address
- GPCDMA_FLV_8CH_NON_SAFE_CHANNEL_CH<X>_MC_SEQ_0[MC_BURST] = 0x0
- GPCDMA_FLV_8CH_NON_SAFE_CHANNEL_CH<X>_DMA_BYTE_TRA_0[TRANSFER_COUNT] = 8x4x (number of samples)

When using this configuration, the GTE interrupt in the AVIC must be masked.

8.9.2.16.10 I2C

The AON cluster implements two I^2C controllers.

Table 8.90 AON I2C Controllers

Global AMAP Name	Key Usage Examples
I2C2	Camera configuration, Touch control, Sensor interfaces
I2C8	Display Aggregator, Sensor Hub Control

8.9.2.16.11 SPI

The AON cluster implements one SPI controller.

Table 8.91 AON SPI Controller

AON SPI Instance	Global AMAP Name
AON_SPI	SPI2

8.9.2.16.12 UART

The AON cluster implements two UARTs for primary debug purposes. UARTC is an instance of the standard UART IP, but is implemented as a two pin UART with the CTS and RTS pins tied off so the IP can always send and receive data. UARTJ is the SBSA Compliant UART for debug purposes. See Universal Asynchronous Receiver/Transmitter (UART)

Table 8.92 AON UARTs

Global AMAP Name	UART Type	Usage Details
UARTC	2pin	Primary Two Pin Debug UART. Can also be controlled by BPMP and CCPLEX with interrupts routed to BPMP cluster and LIC. This is the case for flow control signals also, i.e., routed to BPMP Cluster DMA and GPC DMA.
UART_J	2 pin	SBSA Compliant UART.

8.9.2.16.13 PWM

The AON cluster implements one PWM. The details of the PMW can be found in the Pulse Width Modulator (PWM) chapter of this document.

8.9.2.16.14 GPIO Controllers

The AON cluster implements a GPIO controller and pinmux module. Both modules have separate APB interfaces. There are four interrupts (two functional and two time-stamp interrupts) from the GPIO controller. One functional interrupt and one timestamp interrupt is sent from the GPIO block to the AON interrupt controller. All functional and timestamp interrupts is sent to LIC.

8.9.2.16.15 DMIC

The AON cluster implements a DMIC controller for supporting voice wake functionality. The AON DMIC controller is optimized for low power voice wake requirements.

8.9.2.16.16 CEC

The Consumer Electronic Controller (CEC) unit is added in the AO domain. The motivation for adding the controller to AON is to be able to wake the system up from SC7 only on messages indicated to the SoC and not to wake up on unintended or spurious toggles of the input (wake capable pad).

8.9.3 AON Programming Guidelines

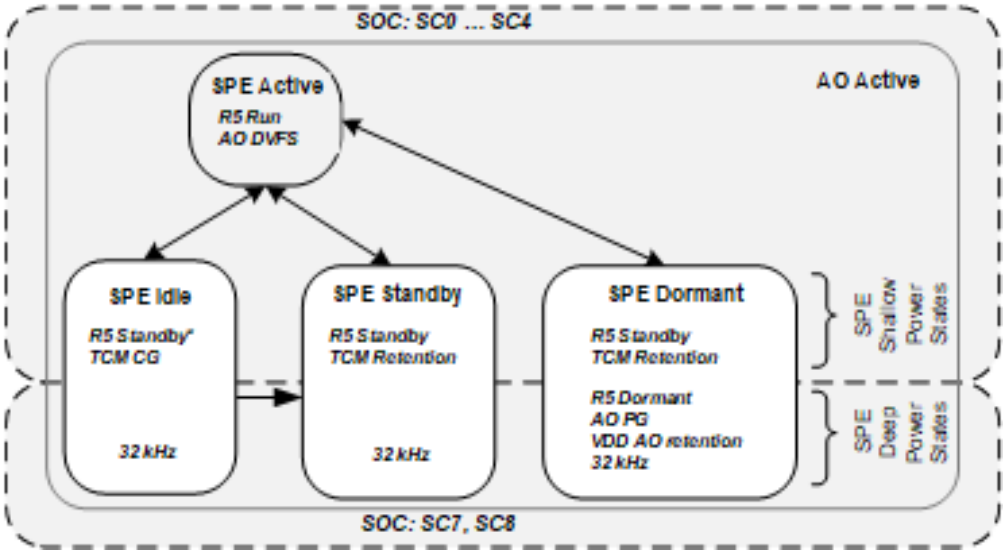
8.9.3.1 Power Management

8.9.3.1.1 Cluster Power State Definitions

The AO Power State framework consists of SPE power states on one hand and the cluster's activity levels mapped to System Power States on the other. The SPE subsystem in the AO cluster is in one of four power states:

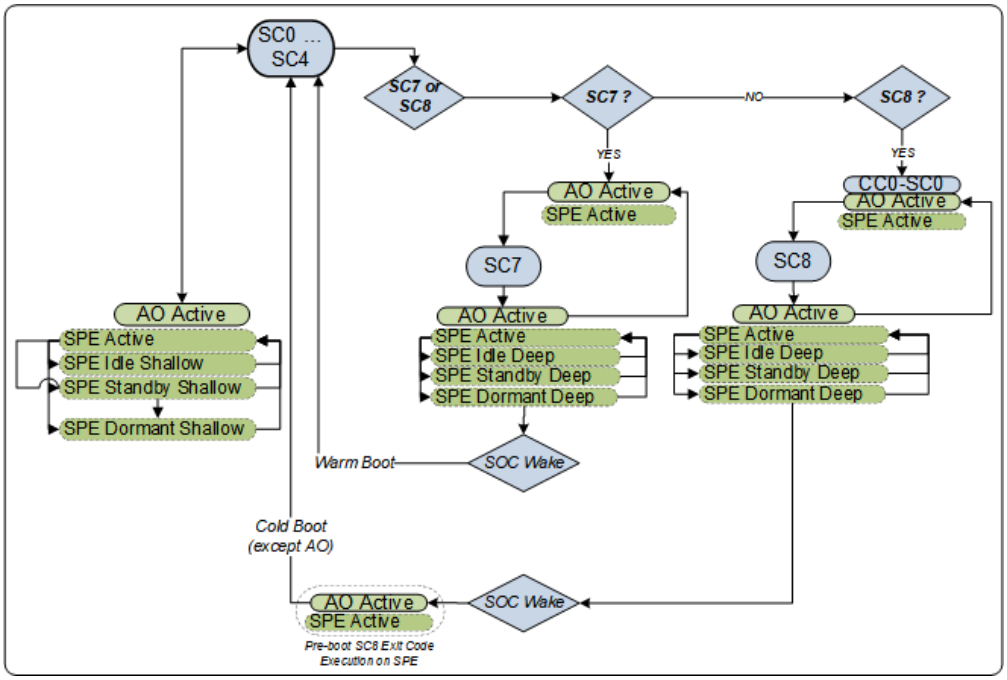
1. SPE-Active
2. SPE-Idle
3. SPE-Standby
4. SPE-Dormant

Figure 8.79 SPE Power States Hierarchy within System Power States Framework



These power states are defined by the sensor processor and sensor I/Os expected activity levels. The system and power management functions in the AO cluster operate at various levels of activity but are never switched off completely except when the SoC is fully powered off (shut down). At the cluster level there is no specific idle or standby state defined. The AO cluster is an always-on domain in the chip. In any of the SoC power states, the AO cluster could be in any of its own power states. However, there are some constraints on the AO cluster state while the chip transitions into and out of the SC7 SoC states.

Figure 8.80 AO Rail Power States Transition Framework



In the steady state, any of the AO cluster power states (except complete power-off) are supported by hardware in any of the SoC power states.

8.9.3.1.2 Cortex-R5 CPU Halted (Clock Gating)

If the CPU is idle for a short period of time during WFI state, then it can be halted which results in its CPU clock-gating. The WFI instruction is executed for putting CPU into halt. After executing WFI, CPU disables most of the clocks in the CPU except the clock for the small wake up logic clock. Exit from halt state occurs when the CPU detects an interrupt on IRQ or FIQ pins.

8.9.3.1.3 TCM Clock Gating

The TCM interface clock can be gated when there are no pending DMA requests and the Cortex-R5 is halted. The nCLKSTOPPED signal is used to indicate when the TCM clock can be gated. The TCM clock must be restarted in no more than four clocks after nCLKSTOPPED is deasserted.

8.9.3.1.4 OSC Power Down

To reduce SC7 power when the AON Cluster is idle for an extended period (greater than 3 ms), all AON Cluster clocks except for the 32 kHz clock can be disabled by powering down the OSC pad. When the OSC is powered down, only the RTC, or an external event, can wake the AON Cluster.

8.9.3.1.5 Software Enable Disable of AON PLL

The AON cluster clocking logic does not allow the AON Cortex-R5 to directly turn on the AON cluster PLL. When the SoC is not in SC7, the AON Cortex-R5 can make a request to BPMP to enable or disable the PLL (normally the AON PLL is on when the SoC is on). When the SoC is in SC7, the BPMP is not powered. If the AON Cortex-R5 requires the PLL to turn on and off in SC7, the Cortex-R5 firmware must use the ACTIVE and ACTIVE_IRQFIQ power states defined in AOPM. The AON Cortex-R5 can use these two states to define a state with the PLL on (ACTIVE_IRQFIQ) and a state with the PLL off (ACTIVE).

The following conditions must be met for AON firmware to control the PLL state in SC7.

- No I/O controllers can be configured to use a clock from the AON PLL.
- The BPMP must configure the ACTIVE_IRQFIQ and ACTIVE clock settings prior to entering SC7.
- Prior to executing WFI firmware must ensure that any previous software request is complete by
  a. [AON-R5-FW] Read AON_SW_TGT_REQ. If 1, go to Step 2, else go to Step 4.
  b. [AON-R5-FW] Poll AON_SW_TGT_ACK until it is 1 in the CLK_RST_CONTROLLER_AON_SEQ_SW_INTERFACE_CTL_0 register.
  c. [AON-R5-FW] Program AON_SW_TGT_REQ to 0 in the CLK_RST_CONTROLLER_AON_SEQ_SW_INTERFACE_CTL_0 register.
  d. [AON-R5-FW] Poll AON_SW_TGT_ACK in the CLK_RST_CONTROLLER_AON_SEQ_SW_INTERFACE_CTL_0 register until it is set to 0 by hardware.

ACTIVE to ACTIVE_IRQFIQ Firmware Sequence

Entering the ACTIVE_IRQFIQ state typically requires an interrupt to be triggered. To prevent the PLL from being turned on every time an interrupt occurs, the ACTIVE_IRQFIQ state should be disabled:
(AOPM_BURST_CLK_0[BURST_CLK_EN]= 0 and AOPM_BURST_CLK_0[BURST_LP_EN]= 0).

1. [AON-R5-FW] Read AON_SW_TGT_REQ. If 1, go to Step 2, else go to Step 4.
2. [AON-R5-FW] Poll AON_SW_TGT_ACK until it is 1 in the CLK_RST_CONTROLLER_AON_SEQ_SW_INTERFACE_CTL_0 register.
3. [AON-R5-FW] Program AON_SW_TGT_REQ to 0 in the CLK_RST_CONTROLLER_AON_SEQ_SW_INTERFACE_CTL_0 register.
4. [AON-R5-FW] Poll AON_SW_TGT_ACK in the CLK_RST_CONTROLLER_AON_SEQ_SW_INTERFACE_CTL_0 register until it is set to 0 by hardware.
5. [AON-R5-FW] Program the AON_SW_TGT_STATE field in the CLK_RST_CONTROLLER_AON_SEQ_SW_INTERFACE_CTL_0 register to 0x1.

6. [AON-R5-FW] Program the AON_SW_TGT_REQ bit in the CLK_RST_CONTROLLER_AON_SEQ_SW_INTERFACE_CTL_0 register to 1.
7. [CAR HW] Transition to ACTIVE_IRQFIQ state and set AON_SW_TGT_ACK when complete.

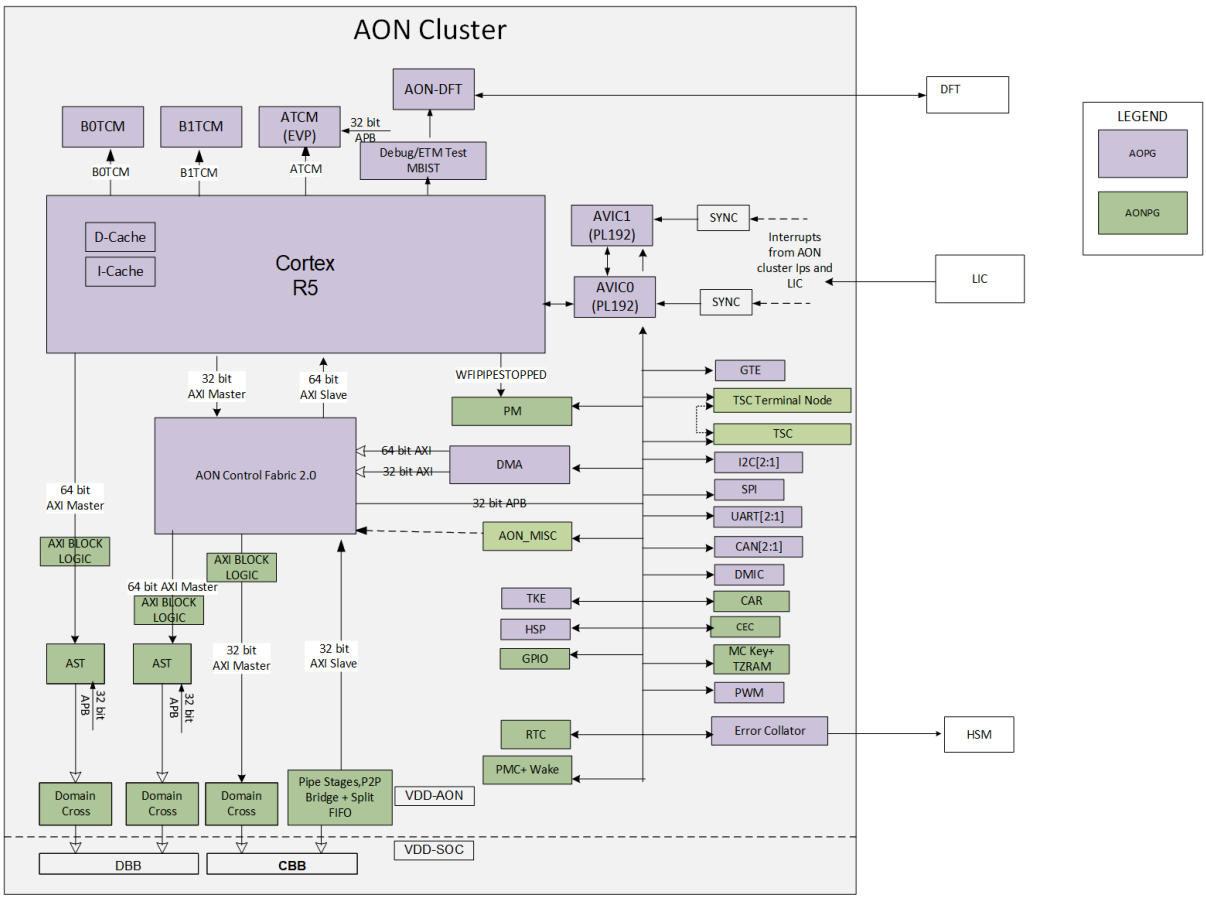
ACTIVE_IRQFIQ to ACTIVE Sequence

1. [AON-R5-FW] Read AON_SW_TGT_REQ. If 1, go to Step 2, else go to Step 4.
2. [AON-R5-FW] Poll AON_SW_TGT_ACK until it is 1 in the CLK_RST_CONTROLLER_AON_SEQ_SW_INTERFACE_CTL_0 register.
3. [AON-R5-FW] Program AON_SW_TGT_REQ to 0 in the CLK_RST_CONTROLLER_AON_SEQ_SW_INTERFACE_CTL_0 register.
4. [AON-R5-FW] Poll AON_SW_TGT_ACK in the CLK_RST_CONTROLLER_AON_SEQ_SW_INTERFACE_CTL_0 register until it is set to 0 by hardware.
5. [AON-R5-FW] Program the AON_SW_TGT_STATE field in the CLK_RST_CONTROLLER_AON_SEQ_SW_INTERFACE_CTL_0 register to 0x0.
6. [AON-R5-FW] Program the AON_SW_TGT_REQ bit in the CLK_RST_CONTROLLER_AON_SEQ_SW_INTERFACE_CTL_0 register to 1.
7. [CAR HW] Transition to ACTIVE state and set AON_SW_TGT_ACK when complete.

8.9.3.1.6 AON Cluster Power Gating

The AON cluster supports power gating most of the units within the cluster to minimize the AON power to meet the Automotive CAN standby power targets. PMC also has control register for power gating the Cortex-R5 cache RAMs, the BTCM RAMs, and the CAN controller RAMs. The figure below shows the expected PG control connections in AON.

Figure 8.81 NPG vs PG view of AON IPs



Gating Entry Sequence

1. [AON-R5-FW] Detects an idle condition when system in SC7 that can tolerate greater than 10 ms of wake time.
2. [AON-R5-FW] Enables wake detection on for AON external wake sources in PMC.
3. [AON-R5-FW] Asserts Reset to each power gateable I/O controller.
4. [AON-R5-FW] Stops first level clocks to each power gateable I/O controller.
5. [AON-R5-FW] Save any required core logic state (Cortex-R5, AVIC, TKE, GTE, HSP) state.
6. [AON-R5-FW] Asserts reset to TKE, GTE, HSP.
7. [AON-R5-FW] Stops first level clocks to TKE, GTE, HSP.
8. [AON-R5-FW] Sets the target power state to DORMANT in AO-PM.
9. [AON-R5-FW] Executes WFI which triggers power gating for the AO cluster logic including the CAN controllers.
10. [PMC] Waits for external wake event.

Power Gating Exit Sequence

1. [PMC] Wake received and triggers AO-PM.
2. [AO-PM] Restores AO Cluster to Active state.
3. [AON-R5-FW] Enable first level clocks to TKE, GTE, HSP.
4. [AON-R5-FW] Deasserts reset to TKE, GTE, HSP.
   a. Reset to TKE should be removed after GTE reset if GTE time stamping is used.
   b. See the Timers chapter for details on resyncing the GTE and TKE with the TSC after exiting PG.
5. [AON-R5-FW] Restore required state to core logic.
6. [AON-R5-FW] Enables first level clocks to each power required I/O controllers.
7. [AON-R5-FW] Deasserts Reset to each required I/O controller.
8. [AON-R5-FW] Initialize required I/O controllers for normal operation.

Cortex-R5 Cache Power Gating

When the system enters SC7, the AON Cortex-R5 can no longer access DRAM. Prior to SC7 entry the AON Cortex-R5 must flush and disable its caches. To save leakage the Cortex-R5 caches can be power-gated after the caches are disabled.

Power Gating Entry Sequence

1. [SW-AP] Decides to go into SC7.
2. [SW-AP] Notifies AON Cluster firmware of SC7 Entry.
3. [FW-AON R5] Flushes L1 data cache and write buffers.
4. [FW-AON R5] Disables the instruction and data caches.
5. [FW-AON R5] Programs AON Cortex-R5 Cache Power-gating control register in PMC.
6. [HW-PMC] Power gates Cortex-R5 cache by asserting the cache SD pins.
7. [FW-AON R5] Notifies AP/BPMP that it is safe to enter SC7.

Power Gating Exit Sequence

1. [FW-BPMP] Notifies AON Cluster firmware that system is out of SC7.
2. [FW-AON R5] Programs AON Cortex-R5 Cache Power-gating control register in PMC.
3. [HW-PMC] Deasserts the SD pins to the Cortex-R5 cache RAMs.
4. [FW-AON R5] Enables the instruction and data caches.
5. [FW-AON R5] Performs invalidate all to initialize the instruction and data caches.

BTCM Power Gating

The BTCM for the AON Cortex-R5 can enter power gating by:

1. Software initiated RAM Shutdown
2. Hardware initiated RAM Sleep or Deep Sleep

3. Software initiated RAM Sleep or Deep Sleep

BTCM RAM Shutdown

Software initiated RAM shutdown is controlled from PMC. PMC contains one power-gating control register for each BTCM bank. When a bank is placed in SD, the BTCM logic must return an error is the bank is accessed. Software is responsible for saving any data in the TCM bank to DRAM if state needs to be maintained.

Power Gating Entry Sequence

1. [FW-AON R5] Decides to go put TCM bank in SD
2. [FW-AON R5] Copies any required data to DRAM.
3. [FW-AON R5] Programs AON TCM Bank N Power-gating control register in PMC.
4. [HW-PMC] Power gates Cortex-R5 cache by asserting the cache SD pins.

Power Gating Exit Sequence

1. [FW-AON R5] Determines a TCM Bank needs to be enabled
2. [FW-AON R5] Programs AON TCM Bank N Power gating control register in PMC.
3. [HW-PMC] Deasserts the SD pins to the Cortex-R5 cache RAMs
4. [FW-AON R5] Restores TCM data from DRAM if required.

CAN Power Gating

The two CAN controllers are power gated at the same time the AON cluster core logic is power gated.

CAN Power Gating Entry Sequence

1. [CAN-SW] Detects idle condition (involves detecting idle condition of network and controller) and system in SC7.
2. [CAN-SW] Puts external Transceiver in Low power (or sleep or listen only) mode.
3. [CAN-SW] Enables Wake detection on CAN WAKE (can be Rx pad for some transceivers) Pad in PMC.
4. [CAN-SW] Asserts Reset to CAN controller.
5. [CAN-SW] Stop CAN[1:2] first Level Clocks.
6. [CAN-SW] Writes to the CAN[1:2] Power-gating control registers in PMC to power gate the CAN message RAM.
7. [CAN-SW] Save any required Cortex-R5 and AVIC state.
8. [CAN-SW] Sets the target power state to DORMANT in AO-PM.
9. [CAN-SW] Executes WFI which triggers power gating for the AO cluster logic including the CAN controllers.
10. [PMC] Waits for Wake event from CAN Network.

CAN Power Gating Exit Sequence

1. [PMC] Wake received and triggers AO-PM.
2. [AO-PM] Restores AO Cluster to Active state.
3. [CAN-SW] Restore required state to Cortex-R5 and AVIC.
4. [CAN-SW] Writes to the CAN[1:2] Power-gating control registers in PMC to ungate the CAN message RAM.
5. [CAN-SW] Enable Clocks.
6. [CAN-SW] Deassert reset.
7. [CAN-SW] Disable Wake on CAN receive pad.
8. [CAN-SW] Bring transceiver out of low power mode.
9. [CAN-SW] Initialize controllers for normal operation.

8.9.3.2 AON Cluster SC7 Entry

Access from the AON cluster to the SOC is blocked by the software when entering SC7. If requests are in flight when the block logic is engaged it is possible a deadlock could occur. Software should follow the sequence below to ensure this deadlock does not happen.

1. BPMP notifies AON FW to stop all SOC access (MMIO and DRAM) via an IPC message
2. AON firmware performs all necessary steps to be able to operate with no SOC access. (Swap code into TCM, flush, and Disable caches, etc.)
3. After this point AON firmware is executing only from TCM.
4. AON firmware sends IPC to BPMP that it is safe to disable SOC access
5. BPMP acknowledges IPC by writing to HSP mailbox in AON.
6. AON FW detects AON mailbox write and acknowledges by writing to AON HSP.
7. BPMP polls AON HSP waiting for the acknowledgment from AON. When the acknowledgment is received, PSC writes to AOPM to enable the block function.

8.9.3.3 AON Cortex-R5 Activity Monitor

The unit activity monitor is a part of the AO-PM module. The interrupt from the unit ACTMON is not merged with the AO-PM interrupt. It is separately routed to the local VIC and to LIC.

AON uses a type-1 unit ACTMON [refer to the Activity Monitor (ACTMON) chapter of this document] to detect idle periods of the processor complex and monitor a few key events that the Cortex-R5 exposes on its event bus. The idle state of the Cortex-R5 is not exposed to the event bus. Therefore, the idle signals nCLKSTOPPED, nWFIPIPESTOPPED, and nWFEPIPESTOPPED must be separately exported to detect idle period statistics and use the information for scaling the performance of the Cortex-R5.

The outputs from the Cortex-R5 are sampled at the processor clock frequency and a cumulative counter incremented for every cycle that the unit_active signal is asserted. The cumulative count is

compared with a target count. When the cumulative count in processor clock rate reaches the target value, the ACTMON is notified.

The EVENTBUS signals hooked up to the ACTMON are indicative of the following events.

- EVENTBUS[5]: Indicates that the instruction is architecturally executed. Asserted on every cycle when an instruction was architecturally executed by the processor.
- EVENTBUS[16]: Stalls because instruction buffer cannot deliver an instruction. This can indicate an instruction cache miss. Asserted every cycle where the condition is present.
- EVENTBUS[19]: External memory request. Examples of this are cache refill, non-cacheable accesses, write-through writes, cache line evictions (write-back).
- EVENTBUS[20]: Stalls because of Load Store Unit (LSU) being busy. The event takes place each clock cycle where the condition is met. A high incidence of this event indicates the pipeline is often waiting for transactions to complete on the external bus.

The unit ACTMON uses aon_cpu_clk as the refclk.

8.9.3.4 Reset and Clocking

8.9.3.4.1 Reset

There are several independent reset domains in the AON Cluster to be able to reset independently different parts of the Cortex-R5 and AON Cluster.

Cortex-R5 Reset

Cortex-R5 can start executing instructions as soon as its reset is released (deasserted). Hence, it is important to ensure that the entire system is ready to accept instructions before Cortex-R5 is released out of reset.

Table 8.93 Cortex-R5 Reset Domains

Reset Domain	Description
ACPRESETn	Since Cortex-R5 subsystem doesn't have an ACP, this is tied inactive (to 1)
nRESET	CPU non-debug logic reset. Notes: Whenever nRESET is asserted, it should be asserted for at least four CLKIN cycles for correct reset operation. This is the signal that is pulsed by the watchdog timer on a "processor hang" if debug through reset is needed.
nSYSPORESET	Cortex-R5 system reset This resets the Cortex-R5 processor and debug logic. Note: Whenever nSYSPORESET is asserted, it should be asserted for at least four CLKIN cycles for correct reset operation.
nPRESETDBG	Cortex-R5 Debug reset. Resets debug-domain debug logic and the APB interface of the CPU.

Reset Domain	Description
nCPUHALT	Prevents the Cortex-R5 from fetching code out of reset.
DBGRESETn	Cortex-R5 Debug reset. Resets core-domain debug logic. This includes breakpoints, watchpoints, and the DCC registers.

AON Cluster Reset

There are four groups of resets in the AON cluster as shown in the table below. They are grouped so that the reset assertion and deassertion can be understood in terms of groups rather than a bunch of independent signals.

Table 8.94 Reset Groups in AON Cluster

Group 0	Group 1	Group 2	Group 3
System reset + All AON cluster resets, with the exception of Groups 1, 2, and 3 resets	CAN1 and CAN2 reset	nSYSPORRESET, VIC0, VIC1	nCPUHALT

The order of deassertion of resets on cold boot is from Group 0 through Group 3.

1. Group 0 reset is the system reset. This reset is the system reset and is generated by PMC. This reset is the reset for most modules in the AON cluster.
2. Group 1 reset is the Group 0 reset qualified by the fuses valid signal, straps valid signal and RAM valid signal. This reset goes to the CAN controllers. DBGRESETn, nPRESETDBG, and nRESET may be extended up to four clocks after the Group 1 reset deasserts.
3. Group 2 reset group is primarily the Cortex-R5 Power-on-Reset and the PL192 resets. This is the last module in the AON cluster that comes out of reset. The rest of the cluster is ready to receive and process Cortex-R5 CPU's requests as soon as Cortex-R5 is released. The Group 2 reset is the same as Group 1 reset except it may be extended up to four clocks.
4. Group 3 reset group includes the nCPUHALT. This isn't truly a processor reset, but can be used to hold processor from executing instructions. For the AON cluster nCPUHALT should remain asserted until firmware has been loaded into the TCM. To do this as FWLoadDone bit is set by the boot code once the TCM load is complete. The FWLoadDone bit is not cleared except by an aon_cluster_reset.

9. I/O Controllers and Interfaces

9.1 High-Speed I/O Cluster

9.1.1 Overview

The High-Speed I/O (HSIO) Cluster in the NVIDIA® Orin™ series System-on-Chip (SoC) is a collection of high-speed I/O controllers that support lane mapping and can be configured as different I/O interfaces based on different use cases. The high-speed I/O controllers share up to twenty-four high-speed multi-mode UPHY physical interface modules, organized as up to three 8-lane groups (also referred to as bricks):

- An 8-lane HSIO UPHY group shared between high-speed I/O controllers of PCIe, UFS, XUSB, and HSSTP.
- An 8-lane NVHS UPHY group used for PCIe high-speed I/O controllers.
- An 8-lane GbE UPHY group shared between high-speed I/O controllers of PCIe and 10 Gigabit Ethernet.

Note that some configurations of the Orin family offer a subset of the features described here.

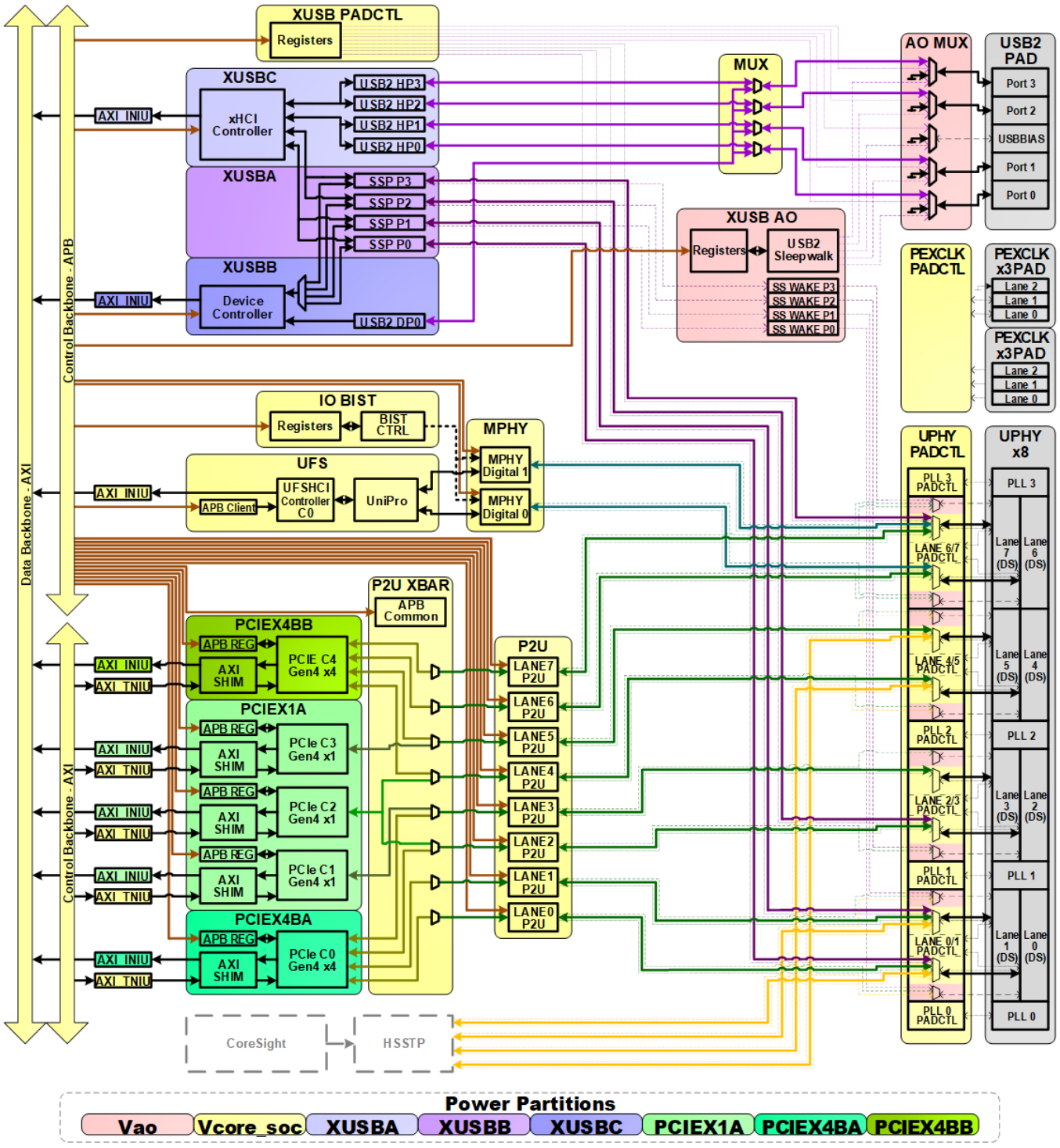
9.1.1.1 HSIO UPHY Group

The HSIO UPHY group supports the following high-speed I/O controllers that share the 8-lane UPHY brick, note that separate USB2.0 pads are used by XUSB.

- PCIe
  - Three PCIe x1 root port controllers
    - Supports Gen1, Gen2, Gen3, and Gen4 link speeds
  - Two PCIe x4 dual-mode controllers (restricted to root port mode)
    - Supports root port mode only
    - Supports lane reversal - link may be trained to x1, x2, or x4 based on connection to device
    - Supports Gen1, Gen2, Gen3 and Gen4 link speeds
    - PCIe C4 can be configured as boot storage controller with NVMe endpoint connected
  - PCIe consists of three power-gate-able partitions and one un-power-gate-able partition
    - Each PCIe x4 in its own partition that can be individually power gated when it is unconnected or in L2 link power states.

        - Three PCIe x1 controllers in one partition that can only be power gated when all three PCIe x1 controllers are unconnected or in L2 link power states
        - P2U XBAR and all PIPE2UPHY are in an un-gated power partition
- UFS
    - One UFS 3.0 x2 host controller
        - Supports HS-GEAR1/2/3/4 in both RATE A/B link speeds
        - Support x1 or x2 link width
        - Can be configured as boot storage controller
    - UFS is in an un-gated VDD_SOC partition
- USB
    - One xHCI host controller
        - Supports four USB3.2 Gen1/Gen2 x1 SuperSpeedPlus ports
        - Supports four USB2.0/USB1.1 High/Full/Low-Speed ports
    - One USB3.2 Gen1/Gen2 x1 device controller
        - Supports one USB3.2 Gen1/Gen2 x1 SuperSpeed port or one USB2.0 port
        - Can be configured as recovery mode controller under either USB2.0 or 2 Gen1/Gen2 x1 speed
    - XUSB consists of three power partitions that can be separately power gated
        - Host controller with its USB2.0 logic
        - Device controller with its USB2.0 logic
        - All SuperSpeedPlus port logic
- HSSTP
    - One HSSTP controller that supports debug stream output
        - Supports 2.5 Gbps, 5 Gbps, and 10 Gbps link speed
        - Supports x2 with Tx output when use both Lane 0 and Lane 1
        - Supports x1 with Tx output when use Lane 0, Lane 1, Lane 4, or Lane 5
    - HSSTP controller resides outside HSIO cluster with its signals connect to UPHY PADCTL

Figure 9.1 HSIO Block Diagram for Controllers Sharing HSIO UPHY

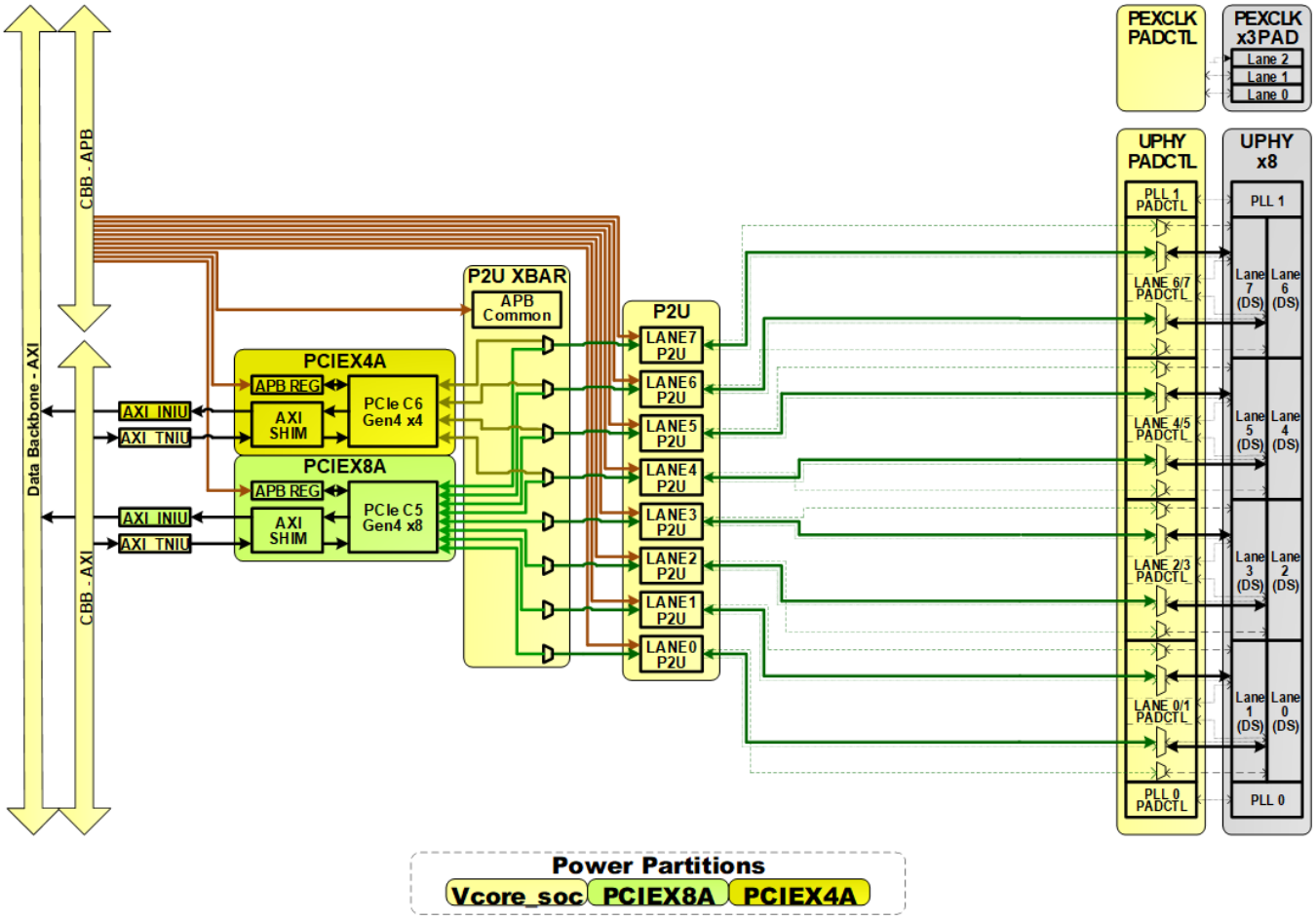


9.1.1.2 NVHS UPHY Group

The NVHS UPHY group supports the following high-speed I/O controllers that share the 8-lane UPHY brick.

- PCIe
  - One PCIe x8 dual-mode controller
    - Configures root port or endpoint mode
    - Supports Gen1, Gen2, Gen3, and Gen4 link speeds
    - Supports lane reversal - link may be trained to x1, x2, x4, or x8 based on connection to device
  - One PCIe x4 dual-mode controller
    - Configures root port or endpoint mode
    - Supports Gen1, Gen2, Gen3, and Gen4 link speeds
    - Supports lane reversal - link may be trained to x1, x2, or x4 based on connection to device
  - PCIe consists of two power-gate-able partitions and one un-power-gate-able partition
    - One PCIe x4 controllers in its own partition that can be individually power gated when it is unconnected or in L2 link power state
    - One PCIe x8 controller in its own partition that can be individually power gated when it is unconnected or in L2 link power state
    - P2U XBAR and all PIPE2UPHY in one un-power-gate-able partition

Figure 9.2 HSIO Block Diagram for Controllers Sharing NVHS UPHY



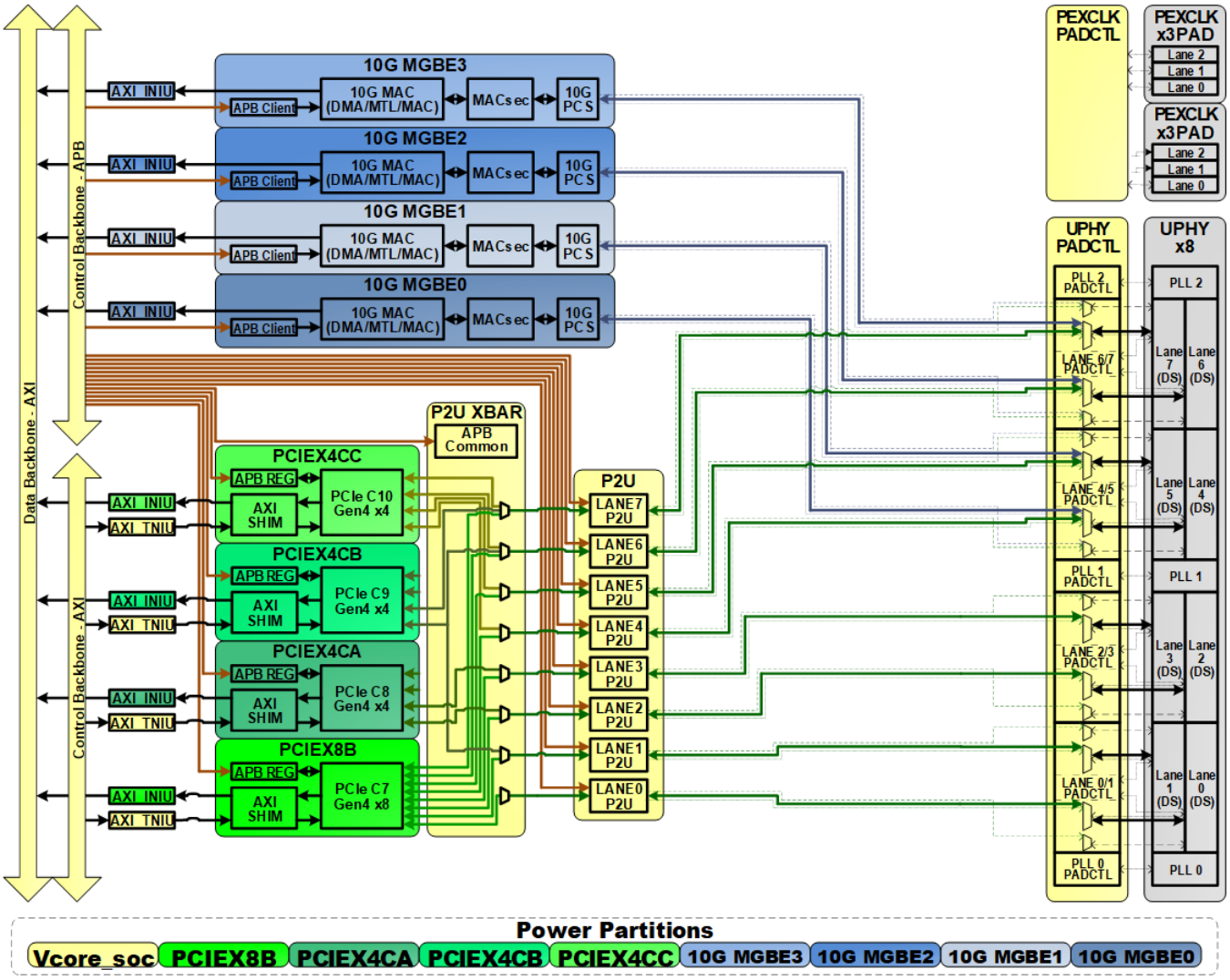
9.1.1.3 GbE UPHY Group

The GbE UPHY group supports the following high-speed I/O controllers that share the 8-lane UPHY brick.

- PCIe
  - One PCIe x8 dual-mode controller
    - Configures root port or endpoint mode
    - Supports Gen1, Gen2, Gen3, and Gen4 link speeds
    - Supports lane reversal - link may be trained to x1, x2, x4, or x8 based on connection to device
  - Three PCIe x4 dual-mode controllers
    - C8 and C9 support root port mode only

        - Software should follow POR use case limitation to setup the controller
      - C10 can be configures root port or endpoint mode
      - Supports Gen1, Gen2, Gen3, and Gen4 link speeds
      - Supports lane reversal - link may be trained to x1, x2, or x4 based on connection to device
    - PCIe consists of four power-gate-able partitions and one unpower-gate-able-partition
      - Each PCIe x4 controllers in its own partition that can be individually power gated when it is unconnected or in L2 link power state
      - One PCIe x8 controller in its own partition that can be individually power gated when it is unconnected or in L2 link power state
      - P2U XBAR and all PIPE2UPHY in one un-power-gate-able partition
- 10-Gigabit Ethernet controller
  - Four IEEE 802.3 10G MAC controllers
    - Support 2.5G, 5G, and 10G speeds
    - Each Ethernet Controller consumes one UPHY lane
  - 10-Gigabit Ethernet consists of four power-gate-able partitions
    - Each 10-Gigabit Ethernet controller in its own partition that can be individually power gated

Figure 9.3 HSIO Block Diagram for Controllers Sharing GbE UPHY



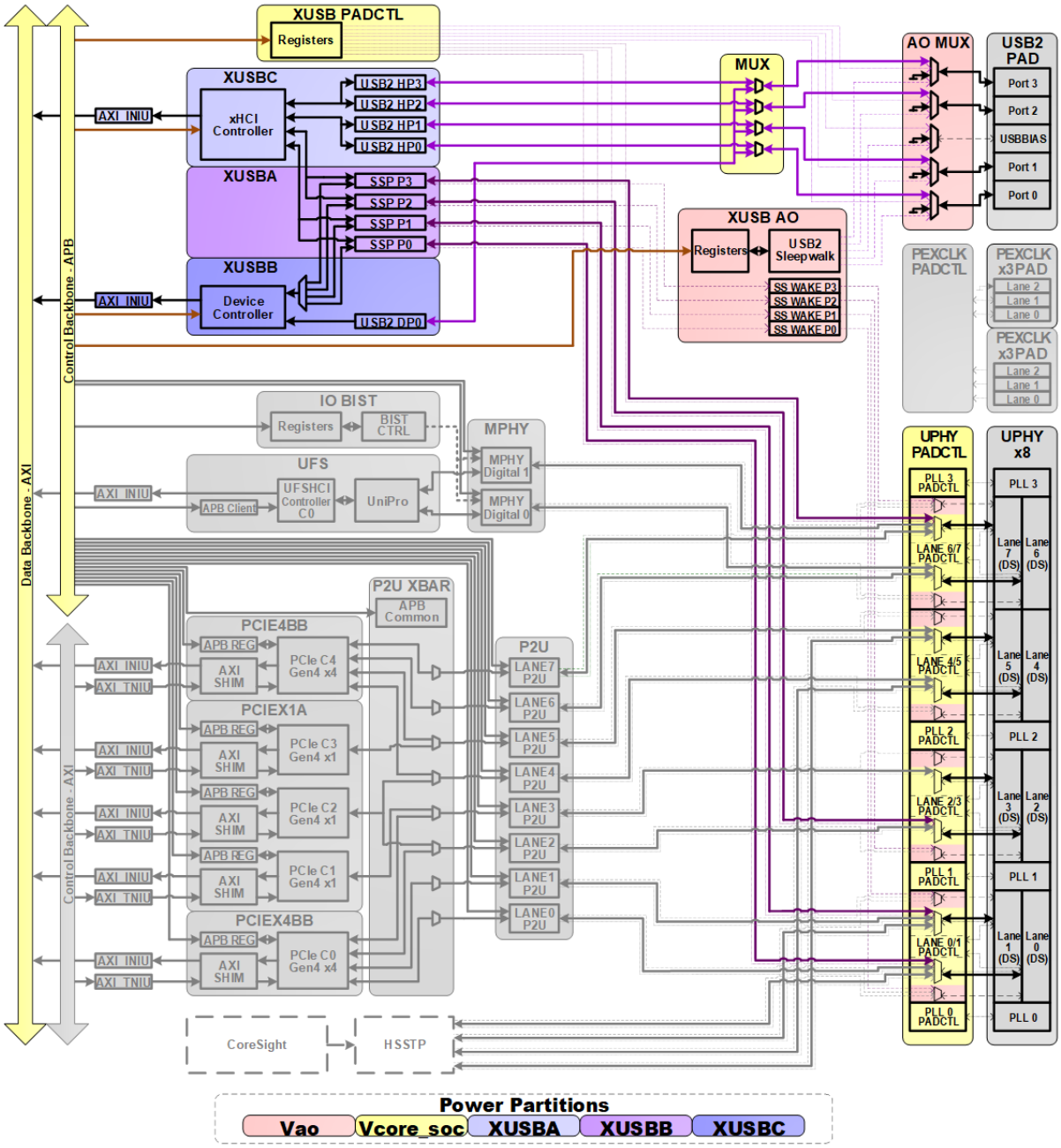
9.2 USB Complex

9.2.1 Overview

The NVIDIA® Orin™ series System-on-Chip (SoC) has one xHCI host controller and one USB 3.2 Gen1 x1 device controller. The two controllers control a total of up to eight exposed ports. There are up to four USB 2.0 ports and up to four USB 3.2 Gen1 x1 ports. Any one the eight ports can be configured as a device port, with the others being host-mode only ports.

The figure below shows the relationship between USB ports and USB controllers.

Figure 9.4 Orin USB Controllers and Interfaces



The USB AO logic is placed under the Always-On power domain; the USB PAD Control logic is placed under the ungated partition of Vcore_soc power domain, where the xHCI controller, the USB 3.2 Gen1 Device controller, UFS, PCIe®, and HSSTP are allocated under power-gate able partitions that are separately power gated.

The USB AO logic is a stand-alone APB slave for setup and access wake detection parameters for USB ports. The USB PAD Control logic is a stand-alone APB slave that accepts downstream requests for programming the PAD MUX and USB2 PAD specific parameters. The USB PAD Control logic also provides the programmability of the capabilities of the individual USB ports.

Below is a summary of the key components in the USB complex:

- xHCI controller supports four USB 2.0 ports and four USB 3.1 Gen2 x1 ports.
- USB 3.2 Gen1 x1 Device controller supports up to one USB 2.0 port or up to one USB 3.1 Gen1 x1 port.

For Type-A and Micro-AB connectors, VBus and ID pin detection are supported via an external PMIC; for Type-C connectors, CC pins detection and signal muxing are supported via an external port policy controller. For USB 3.2 Gen1 x1 Device Controller, the VBUS detection status is notified via software setting the status register bit in USB PAD Control logic.

9.2.2 USB Functional Description

9.2.2.1 USB Controller

Orin has an xHCI controller that supports the xHCI programming model for scheduling transactions and interface managements as a host that natively supports USB 3.2, USB 2.0, and USB 1.1 transactions with its USB 3.2 and USB 2.0 interfaces. The xHCI controller supports up to four USB 3.2 Gen2 x1 ports and their companion USB 2.0 ports, or up to four standalone USB 2.0 ports. The xHCI controller provides mechanisms to communicate with USB 2.0 peripherals, such as keyboards, mice, card readers, and USB 3.2 peripherals, such as cameras and storage devices.

Orin has an USB 3.2 Device controller that supports a device port, allowing Orin to be accessed from an external host device. The USB 3.2 Device controller supports USB 2.0 or USB 3.2 Gen1 x1 with up to 15 IN and 15 OUT endpoints, where a control endpoint consists of one bidirectional endpoint. The endpoints can be configured by the driver to support transfer types of different device classes such as modem, storage, or input devices.

Both the xHCI controller and the USB 3.2 Device controller support USB link power managements. Both controllers support remote wake up, wake on connect, wake on disconnect, and wake on over current in all power states, including deep sleep mode.

> The xHCI Controller and USB 3.2 Device controller are collectively called XUSB internally in NVIDIA to distinguish them from earlier USB 2.0 only controllers. Thus some of register and power partition names have reference to XUSB.

9.2.2.1.1 USB 2 Ports

Each USB 2.0 port operates in USB 2.0 High-Speed mode when connecting directly to a USB 2.0 peripheral or in USB 1.1 full- and low-speed modes when connecting directly to a USB 1.1 peripheral. When operating in High-Speed mode, each USB 2.0 port is allocated with one High-Speed unit bandwidth. Approximately 480 Mb/s bandwidth is allocated to each USB 2.0 port. When operating in full- or low-speed modes all USB 2.0 ports share one full/low-speed unit bandwidth. Approximately a 12 Mb/s bandwidth is distributed across these ports.

All USB 2.0 ports support software initiated L1 and L2 (suspend) link power management. USB 2.0 ports do not support hardware initiated L1 link power management.

9.2.2.1.2 USB 3.1 Ports

USB 3.2 ports support both Generation 1 SuperSpeed USB and Generation 2 SuperSpeedPlus USB 10 Gbps transfer rates. All USB 3.2 ports support x1 only.

Note that due to throughput limitations in the xHCI controller, USB 3.2 port 0 and port 1 are connected to the same SuperSpeedPlus hub to share 10 Gbps total bandwidth, while USB 3.2 port 2 and port 3 share another 10 Gbps total bandwidth. Note also that in the case of two SuperSpeed devices connected to the same SuperSpeedPlus hub, due to scheduling policy reasons each SuperSpeed device may not be able sustain the full SuperSpeed unit bandwidth if the other is device is active.

All USB 3.2 ports support hardware initiated U1 and U2 link power management as well as software initiated U3 (suspend) link power management.

Falcon Microcontroller

The Orin xHCI controller integrates an NVIDIA Falcon microcontroller to perform the following tasks:

- Command ring processing
- Event ring management
- Endpoint scheduling
- Context save and restore

9.2.2.2 Interface Restrictions

A USB 3.2 connector includes both USB 2.0 and USB 3.2 interface signals. The USB 2.0 interface signals of a USB 3.2 connector must also be assigned to the USB controller for xHCI specification compliance. Similarly, if the USB 3.2 interface signals are used to connect to a peripheral on the system board, such as a USB 3.2 hub, the USB 2.0 signals connecting to that peripheral must also be assigned to the USB controller.

Any one of the USB 2.0 ports can be configured as a device port. Any one of the USB 3.2 port can be exclusively paired with the selected USB 2.0 port and operates as the USB 3.2 interface of the device port. Either a USB 2.0 or USB 3.2 interface can be active when the device port is connected to a remote host port.

9.2.2.3 Host and Memory Access Interfaces

The xHCI controller and USB 3.2 device controller use the APB interface as a register access interface for accessing the controller configuration and memory-mapped I/O registers. Both controllers have memory controller interfaces for direct memory accesses.

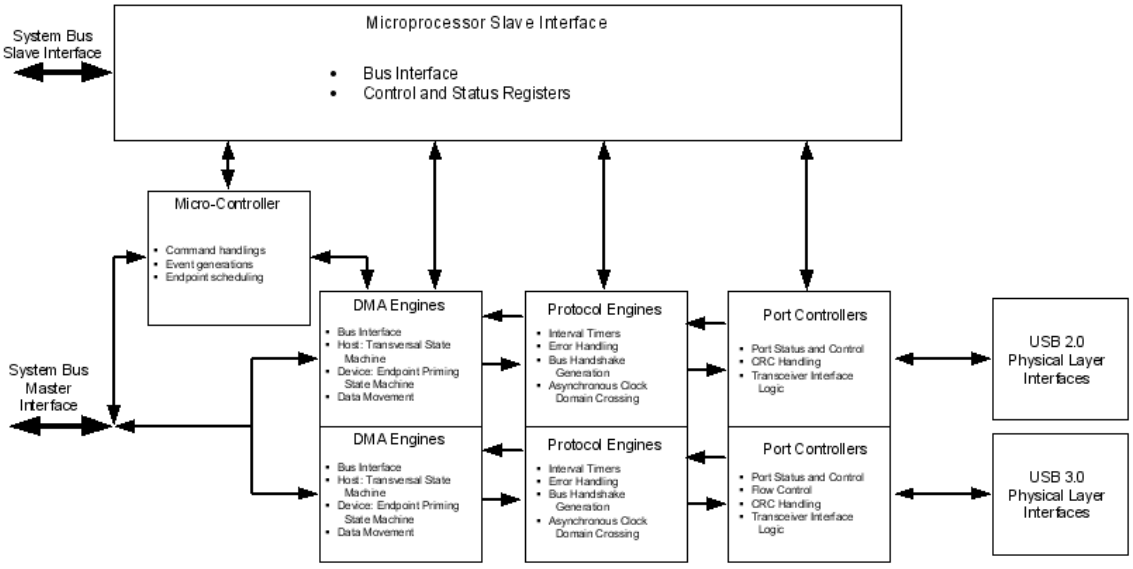
9.2.2.4 USB 3.2 Device Controller

The USB 3.2 device controller allows the mobile platform to be accessed from a host device. Device Mode supports both high/full speed and super speed with up to 15 IN and 15 OUT endpoints, where a control endpoint consists of one bidirectional endpoint. The endpoints can be configured by the driver to support transfer types of different device classes such as modem, storage, or input devices.

9.2.2.5 xHCI Controller Programming Interface

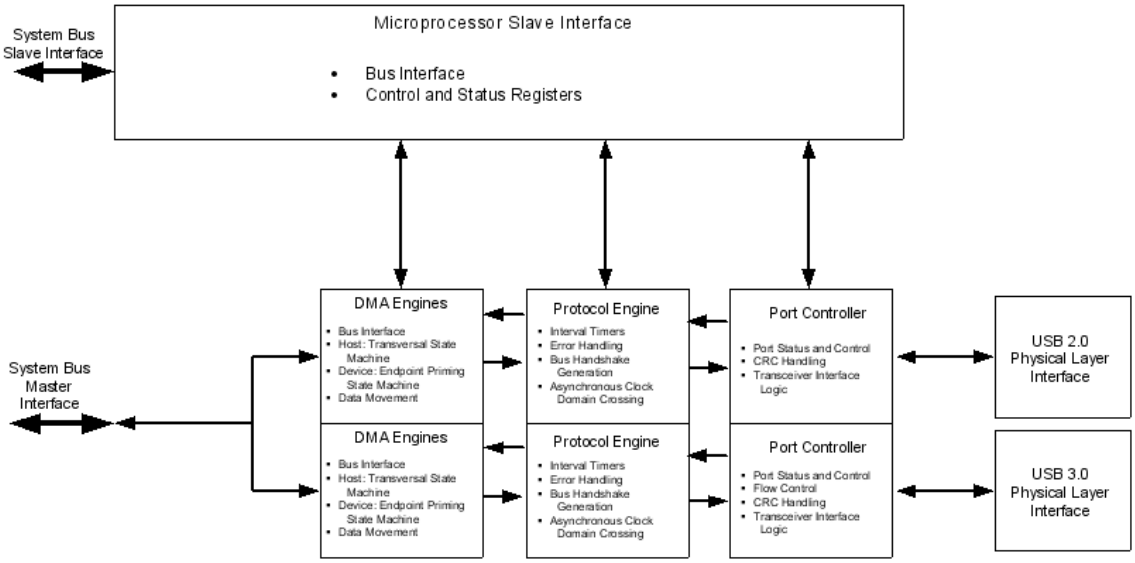
The xHCI controller supports host functionality with host controller registers and data structures implemented as standard xHCI programming interface. The figure below illustrates the control and data paths block diagram of the xHCI controller.

Figure 9.5 xHCI Controller Block Diagram



The USB 3.2 device controller supports device functionality with device controller registers and data structures developed to resemble xHCI programming interface. The figure below illustrates the control and data paths block diagram of the USB 3.2 Device controller.

Figure 9.6 USB 3.2 Device Controller Block Diagram



9.2.2.6 USB PADCTL

The USB PAD Control logic is part of the ungated partition of the Vaux_soc power domain.

The USB PAD Control logic is a stand-alone APB slave that accepts downstream requests for programming the PAD MUX and PAD specific parameters. The USB PAD Control logic also provides the programmability of the capabilities of the individual ports of xHCI controller and USB 3.2 Device controller.

9.2.2.6.1 USB PADCTL Features

- Stand-alone APB slave
  - MMIO registers implemented under a stand-alone APB slave
  - MMIO space outside of xHCI host controller and device mode
- USB Port Control
  - Global configuration for USB ports
- Virtual GPIO
  - VBUS and ID assertion/detection status via dedicated sideband signals from VGPIO
  - Software overrides allow software to directly update status from accessing PMIC
- Battery Charging

- Supports standard and charging downstream port identification control when programmed as host/downstream ports
- Supports charging port detection reporting as when programmed device/upstream ports

9.2.2.7 USB AO

The USB AO logic is part of the Always-On power domain. The USB AO logic is a stand-alone APB slave that accepts downstream requests for setting up and accessing the USB port wake detection and reporting features.

9.2.2.7.1 USB AO Features

- Stand-alone APB slave
  - MMIO registers implemented under a stand-alone APB slave
  - MMIO space outside of xHCI host controller and device mode
- USB 2.0 PAD parameters
  - USB 2.0 pad parameter configurations
  - AO version pad parameters applied under master_enable mode
- USB 2.0 sleepwalk
  - Detect USB 2.0 wake events with programmable conditions
  - React to wake events with four-step sleepwalk sequencer
- USB 3.2 Wake
  - Detect USB 3.2 wake events with port status from each USB 3.1 port

9.2.3 USB Programming Guidelines

9.2.3.1 Cold Boot

Step 1

Boot ROM deasserts reset to USB AO block.

1. Set the following CAR register bit to '0' to deassert reset to USB AO Control block.
   - CLK_RST_CONTROLLER_RST_DEV_XUSB_AON_0[SWR_XUSB_VAUX_AON_RST]

Step 2

1. Set the following CAR register bit to '1' to enable clock to XUSB PAD Control block.
   - CLK_RST_CONTROLLER_CLK_OUT_ENB_XUSB_0[CLK_ENB_XUSB]
2. Set the following CAR register bit to '0' to deassert reset to XUSB PAD Control block.
   - CLK_RST_CONTROLLER_RST_DEV_XUSB_0[SWR_XUSB_PADCTL_RST]

3. Program the following XUSB_PADCTL register to enable device operation on port 0
   - XUSB_PADCTL_USB2_PORT_CAP_0[PORT0_CAP]
4. Initialize the BIAS pad registers
   - XUSB_PADCTL_USB2_BIAS_PAD_CTL_0_0[PD] to 1'b0
   - XUSB_PADCTL_USB2_BIAS_PAD_CTL_0_0[HS_SQUELCH_LEVEL] from fuse USB_CALIB[31:29]
   - Perform Bias pad tracking as described in the USB Programming Guidelines USB2 Pad Tracking Programming section.
5. Initialize the USB2 PAD0 registers
   - XUSB_PADCTL_USB2_OTG_PAD0_CTL_0_0[PD_ZI] to 1'b0
   - XUSB_PADCTL_USB2_OTG_PAD0_CTL_0_0[PD] to 1'b0
   - XUSB_PADCTL_USB2_OTG_PAD0_CTL_0_0[HS_CURR_LEVEL] from fuse USB_CALIB[5:0]
   - TERM_RANGE_ADJ fields in XUSB_PADCTL_USB2_OTG_PAD0_CTL_1_0 register to USB_CALIB[10:7]
   - RPD_CTRL field in XUSB_PADCTL_USB2_OTG_PAD0_CTL_1_0 register to USB_CALIB_EXT[4:0]
   - XUSB_PADCTL_USB2_OTG_PAD0_CTL_1_0[PD_DR] to 1'b0
6. Turn ON protection circuit by writing below XUSB_PADCTL registers
   - XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL1_0[PD_VREG] to 1'b0
   - XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL1_0[VREG_DIR] to 2'b01
   - XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL1_0[VREG_LEV] based on current requirements from VBUS
     - For sinking 500mA, set VREG_LEV[1:0] = [00]
     - For sinking 900mA/1A, set REG_LEV[1:0] = [01]
     - For sinking 1.5A, set VREG_LEV[1:0] = [10]
     - For sinking 2A, set VREG_LEV[1:0] = [11]

Boot ROM performs battery charging operations as described in the USB Programming Guidelines USB2 Pad Tracking Programming section.

Step 3

Boot ROM hands over XUSB to Boot Loader with the following sequence:

- Boot ROM halts all endpoints then notifies Boot Loader.
- Boot Loader updates the Command Ring Pointer, Device Context Base Address Array Pointer, Event Ring Segment Table Base Address, and Event Ring Dequeue Pointer registers to the data structures setup by Boot Loader.
- Boot Loader resets the endpoints then rings the doorbells of the endpoints.

Boot Loader loads XUSB host firmware image to DRAM carveout then notifies XUSB host controller.

- Boot Loader sets up a carveout in DRAM and loads XUSB firmware image to the carveout from boot storage.
- Boot Loader writes the based address to the following registers in XUSB AO:
  - XUSB_AO_IFRDMA_CFG1_0[SWR_BASE_ADDR_39_32]
  - XUSB_AO_IFRDMA_CFG0_0[SWR_BASE_ADDR_31_0]

Note: These are write-once registers that cannot be modified until the system goes through cold boot. Writing 0 to these registers disables XUSB firmware loading.

Step 4

PAD driver switches USB 2.0 related PLLs to under hardware control after PAD driver is loaded. PAD driver enables USB3.0 related PLLs.

Note: Boot Loader may decide to keep the PLLs enabled with the software override in case there are bugs in hardware control preventing PLLs from being enabled correctly.

Step 5

PAD driver enables platform specific regulators to enable power rails to the pads, VBUS, and pull-up voltage to the VBUS control PMIC's EN input.

PAD driver deasserts reset to UPHY PAD and UPHY PAD Macro.

1. Set the following CAR register bit to '0' to deassert reset to UPHY Pad and UPHY Pad Macros.
   - CLK_RST_CONTROLLER_RST_DEV_UPHY_0[SWR_UPHY_RST]
   - CLK_RST_CONTROLLER_RST_DEV_PEX_USB_UPHY_0[SWR_PEX_USB_UPHY_RST]

PAD driver setup the I/O pins for the VBUS control and over current report if VBUS control is used in the platform.

2. Program the following PADCTL EDP registers to setup the I/O pins for USB 2.0 ports, according to the platform specific configuration:
   - PADCTL_EDP_USB_VBUS_EN0_0[E_IO_HV]
   - PADCTL_EDP_USB_VBUS_EN0_0[E_INPUT]
   - PADCTL_EDP_USB_VBUS_EN0_0[PARK]
   - PADCTL_EDP_USB_VBUS_EN0_0[TRISTATE]
   - PADCTL_EDP_USB_VBUS_EN0_0[PUPD]
   - PADCTL_EDP_USB_VBUS_EN1_0[E_IO_HV]
   - PADCTL_EDP_USB_VBUS_EN1_0[E_INPUT]
   - PADCTL_EDP_USB_VBUS_EN1_0[PARK]
   - PADCTL_EDP_USB_VBUS_EN1_0[TRISTATE]

- PADCTL_EDP_USB_VBUS_EN1_0[PUPD]

3. Program the following XUSB PADCTL registers to use local override for VBUS and ID status reporting:

- XUSB_PADCTL_USB2_VBUS_ID_0[ID_SOURCE_SELECT] to 'ID_OVERRIDE'
- XUSB_PADCTL_USB2_VBUS_ID_0[VBUS_SOURCE_SELECT] to 'VBUS_OVERRIDE'

4. Write '1' to the following XUSB PADCTL registers to clear false reporting of VBUS and ID status changes:

- XUSB_PADCTL_USB2_VBUS_ID_0[IDDIG_ST_CHNG]
- XUSB_PADCTL_USB2_VBUS_ID_0[VBUS_VALID_ST_CHNG]

PAD driver programs the port capabilities and pad parameters of ports assigned to XUSB after booted to OS. PAD driver sets the capabilities of the ports to OTG for ports that are used to form the Type-C port.

5. Program the following XUSB PADCTL registers to assign the port capabilities for USB2.0 ports according to the platform specific configuration:

- XUSB_PADCTL_USB2_PORT_CAP_0[PORT2_CAP] for host/disabled
- XUSB_PADCTL_USB2_PORT_CAP_0[PORT2_INTERNAL] for whether its internal port
- XUSB_PADCTL_USB2_PORT_CAP_0[PORT1_CAP] for host/disabled
- XUSB_PADCTL_USB2_PORT_CAP_0[PORT1_INTERNAL] for whether its internal port
- XUSB_PADCTL_USB2_PORT_CAP_0[PORT0_CAP] for host/device/OTG/disabled
- XUSB_PADCTL_USB2_PORT_CAP_0[PORT0_INTERNAL] for whether its internal port

6. Program the following XUSB PADCTL registers to assign the port capabilities for SuperSpeed ports according to the platform specific configuration:

- XUSB_PADCTL_SS_PORT_CAP_0[PORT2_CAP] for host/device/OTG/disabled
- XUSB_PADCTL_SS_PORT_CAP_0[PORT2_INTERNAL] for whether its internal port
- XUSB_PADCTL_SS_PORT_CAP_0[PORT1_CAP] for host/device/OTG/disabled
- XUSB_PADCTL_SS_PORT_CAP_0[PORT1_INTERNAL] for whether its internal port
- XUSB_PADCTL_SS_PORT_CAP_0[PORT1_TYPEC_CAP_EN] to 1 if the port is used to form the Type-C port
- XUSB_PADCTL_SS_PORT_CAP_0[PORT0_CAP] for host/device/OTG/disabled
- XUSB_PADCTL_SS_PORT_CAP_0[PORT0_INTERNAL] for whether its internal port
- XUSB_PADCTL_SS_PORT_CAP_0[PORT0_TYPEC_CAP_EN] to 1 if the port is used to form the Type-C port
- To have USB2 only OTG, or USB2 only device mode port, have one USB2_PORT_CAP set to device/OTG and NONE of the SS_PORT_CAP set to device/OTG.

7. Program the following XUSB PADCTL registers to 'OC_DETECTION_DISABLED' to disable the over current signal mapping for USB 2.0 and SS ports:

- XUSB_PADCTL_USB2_OC_MAP_0[PORT2_OC_PIN]
- XUSB_PADCTL_USB2_OC_MAP_0[PORT1_OC_PIN]
- XUSB_PADCTL_USB2_OC_MAP_0[PORT0_OC_PIN]
- XUSB_PADCTL_SS_OC_MAP_0[PORT2_OC_PIN]
- XUSB_PADCTL_SS_OC_MAP_0[PORT1_OC_PIN]
- XUSB_PADCTL_SS_OC_MAP_0[PORT0_OC_PIN]
- XUSB_PADCTL_VBUS_OC_MAP_0[VBUS_ENABLE1_OC_MAP]
- XUSB_PADCTL_VBUS_OC_MAP_0[VBUS_ENABLE0_OC_MAP]

Program the UPHY Lane registers to assign the UPHY lanes, set static UPHY LANE and PLL parameters according to the platform specific configuration, and assign the static UPHY LANE parameters of ports owned by XUSB according to the platform specific configuration.

8. Program the following XUSB PADCTL registers to assign the static USB2.0 PAD parameters of ports owned by XUSB, according to the platform specific configuration:

- XUSB_PADCTL_USB2_OTG_PAD0_CTL_0_0
- XUSB_PADCTL_USB2_OTG_PAD0_CTL_1_0
- XUSB_PADCTL_USB2_OTG_PAD1_CTL_0_0
- XUSB_PADCTL_USB2_OTG_PAD1_CTL_1_0
- XUSB_PADCTL_USB2_OTG_PAD2_CTL_0_0
- XUSB_PADCTL_USB2_OTG_PAD2_CTL_1_0
- XUSB_PADCTL_USB2_BIAS_PAD_CTL_0_0
- XUSB_PADCTL_USB2_BIAS_PAD_CTL_1_0

9. Program the following XUSB PADCTL register bits to ‘0’ to disable power down of the USB2.0 of ports owned by XUSB, according to the platform specific configuration.

- XUSB_PADCTL_USB2_OTG_PAD0_CTL_0_0[PD]
- XUSB_PADCTL_USB2_OTG_PAD1_CTL_0_0[PD]
- XUSB_PADCTL_USB2_OTG_PAD2_CTL_0_0[PD]
- XUSB_PADCTL_USB2_OTG_PAD0_CTL_0_0[PD_ZI]
- XUSB_PADCTL_USB2_OTG_PAD1_CTL_0_0[PD_ZI]
- XUSB_PADCTL_USB2_OTG_PAD2_CTL_0_0[PD_ZI]
- XUSB_PADCTL_USB2_OTG_PAD0_CTL_1_0[PD_DR]
- XUSB_PADCTL_USB2_OTG_PAD1_CTL_1_0[PD_DR]
- XUSB_PADCTL_USB2_OTG_PAD2_CTL_1_0[PD_DR]
- XUSB_PADCTL_USB2_BIAS_PAD_CTL_0_0[PD]

10. Program the following XUSB PADCTL register bits to ‘0’ to release the XUSB SS wake logic state latching.

- XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP2_ELPG_CLAMP_EN]

- XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP2_ELPG_CLAMP_EN_EARLY]
- XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP2_ELPG_VCORE_DOWN]
- XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP1_ELPG_CLAMP_EN]
- XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP1_ELPG_CLAMP_EN_EARLY]
- XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP1_ELPG_VCORE_DOWN]
- XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP0_ELPG_CLAMP_EN]
- XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP0_ELPG_CLAMP_EN_EARLY]
- XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP0_ELPG_VCORE_DOWN]

Note: To avoid confliction or ambiguity, the XUSB controller should be under reset when updating the port assignments.

Step 6

PAD driver programs the clocks and deasserts the resets to the controllers.

1. Set the following CAR register bits to '1' to enable the clocks to XUSB:
   - CLK_RST_CONTROLLER_CLK_OUT_ENB_XUSB_SET_0[SET_CLK_ENB_XUSB]

2. Set the following CAR register bits to '1' to enable the clocks to individual XUSB partitions:
   - CLK_RST_CONTROLLER_CLK_OUT_ENB_XUSB_SET_0[SET_CLK_ENB_XUSB_HOST]
   - CLK_RST_CONTROLLER_CLK_OUT_ENB_XUSB_SET_0[SET_CLK_ENB_XUSB_DEV]
   - CLK_RST_CONTROLLER_CLK_OUT_ENB_XUSB_SET_0[SET_CLK_ENB_XUSB_SS]

3. Program the following CAR register bits to set the source of XUSB clocks, where PLLP_OUT0 runs at 408 MHz:
   - CLK_RST_CONTROLLER_CLK_SOURCE_XUSB_CORE_HOST_0[XUSB_CORE_HOST_CLK_SRC] to PLLP_OUT0
   - CLK_RST_CONTROLLER_CLK_SOURCE_XUSB_CORE_HOST_0[XUSB_CORE_HOST_CLK_DIVISOR] to 0x6
   - CLK_RST_CONTROLLER_CLK_SOURCE_XUSB_CORE_DEV_0[XUSB_CORE_DEV_CLK_SRC] to PLLP_OUT0
   - CLK_RST_CONTROLLER_CLK_SOURCE_XUSB_CORE_DEV_0[XUSB_CORE_DEV_CLK_DIVISOR] to 0x6
   - CLK_RST_CONTROLLER_CLK_SOURCE_XUSB_FALCON_0[XUSB_FALCON_CLK_SRC] to PLLP_OUT0
   - CLK_RST_CONTROLLER_CLK_SOURCE_XUSB_FALCON_0[XUSB_FALCON_CLK_DIVISOR] to 0x0
   - CLK_RST_CONTROLLER_CLK_SOURCE_XUSB_FS_0[XUSB_FS_CLK_SRC] to FO_48M
   - CLK_RST_CONTROLLER_CLK_SOURCE_XUSB_FS_0[XUSB_FS_CLK_DIVISOR] to 0x0
   - CLK_RST_CONTROLLER_CLK_SOURCE_XUSB_SS_0[XUSB_SS_CLK_SRC] to HSIC_480

- CLK_RST_CONTROLLER_CLK_SOURCE_XUSB_SS_0[XUSB_SS_CLK_DIVISOR] to 0x6

4. Set the following CAR register bits to '0' to deassert reset to XUSB:

- CLK_RST_CONTROLLER_RST_DEV_XUSB_0[SWR_XUSB_HOST_RST]
- CLK_RST_CONTROLLER_RST_DEV_XUSB_0[SWR_XUSB_DEV_RST]
- CLK_RST_CONTROLLER_RST_DEV_XUSB_0[SWR_XUSB_SS_RST]

Step 7

PAD driver brings the UPHY out of IDDQ. PAD driver enables the VBUS to the USB ports.

1. Program the following XUSB PADCTL registers to assign the over current signal mapping for USB 2.0 and SS ports according to the platform specific configuration:

- XUSB_PADCTL_USB2_OC_MAP_0[PORT2_OC_PIN]
- XUSB_PADCTL_USB2_OC_MAP_0[PORT1_OC_PIN]
- XUSB_PADCTL_USB2_OC_MAP_0[PORT0_OC_PIN]
- XUSB_PADCTL_SS_OC_MAP_0[PORT2_OC_PIN]
- XUSB_PADCTL_SS_OC_MAP_0[PORT1_OC_PIN]
- XUSB_PADCTL_SS_OC_MAP_0[PORT0_OC_PIN]
- XUSB_PADCTL_VBUS_OC_MAP_0[VBUS_ENABLE1_OC_MAP]
- XUSB_PADCTL_VBUS_OC_MAP_0[VBUS_ENABLE0_OC_MAP]

2. Write '1' to the following XUSB PADCTL register bits to clear possible false reporting of over current events before the over current signal mappings are properly programmed:

- XUSB_PADCTL_OC_DET_0[OC_DETECTED2]
- XUSB_PADCTL_OC_DET_0[OC_DETECTED1]
- XUSB_PADCTL_OC_DET_0[OC_DETECTED0]
- XUSB_PADCTL_OC_DET_0[OC_DETECTED_VBUS_PAD2]
- XUSB_PADCTL_OC_DET_0[OC_DETECTED_VBUS_PAD1]
- XUSB_PADCTL_OC_DET_0[OC_DETECTED_VBUS_PAD0]

3. Wait 1 us

4. Set the following XUSB PADCTL register bits to '1' to enable the VBUS of the host ports:

- XUSB_PADCTL_VBUS_OC_MAP_0[VBUS_ENABLE1]
- XUSB_PADCTL_VBUS_OC_MAP_0[VBUS_ENABLE0]

Step 8

xHCI platform driver sets up StreamID of XUSB Host's physical function, enabled virtual functions, and firmware loading state machine.

1. Program the following XUSB AO registers to set up StreamID:
   - XUSB_AO_IFR_AXI_STREAMID_0[IFR_STREAMID]

2. Program the following XUSB PADCTL registers to set up StreamID:
   - XUSB_PADCTL_HOST_AXI_STREAMID_PF_0[PF_STREAMID]
   - XUSB_PADCTL_HOST_AXI_STREAMID_VF_0[VF_STREAMID]
   - XUSB_PADCTL_HOST_AXI_STREAMID_VF_1[VF_STREAMID]
   - XUSB_PADCTL_HOST_AXI_STREAMID_VF_2[VF_STREAMID]
   - XUSB_PADCTL_HOST_AXI_STREAMID_VF_3[VF_STREAMID]

xHCI platform driver sets up StreamID of XUSB Device's physical function.

3. Program the following XUSB PADCTL registers to set up StreamID:
   - XUSB_PADCTL_DEV_AXI_STREAMID_PF_0[PF_STREAMID]

xHCI platform driver initializes XUSB Host's configuration registers.

4. Program the following XUSB Host configuration registers to initialize XUSB:
   - NV_PROJ__XUSB_CFG_1_BUS_MASTER to '1'
   - NV_PROJ__XUSB_CFG_1_MEMORY_SPACE to '1'
   - NV_PROJ__XUSB_CFG_4_BASE_ADDRESS to '0x500000'

xHCI Device driver initializes XUSB Device's configuration registers.

5. Program the following XUSB Device configuration registers to initialize XUSB:
   - NV_PROJ__XUSB_DEV_CFG_1_BUS_MASTER to '1'
   - NV_PROJ__XUSB_DEV_CFG_1_MEMORY_SPACE to '1'
   - NV_PROJ__XUSB_DEV_CFG_4_BASE_ADDRESS to '0x450000'

xHCI Device driver enables XUSB Device controller.

6. Program the following XUSB Device MMIO registers to enable XUSB Device:
   - NV_PROJ__XUSB_DEV_XHCI_CTRL_0_ENABLE to '1'

9.2.3.2 SC7 (LP0)

9.2.3.2.1 SC7 Entry

Step 1
The xHCI driver performs a context save operation as described in the xHCI specification. The xHCI

PEP driver performs an XUSB specific context save operation. The xHCI PEP driver performs an XUSB IPFS specific context save operation.

1. Read and store the value of the following registers:
   - XUSB_{HOST,DEV}_MSI_BAR_SZ_0
   - XUSB_{HOST,DEV}_MSI_AXI_BAR_ST_0
   - XUSB_{HOST,DEV}_MSI_FPCI_BAR_ST_0
   - XUSB_{HOST,DEV}_MSI_VEC0_0
   - XUSB_{HOST,DEV}_MSI_EN_VEC0_0
   - XUSB_{HOST,DEV}_FPCI_ERROR_MASKS_0
   - XUSB_{HOST,DEV}_INTR_MASK_0
   - XUSB_{HOST,DEV}_IPFS_INTR_ENABLE_0
   - XUSB_{HOST,DEV}_UFPCI_CONFIG_0
   - XUSB_{HOST,DEV}_CLKGATE_HYSTERESIS_0
   - XUSB_{HOST,DEV}_XUSB_HOST_MCCIF_FIFOCTRL_0

Step 2
The System Power Management driver programs the PMC USB2.0 sleepwalk logic as described in sections. The System Power Management driver should enable the wake events according to the port states and Wake-on-Connect, Wake-on-Disconnect, and Wake-on-Over-Current settings of XUSB registers ports accordingly.

The System Power Management driver enables wake events from USB ports.

1. Set the following PMC WAKE register bits to '1' to set the wake signal active level to 'HIGH'
   - WAKE_AOWAKE_CNTRL_76[3] for SS port 0 wakeup
   - WAKE_AOWAKE_CNTRL_77[3] for SS port 1 wakeup
   - WAKE_AOWAKE_CNTRL_78[3] for SS port 2 and port 3 wakeup
   - WAKE_AOWAKE_CNTRL_79[3] for USB2 NVWRAP port 0 wakeup
   - WAKE_AOWAKE_CNTRL_80[3] for USB2 NVWRAP port 1 wakeup
   - WAKE_AOWAKE_CNTRL_81[3] for USB2 NVWRAP port 2 wakeup
   - WAKE_AOWAKE_CNTRL_82[3] for USB2 NVWRAP port 3 wakeup
2. Set the following PMC WAKE register bits to '1' to enable USB wake events
   - WAKE_AOWAKE_MASK_W_76[0] for SS port 0 wakeup
   - WAKE_AOWAKE_MASK_W_77[0] for SS port 1 wakeup
   - WAKE_AOWAKE_MASK_W_78[0] for SS port 2 and port 3 wakeup
   - WAKE_AOWAKE_MASK_W_79[0] for USB2 NVWRAP port 0 wakeup
   - WAKE_AOWAKE_MASK_W_80[0] for USB2 NVWRAP port 1 wakeup
   - WAKE_AOWAKE_MASK_W_81[0] for USB2 NVWRAP port 2 wakeup
   - WAKE_AOWAKE_MASK_W_82[0] for USB2 NVWRAP port 3 wakeup

Step 3
The System Power Management driver sets up the VBUS_ENABLE I/O pins as GPIO with wake events enabled to enable wake on overcurrent.

1. Set the following USB PADCTL registers to 'OC_DETECTION_DISABLED' to disable overcurrent wake events reporting from USB PADCTL:
   - XUSB_PADCTL_VBUS_OC_MAP_0[VBUS_ENABLE3_OC_MAP]
   - XUSB_PADCTL_VBUS_OC_MAP_0[VBUS_ENABLE2_OC_MAP]
   - XUSB_PADCTL_VBUS_OC_MAP_0[VBUS_ENABLE1_OC_MAP]
   - XUSB_PADCTL_VBUS_OC_MAP_0[VBUS_ENABLE0_OC_MAP]
2. Set the following PMC WAKE register bits to '1' to set the wake signal active level to 'HIGH' and enable the wake events:
   - WAKE_AOWAKE_CNTRL_62[3] for VBUS_ENABLE1
   - WAKE_AOWAKE_CNTRL_61[3] for VBUS_ENABLE0
   - WAKE_AOWAKE_MASK_W_62[0] for VBUS_ENABLE1
   - WAKE_AOWAKE_MASK_W_61[0] for VBUS_ENABLE0

Step 4
The system software puts the system in SC7.

9.2.3.2.2 SC7 Exit

After exiting SC7, the PMC restores the partition power gating to the state before SC7 entry. This means if a partition was power gated before entering SC7, that partition would stay power gated after exiting SC7, and only partitions that were powered before entering SC7 would have their power restored after exiting SC7.

Step 1
The Boot ROM deasserts reset to USB PADCTL block.

Step 2
The Boot ROM performs battery charging operations through USB2 Controller.

Step 3
The SC7 exit sequence is described in the Power Management Controller chapter of this TRM.

Step 4
The PAD driver checks the VBUS_ENABLE I/O pins for overcurrent events and disables wake event reporting.

1. Read the following PMC WAKE register bits
   - WAKE_AOWAKE_MASK_R_63_32_0[30] for VBUS_ENABLE1
   - WAKE_AOWAKE_MASK_R_63_32_0[29] for VBUS_ENABLE0
2. The PAD driver forwards the overcurrent events to XUSB if they occurred during SC7.

3. Program the following USB PADCTL registers to enable software override of overcurrent wake events
   - XUSB_PADCTL_OC_DET_0[SET_OC_DETECTED0]
4. Program the following USB PADCTL registers to 'OC_DETECTED0' to report the overcurrent wake events to XUSB or USB 2 ports, according to the platform specific configuration:
   - XUSB_PADCTL_USB2_OC_MAP_0[PORT3_OC_PIN]
   - XUSB_PADCTL_USB2_OC_MAP_0[PORT2_OC_PIN]
   - XUSB_PADCTL_USB2_OC_MAP_0[PORT1_OC_PIN]
   - XUSB_PADCTL_USB2_OC_MAP_0[PORT0_OC_PIN]
   - XUSB_PADCTL_SS_OC_MAP_0[PORT3_OC_PIN]
   - XUSB_PADCTL_SS_OC_MAP_0[PORT2_OC_PIN]
   - XUSB_PADCTL_SS_OC_MAP_0[PORT1_OC_PIN]
   - XUSB_PADCTL_SS_OC_MAP_0[PORT0_OC_PIN]
   - XUSB_PADCTL_VBUS_OC_MAP_0[VBUS_ENABLE1_OC_MAP]
   - XUSB_PADCTL_VBUS_OC_MAP_0[VBUS_ENABLE0_OC_MAP]

Step 5
The PAD driver assigns the USB port to the controllers, then programs the port capabilities and pad parameters of ports assigned to XUSB according to the platform specific configuration.

Refer to Step 3, #4 through #7 in for the programming sequence.

Step 6
The PAD driver programs the clocks and deasserts the resets to the controllers.

1. Refer to Step 4 the Cold Boot Section for the start of the programming sequence.
2. Wait 1 µs.
3. Program the following USB PADCTL register bits to '0' to release the XUSB SS wake logic state latching:
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP3_ELPG_VCORE_DOWN]
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP2_ELPG_VCORE_DOWN]
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP1_ELPG_VCORE_DOWN]
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP0_ELPG_VCORE_DOWN]
4. Wait 100 µs.
5. Program the following USB PADCTL register bits to '0' to release the XUSB SS wake logic state latching:
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP3_ELPG_CLAMP_EN_EARLY]
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP2_ELPG_CLAMP_EN_EARLY]
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP1_ELPG_CLAMP_EN_EARLY]
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP0_ELPG_CLAMP_EN_EARLY]
6. Wait 100 µs.

7. Program the following USB PADCTL register bits to '0' to release the XUSB SS wake logic state latching:
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP3_ELPG_CLAMP_EN]
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP2_ELPG_CLAMP_EN]
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP1_ELPG_CLAMP_EN]
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP0_ELPG_CLAMP_EN]

Step 7
The PAD driver programs the capability and pad parameters of ports owned by USB2 controllers.

Refer to Step 3, #8 of the Cold Boot Section for the USB 2 port capability and PAD/PLL parameter programming sequence.

Step 8
The PAD driver enables the VBUS to the USB ports if VBUS was disabled during SC7.

Refer to Step 5 of the Cold Boot Section for the programming sequence.

Step 9
The xHCI PEP driver performs the XUSB specific context restore operation, if XUSB was not in ELPG before entering SC7 or the SC7 exit is due to wake event from ports of XUSB.

The xHCI PEP driver performs XUSB register initialization as described below:

xHCI PEP driver initializes XUSB Host's configuration registers

1. Program the following XUSB configuration registers to initialize XUSB:
   - T_XUSB_T_ XUSB_CFG_1_BUS_MASTER to '1'
   - T_XUSB_T_ XUSB_CFG_1_MEMORY_SPACE to '1'
   - T_XUSB_T_ XUSB_CFG_4_BASE_ADDRESS to 0x02000

xHCI Device driver initializes XUSB Host's configuration registers

1. Program the following XUSB configuration registers to initialize XUSB:
   - T_XUSB_T_ XUSB_DEV_CFG_1_BUS_MASTER to '1'
   - T_XUSB_T_ XUSB_DEV_CFG_1_MEMORY_SPACE to '1'
   - T_XUSB_T_ XUSB_DEV_CFG_4_BASE_ADDRESS to 0x02000

Step 10
The xHCI PEP Driver loads XUSB firmware. If XUSB was not in ELPG before entering SC7 and the SC7 exit is not due to wake event from ports of XUSB.

Step 11
The System Power Management driver programs the PMC USB 2.0 sleepwalk logic.

9.2.3.3 XUSB Controller Power Gating

The XUSB controller has three partitions that can be selectively power gated with the following use cases, where unlisted combinations are not supported.

Table 9.1 Selective Power Gating for Three Partitions

XUSBC (Host/USB 2.0)	XUSBB (Device)	XUSBA (SuperSpeed)	Use Case Descriptions
Powered	Powered	Powered	Both Host and Device modes in normal operations, with SuperSpeed links operational.
Powered	Power Gated	Powered	Host mode in normal operations, with SuperSpeed links operational. Device mode power gated. Device mode SuperSpeed and USB 2.0 link wake events report through USB PADCTL.
Power Gated	Power Gated	Power Gated	Both Host and Device modes power gated, with SuperSpeed links in low power states. All Host and Device modes link wake events report through USB PADCTL.
Power Gated	Powered	Powered	Host mode in ELPG while Device mode is not in ELPG. SuperSpeed links in normal operation.

Due to the wake latency requirements, device partition can only be power-gated when the device port is not connected.

9.2.3.3.1 All Partitions ELPG Entry

Step 1

The xHCI driver performs context save operation as described in the xHCI specification.

The xHCI PEP driver performs XUSB_HOST specific context save operation. In case the SuperSpeed partition has already been power gated, xHCI firmware (FW) and XUSB Device Mode drivers should not save the context of the SuperSpeed partition again.

The XUSB Device Mode driver performs XUSB_DEVICE specific context save operations.

Step 2

The System Power Management driver programs the PMC USB 2.0 sleepwalk logic for ports assigned to XUSB host mode and XUSB Device Mode according as described in Initialize PMC Sleepwalk Logic and Enable PMC Sleepwalk Logic sections.

The System Power Management driver should enable the wake events according to the port states and Wake-on-Connect, Wake-on-Disconnect, and Wake-on-Over-Current settings of the USB 2 ports accordingly.

Step 3

The xHCI PEP driver and XUSB Device Mode driver enable the XUSB wakeup interrupts for the SuperSpeed and USB 2.0 ports assigned to host and device.

1. Write '1' to the following USB PADCTL register bits to clear the interrupt status.
   - XUSB_PADCTL_ELPG_PROGRAM_0_0[SS_PORT3_WAKEUP_EVENT]
   - XUSB_PADCTL_ELPG_PROGRAM_0_0[SS_PORT2_WAKEUP_EVENT]
   - XUSB_PADCTL_ELPG_PROGRAM_0_0[SS_PORT1_WAKEUP_EVENT]
   - XUSB_PADCTL_ELPG_PROGRAM_0_0[SS_PORT0_WAKEUP_EVENT]
   - XUSB_PADCTL_ELPG_PROGRAM_0_0[USB2_PORT3_WAKEUP_EVENT]
   - XUSB_PADCTL_ELPG_PROGRAM_0_0[USB2_PORT2_WAKEUP_EVENT]
   - XUSB_PADCTL_ELPG_PROGRAM_0_0[USB2_PORT1_WAKEUP_EVENT]
   - XUSB_PADCTL_ELPG_PROGRAM_0_0[USB2_PORT0_WAKEUP_EVENT]
   - Set the following USB PADCTL register bits to '1' to enable the interrupt.
   - XUSB_PADCTL_ELPG_PROGRAM_0_0[SS_PORT3_WAKE_INTERRUPT_ENABLE]
   - XUSB_PADCTL_ELPG_PROGRAM_0_0[SS_PORT2_WAKE_INTERRUPT_ENABLE]
   - XUSB_PADCTL_ELPG_PROGRAM_0_0[SS_PORT1_WAKE_INTERRUPT_ENABLE]
   - XUSB_PADCTL_ELPG_PROGRAM_0_0[SS_PORT0_WAKE_INTERRUPT_ENABLE]
   - XUSB_PADCTL_ELPG_PROGRAM_0_0[USB2_PORT3_WAKE_INTERRUPT_ENABLE]
   - XUSB_PADCTL_ELPG_PROGRAM_0_0[USB2_PORT2_WAKE_INTERRUPT_ENABLE]
   - XUSB_PADCTL_ELPG_PROGRAM_0_0[USB2_PORT1_WAKE_INTERRUPT_ENABLE]
   - XUSB_PADCTL_ELPG_PROGRAM_0_0[USB2_PORT0_WAKE_INTERRUPT_ENABLE]

Step 4

The xHCI PEP driver and XUSB Device Mode driver initiate the signal sequence to enable the XUSB SSwake detection logic for the SuperSpeed ports assigned to host and device.

1. Write '1' to the following USB PADCTL register bits to assert the clamp_en_early signal.
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP3_ELPG_CLAMP_EN_EARLY]
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP2_ELPG_CLAMP_EN_EARLY]
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP1_ELPG_CLAMP_EN_EARLY]
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP0_ELPG_CLAMP_EN_EARLY]
2. Write '1' to the following USB PADCTL register bits to assert the clamp_en signal.
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP3_ELPG_CLAMP_EN]
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP2_ELPG_CLAMP_EN]

- XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP1_ELPG_CLAMP_EN]
- XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP0_ELPG_CLAMP_EN]

3. Wait 250 µs

Step 5

System Power Management driver flushes MCCIF and partition clients.

1. Set the following MC register bits to '1' to enable flush to XUSB:
   - MC_CLIENT_HOTRESET_CTRL_0[XUSB_HOST_FLUSH_ENABLE] for host mode
   - MC_CLIENT_HOTRESET_CTRL_0[XUSB_DEV_FLUSH_ENABLE] for device mode
2. Read the following MC register bits to be '1' to ensure flush to XUSB is enabled:
   - MC_CLIENT_HOTRESET_STATUS_0[XUSB_HOST_HOTRESET_STATUS] for host mode
   - MC_CLIENT_HOTRESET_STATUS_0[XUSB_DEV_HOTRESET_STATUS] for device mode

Step 6

The System Power Management driver asserts reset to XUSB then disables its clocks.

1. Set the following CAR register bits to '1' to assert reset to XUSB:
   - CLK_RST_CONTROLLER_RST_DEV_XUSB_0[SWR_XUSB_HOST_RST] for host mode
   - CLK_RST_CONTROLLER_RST_DEV_XUSB_0[SWR_XUSB_DEV_RST] for device mode
   - CLK_RST_CONTROLLER_RST_DEV_XUSB_0[SWR_XUSB_SS_RST] for SuperSpeed ports
2. Set the following CAR register bits to '1' to disable the clocks to individual XUSB partitions:
   - CLK_RST_CONTROLLER_CLK_OUT_ENB_XUSB_CLR_0[CLR_CLK_ENB_XUSB_HOST] for host mode
   - CLK_RST_CONTROLLER_CLK_OUT_ENB_XUSB_CLR_0[CLR_CLK_ENB_XUSB_DEV] for device mode
   - CLK_RST_CONTROLLER_CLK_OUT_ENB_XUSB_CLR_0[CLR_CLK_ENB_XUSB_SS] for SS ports

Step 7

The System Power Management driver disables the XUSB power rails.

1. Program the following PMC register bits in a single write to disable the power rail to XUSB host:
   - PMC_IMPL_PART_XUSBC_POWER_GATE_CONTROL_0[INTER_PART_DELAY_EN] to 'ENABLE'
   - PMC_IMPL_PART_XUSBC_POWER_GATE_CONTROL_0[SRAM_SLEEP_EN] to 'ON'
   - PMC_IMPL_PART_XUSBC_POWER_GATE_CONTROL_0[SRAM_RET_EN] to 'OFF'
   - PMC_IMPL_PART_XUSBC_POWER_GATE_CONTROL_0[LOGIC_SLEEP] to 'ON'
   - PMC_IMPL_PART_XUSBC_POWER_GATE_CONTROL_0[START] to '1'
2. Read the following PMC register bit to confirm the power gating status of XUSB host:
   - PMC_IMPL_PART_XUSBC_POWER_GATE_CONTROL_0[START] equals '0'

3. Read the following PMC register bits to confirm the power gating status of XUSB host:
   - PMC_IMPL_PART_XUSBC_POWER_GATE_STATUS_0[LOGIC_SLEEP_STS] equals 'ON'
   - PMC_IMPL_PART_XUSBC_POWER_GATE_STATUS_0[SRAM_SLEEP_EN_STS] equals 'ON'
   - PMC_IMPL_PART_XUSBC_POWER_GATE_STATUS_0[SRAM_RET_STS] equals 'OFF'
4. Program the following PMC register bits in a single write to disable the power rail to XUSB device:
   - PMC_IMPL_PART_XUSBB_POWER_GATE_CONTROL_0[INTER_PART_DELAY_EN] to 'ENABLE'
   - PMC_IMPL_PART_XUSBB_POWER_GATE_CONTROL_0[SRAM_SLEEP_EN] to 'ON'
   - PMC_IMPL_PART_XUSBB_POWER_GATE_CONTROL_0[SRAM_RET_EN] to 'OFF'
   - PMC_IMPL_PART_XUSBB_POWER_GATE_CONTROL_0[LOGIC_SLEEP] to 'ON'
   - PMC_IMPL_PART_XUSBB_POWER_GATE_CONTROL_0[START] to '1'
5. Read the following PMC register bit to confirm the power gating status of XUSB device:
   - PMC_IMPL_PART_XUSBB_POWER_GATE_CONTROL_0[START]
6. Read the following PMC register bits to confirm the power gating status of XUSB device:
   - PMC_IMPL_PART_XUSBB_POWER_GATE_STATUS_0[LOGIC_SLEEP_STS] equals 'ON'
   - PMC_IMPL_PART_XUSBB_POWER_GATE_STATUS_0[SRAM_SLEEP_STS] equals 'ON'
   - PMC_IMPL_PART_XUSBB_POWER_GATE_STATUS_0[SRAM_RET_STS] equals 'OFF'
7. Program the following PMC register bits in a single write to disable the power rail to XUSB SuperSpeed:
   - PMC_IMPL_PART_XUSBB_POWER_GATE_CONTROL_0[INTER_PART_DELAY_EN] to 'ENABLE'
   - PMC_IMPL_PART_XUSBB_POWER_GATE_CONTROL_0[SRAM_SLEEP_EN] to 'ON'
   - PMC_IMPL_PART_XUSBB_POWER_GATE_CONTROL_0[SRAM_RET_EN] to 'OFF'
   - PMC_IMPL_PART_XUSBB_POWER_GATE_CONTROL_0[LOGIC_SLEEP] to 'ON'
   - PMC_IMPL_PART_XUSBB_POWER_GATE_CONTROL_0[START] to '1'
8. Poll the following PMC register bit to become '0' to ensure power rail is disabled:
   - PMC_IMPL_PART_XUSBB_POWER_GATE_CONTROL_0[START]
9. Read the following PMC register bits to confirm the power gating status of XUSB device:
   - PMC_IMPL_PART_XUSBB_POWER_GATE_STATUS_0[LOGIC_SLEEP_STS] equals 'ON'
   - PMC_IMPL_PART_XUSBB_POWER_GATE_STATUS_0[SRAM_SLEEP_STS] equals 'ON'
   - PMC_IMPL_PART_XUSBB_POWER_GATE_STATUS_0[SRAM_RET_STS] equals 'OFF'

Step 8

The xHCI PEP driver and XUSB Device Mode driver initiate the signal sequence to enable the XUSB SS wake detection logic for the SuperSpeed ports assigned to host and device.

1. Write '1' to the following USB PADCTL register bits to assert the vcore_off signal:
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP3_ELPG_VCORE_DOWN]
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP2_ELPG_VCORE_DOWN]
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP1_ELPG_VCORE_DOWN]

- XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP0_ELPG_VCORE_DOWN]

9.2.3.3.2 Host Mode Partition ELPG Entry

Step 1

The xHCI driver performs context save operation as described in the xHCI specification.

The xHCI PEP driver performs XUSB_HOST specific context save operations. In case the SuperSpeed partition has already been power gated, xHCI firmware should not save the context of the SuperSpeed partition again.

Step 2

The System Power Management driver programs the PMC USB 2.0 sleepwalk logic for ports assigned to XUSB host mode as described in the PMC Programming section.

The System Power Management driver should enable the wake events according to the port states and Wake-on-Connect, Wake-on-Disconnect, and Wake-on-Over-Current settings of the USB 2 ports accordingly.

Step 3

The xHCI PEP driver enables the XUSB wakeup interrupts for the SuperSpeed and USB 2.0 ports assigned to the host.

Step 4

The System Power Management driver flushes MCCIF and partition clients.

1. Set the following MC register bit to '1' to enable flush to XUSB:
   - MC_CLIENT_HOTRESET_CTRL_0[XUSB_HOST_FLUSH_ENABLE] for host mode
2. Read the following MC register bit to be '1' to ensure flush to XUSB is enabled:
   - MC_CLIENT_HOTRESET_STATUS_0[XUSB_HOST_HOTRESET_STATUS] for host mode

Step 5

The System Power Management driver asserts reset to XUSB then disables its clocks.

1. Set the following CAR register bit to '1' to assert reset to XUSB:
   - CLK_RST_CONTROLLER_RST_DEV_XUSB_0[SWR_XUSB_HOST_RST] for host mode
2. Set the following CAR register bit to '1' to disable the clocks to individual XUSB partitions:
   - CLK_RST_CONTROLLER_CLK_OUT_ENB_XUSB_CLR_0[CLR_CLK_ENB_XUSB_HOST] for host mode

Step 6
The System Power Management driver disables the XUSB power rails.

1. Program the following PMC register bits in a single write to disable the power rail to XUSB host:
   - PMC_IMPL_PART_XUSBC_POWER_GATE_CONTROL_0[INTER_PART_DELAY_EN] to 'ENABLE'
   - PMC_IMPL_PART_XUSBC_POWER_GATE_CONTROL_0[SRAM_SLEEP_EN] to 'ON'
   - PMC_IMPL_PART_XUSBC_POWER_GATE_CONTROL_0[SRAM_RET_EN] to 'OFF'
   - PMC_IMPL_PART_XUSBC_POWER_GATE_CONTROL_0[LOGIC_SLEEP] to 'ON'
   - PMC_IMPL_PART_XUSBC_POWER_GATE_CONTROL_0[START] to '1'
2. Read the following PMC register bit to confirm the power gating status of XUSB host:
   - PMC_IMPL_PART_XUSBC_POWER_GATE_CONTROL_0[START] equals '0'
3. Read the following PMC register bits to confirm the power gating status of XUSB host
   - PMC_IMPL_PART_XUSBC_POWER_GATE_STATUS_0[LOGIC_SLEEP_STS] equals 'ON'
   - PMC_IMPL_PART_XUSBC_POWER_GATE_STATUS_0[SRAM_SLEEP_STS] equals 'ON'
   - PMC_IMPL_PART_XUSBC_POWER_GATE_STATUS_0[SRAM_RET_STS] equals 'OFF'

9.2.3.3.3 Device Mode Partition ELPG Entry

Device mode partition should only be power gated when the device port is not connected.

Step 1

The XUSB Device Mode driver performs XUSB_DEVICE specific context save operations.

Step 2

The System Power Management driver programs the PMC to enable wake on connect where connection event is triggered by valid VBUS.

Step 3

The System Power Management driver flushes MCCIF and partition clients.

1. Set the following MC register bit to '1' to enable flush to XUSB:
   - MC_CLIENT_HOTRESET_CTRL_0[XUSB_DEV_FLUSH_ENABLE] for device mode
2. Read the following MC register bit to be '1' to ensure flush to XUSB is enabled:
   - MC_CLIENT_HOTRESET_STATUS_0[XUSB_DEV_HOTRESET_STATUS] for device mode

Step 4

The System Power Management driver asserts reset to XUSB then disables its clocks.

1. Set the following CAR register bit to '1' to assert reset to XUSB:
   - CLK_RST_CONTROLLER_RST_DEV_XUSB_0[SWR_XUSB_DEV_RST] for device mode
2. Set the following CAR register bit to '1' to disable the clocks to individual XUSB partitions:

- CLK_RST_CONTROLLER_CLK_OUT_ENB_XUSB_CLR_0[CLR_CLK_ENB_XUSB_DEV] for device mode

Step 5

The System Power Management driver disables the XUSB power rails.

1. Programming the following PMC register bits in a single write to disable the power rail to XUSB device:
   - PMC_IMPL_PART_XUSBB_POWER_GATE_CONTROL_0[INTER_PART_DELAY_EN] to 'ENABLE'
   - PMC_IMPL_PART_XUSBB_POWER_GATE_CONTROL_0[SRAM_SLEEP_EN] to 'ON'
   - PMC_IMPL_PART_XUSBB_POWER_GATE_CONTROL_0[SRAM_RET_EN] to 'OFF'
   - PMC_IMPL_PART_XUSBB_POWER_GATE_CONTROL_0[LOGIC_SLEEP] to 'ON'
   - PMC_IMPL_PART_XUSBB_POWER_GATE_CONTROL_0[START] to '1'
2. Read the following PMC register bit to confirm the power gating status of XUSB device:
   - PMC_IMPL_PART_XUSBB_POWER_GATE_CONTROL_0[START]
3. Read the following PMC register bits to confirm the power gating status of XUSB Device
   - PMC_IMPL_PART_XUSBB_POWER_GATE_STATUS_0[LOGIC_SLEEP_STS] equals 'ON'
   - PMC_IMPL_PART_XUSBB_POWER_GATE_STATUS_0[SRAM_SLEEP_STS] equals 'ON'
   - PMC_IMPL_PART_XUSBB_POWER_GATE_STATUS_0[SRAM_RET_STS] equals 'OFF'

Step 6

The XUSB Device Mode driver initiates the signal sequence to enable the XUSB SS wake detection logic for the SuperSpeed ports assigned to device as described in Step 7 of the All Partitions ELPG Entry section.

1. Write '1' to the following USB PADCTL register bit to assert the vcore_off signal.
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP3_ELPG_VCORE_DOWN]
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP2_ELPG_VCORE_DOWN]
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP1_ELPG_VCORE_DOWN]
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP0_ELPG_VCORE_DOWN]

9.2.3.3.4 All Partitions ELPG Exit

In the case the wake event is not generated by SuperSpeed ports, the SuperSpeed related programming should be skipped, where the SuperSpeed partition should exit ELPG due to wake events detected by SuperSpeed ports.

Step 1

The System Power Management driver enables the XUSB power rails.

1. Program the following PMC register bits in a single write to enable the power rail to XUSB host:

- PMC_IMPL_PART_XUSBC_POWER_GATE_CONTROL_0[SRAM_SLEEP_EN] to 'OFF'
- PMC_IMPL_PART_XUSBC_POWER_GATE_CONTROL_0[SRAM_RET_EN] to 'OFF'
- PMC_IMPL_PART_XUSBC_POWER_GATE_CONTROL_0[LOGIC_SLEEP] to 'OFF'
- PMC_IMPL_PART_XUSBC_POWER_GATE_CONTROL_0[START] to '1'

2. Poll the following PMC register bit to become '0' to ensure power rail is enabled
   - PMC_IMPL_PART_XUSBC_POWER_GATE_CONTROL_0[START]
3. Read the following PMC register bits to confirm the power gating status of XUSB host
   - PMC_IMPL_PART_XUSBC_POWER_GATE_STATUS_0[LOGIC_SLEEP_STS] equals 'OFF'
   - PMC_IMPL_PART_XUSBC_POWER_GATE_STATUS_0[SRAM_SLEEP_STS] equals 'OFF'
   - PMC_IMPL_PART_XUSBC_POWER_GATE_STATUS_0[SRAM_RET_STS] equals 'OFF'
4. Program the following PMC register bits to enable the power rail to XUSB Device
   - PMC_IMPL_PART_XUSBB_POWER_GATE_CONTROL_0[SRAM_SLEEP_EN] to 'OFF'
   - PMC_IMPL_PART_XUSBB_POWER_GATE_CONTROL_0[SRAM_RET_EN] to 'OFF'
   - PMC_IMPL_PART_XUSBB_POWER_GATE_CONTROL_0[LOGIC_SLEEP] to 'OFF'
   - PMC_IMPL_PART_XUSBB_POWER_GATE_CONTROL_0[START] to '1'
5. Poll the following PMC register bit to become '0' to ensure power rail is enabled
   - PMC_IMPL_PART_XUSBB_POWER_GATE_CONTROL_0[START]
6. Read the following PMC register bits to confirm the power gating status of XUSB Device
   - PMC_IMPL_PART_XUSBB_POWER_GATE_STATUS_0[LOGIC_SLEEP_STS] equals 'OFF'
   - PMC_IMPL_PART_XUSBB_POWER_GATE_STATUS_0[SRAM_SLEEP_STS] equals 'OFF'
   - PMC_IMPL_PART_XUSBB_POWER_GATE_STATUS_0[SRAM_RET_STS] equals 'OFF'
7. Program the following PMC register bits to enable the power rail to XUSB SuperSpeed
   - PMC_IMPL_PART_XUSBA_POWER_GATE_CONTROL_0[SRAM_SLEEP_EN] to 'OFF'
   - PMC_IMPL_PART_XUSBA_POWER_GATE_CONTROL_0[SRAM_RET_EN] to 'OFF'
   - PMC_IMPL_PART_XUSBA_POWER_GATE_CONTROL_0[LOGIC_SLEEP] to 'OFF'
   - PMC_IMPL_PART_XUSBA_POWER_GATE_CONTROL_0[START] to '1'
8. Poll the following PMC register bit to become '0' to ensure power rail is enabled
   - PMC_IMPL_PART_XUSBA_POWER_GATE_CONTROL_0[START]
9. Read the following PMC register bits to confirm the power gating status of XUSB SuperSpeed
   - PMC_IMPL_PART_XUSBA_POWER_GATE_STATUS_0[LOGIC_SLEEP_STS] equals 'OFF'
   - PMC_IMPL_PART_XUSBA_POWER_GATE_STATUS_0[SRAM_SLEEP_STS] equals 'OFF'
   - PMC_IMPL_PART_XUSBA_POWER_GATE_STATUS_0[SRAM_RET_STS] equals 'OFF'

Step 2

The System Power Management driver enables XUSB clocks.

1. Set the following CAR register bits to '1' to enable the clocks to individual XUSB partitions.

- CLK_RST_CONTROLLER_CLK_OUT_ENB_XUSB_SET_0[SET_CLK_ENB_XUSB_HOST] for host mode
- CLK_RST_CONTROLLER_CLK_OUT_ENB_XUSB_SET_0[SET_CLK_ENB_XUSB_DEV] for device mode
- CLK_RST_CONTROLLER_CLK_OUT_ENB_XUSB_SET_0[SET_CLK_ENB_XUSB_SS] for SS ports

2. Wait 2 µs

Step 3

The System Power Management driver removes power clamps to XUSB partitions.

1. Set the following PMC register bits to '0' to remove the power clamps to individual XUSB partitions.
   - PMC_IMPL_PART_XUSBC_CLAMP_CONTROL_0[CLAMP] for Host partition
   - PMC_IMPL_PART_XUSBB_CLAMP_CONTROL_0[CLAMP] for Device partition
   - PMC_IMPL_PART_XUSBA_CLAMP_CONTROL_0[CLAMP] for SuperSpeed partition
2. Wait 200 ns

Step 4

The System Power Management driver deasserts reset to XUSB.

1. Set the following CAR register bits to '0' to deassert reset to XUSB:
   - CLK_RST_CONTROLLER_RST_DEV_XUSB_0[SWR_XUSB_HOST_RST] for host mode
   - CLK_RST_CONTROLLER_RST_DEV_XUSB_0[SWR_XUSB_DEV_RST] for device mode
   - CLK_RST_CONTROLLER_RST_DEV_XUSB_0[SWR_XUSB_SS_RST] for SuperSpeed ports
2. Wait 1 µs

Step 5

The System Power Management driver disables flushes of MCCIF and partition clients.

1. Set the following MC register bits to '0' to disable flush to XUSB:
   - MC_CLIENT_HOTRESET_CTRL_0[XUSB_HOST_FLUSH_ENABLE] for host mode
   - MC_CLIENT_HOTRESET_CTRL_0[XUSB_DEV_FLUSH_ENABLE] for device mode

Step 6

The xHCI PEP driver and XUSB Device Mode driver disable the XUSB wakeup interrupts.

1. Set the following USB PADCTL register bits to '0' to disable the interrupts.

- XUSB_PADCTL_ELPG_PROGRAM_0_0[SS_PORT3_WAKE_INTERRUPT_ENABLE]
- XUSB_PADCTL_ELPG_PROGRAM_0_0[SS_PORT2_WAKE_INTERRUPT_ENABLE]
- XUSB_PADCTL_ELPG_PROGRAM_0_0[SS_PORT1_WAKE_INTERRUPT_ENABLE]

- XUSB_PADCTL_ELPG_PROGRAM_0_0[SS_PORT0_WAKE_INTERRUPT_ENABLE]
- XUSB_PADCTL_ELPG_PROGRAM_0_0[USB2_PORT3_WAKE_INTERRUPT_ENABLE]
- XUSB_PADCTL_ELPG_PROGRAM_0_0[USB2_PORT2_WAKE_INTERRUPT_ENABLE]
- XUSB_PADCTL_ELPG_PROGRAM_0_0[USB2_PORT1_WAKE_INTERRUPT_ENABLE]
- XUSB_PADCTL_ELPG_PROGRAM_0_0[USB2_PORT0_WAKE_INTERRUPT_ENABLE]

Step 7

The xHCI PEP driver and XUSB Device Mode driver initiate the signal sequence to disable the XUSB SS wake detection logic.

1. Write '0' to the following USB PADCTL register bits to deassert the vcore_off signal.
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP3_ELPG_VCORE_DOWN]
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP2_ELPG_VCORE_DOWN]
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP1_ELPG_VCORE_DOWN]
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP0_ELPG_VCORE_DOWN]
2. Wait 100 µs.
3. Write '0' to the following USB PADCTL register bits to deassert the clamp_en_early signals.
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP3_ELPG_CLAMP_EN_EARLY]
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP2_ELPG_CLAMP_EN_EARLY]
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP1_ELPG_CLAMP_EN_EARLY]
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP0_ELPG_CLAMP_EN_EARLY]
4. Wait 100 µs.
5. Write '0' to the following USB PADCTL register bits to deassert the clamp_en signals.
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP3_ELPG_CLAMP_EN]
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP2_ELPG_CLAMP_EN]
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP1_ELPG_CLAMP_EN]
   - XUSB_PADCTL_ELPG_PROGRAM_1_0[SSP0_ELPG_CLAMP_EN]

Step 8

The xHCI PEP driver performs XUSB register initialization.

The xHCI PEP driver performs XUSB context restore operation.

The XUSB Device Mode driver performs XUSB_DEVICE context restore operations.

Step 9

The xHCI PEP driver loads XUSB firmware.

The xHCI PEP driver notifies XUSB firmware whether context of SuperSpeed Partition should be restored.

Step 10

The xHCI driver performs context restore operations as described in the xHCI specification section.

Step 11

The System Power Management driver programs the PMC USB 2.0 sleepwalk logic for ports assigned to XUSB host mode and XUSB Device Mode to disable the sleepwalk logic as described in the PMC Programming section.

The xHCI PEP driver and XUSB Device Mode driver clear the XUSB wakeup events.

1. Write '1' to the following USB PADCTL register bits to clear the events.

- XUSB_PADCTL_ELPG_PROGRAM_0_0[SS_PORT3_WAKE_EVENT]
- XUSB_PADCTL_ELPG_PROGRAM_0_0[SS_PORT2_WAKE_EVENT]
- XUSB_PADCTL_ELPG_PROGRAM_0_0[SS_PORT1_WAKE_EVENT]
- XUSB_PADCTL_ELPG_PROGRAM_0_0[SS_PORT0_WAKE_EVENT]
- XUSB_PADCTL_ELPG_PROGRAM_0_0[USB2_PORT3_WAKE_EVENT]
- XUSB_PADCTL_ELPG_PROGRAM_0_0[USB2_PORT2_WAKE_EVENT]
- XUSB_PADCTL_ELPG_PROGRAM_0_0[USB2_PORT1_WAKE_EVENT]
- XUSB_PADCTL_ELPG_PROGRAM_0_0[USB2_PORT0_WAKE_EVENT]

9.2.3.3.5 Host Mode ELPG Exit

Step 1
The System Power Management driver enables the XUSB power rails.

1. Program the following PMC register bits in a single write to enable the power rail to XUSB host:
   - PMC_IMPL_PART_XUSBC_POWER_GATE_CONTROL_0[SRAM_SLEEP_EN] to 'OFF'
   - PMC_IMPL_PART_XUSBC_POWER_GATE_CONTROL_0[SRAM_RET_EN] to 'OFF'
   - PMC_IMPL_PART_XUSBC_POWER_GATE_CONTROL_0[LOGIC_SLEEP] to 'OFF'
   - PMC_IMPL_PART_XUSBC_POWER_GATE_CONTROL_0[START] to '1'
2. Poll the following PMC register bit to become '0' to ensure power rail is enabled
   - PMC_IMPL_PART_XUSBC_POWER_GATE_CONTROL_0[START]
3. Read the following PMC register bits to confirm the power gating status of XUSB host
   - PMC_IMPL_PART_XUSBC_POWER_GATE_STATUS_0[LOGIC_SLEEP_STS] equals 'OFF'
   - PMC_IMPL_PART_XUSBC_POWER_GATE_STATUS_0[SRAM_SLEEP_STS] equals 'OFF'
   - PMC_IMPL_PART_XUSBC_POWER_GATE_STATUS_0[SRAM_RET_STS] equals 'OFF'

Step 2
The System Power Management driver enables XUSB clocks.

1. Set the following CAR register bit to '1' to enable the clocks to individual XUSB partitions:

- CLK_RST_CONTROLLER_CLK_OUT_ENB_XUSB_SET_0[SET_CLK_ENB_XUSB_HOST] for host mode

2. Wait 2 µs

Step 3
The System Power Management driver removes power clamps to XUSB partitions.

1. Set the following PMC register bit to '0' to remove the power clamps to individual XUSB partitions:
   - PMC_IMPL_PART_XUSBC_CLAMP_CONTROL_0[CLAMP]
2. Wait 200 ns

Step 4
The System Power Management driver deasserts reset to XUSB.

1. Set the following CAR register bit to '0' to deassert reset to XUSB
   - CLK_RST_CONTROLLER_RST_DEV_XUSB_0[SWR_XUSB_HOST_RST] for host mode
2. Wait 1 µs

Step 5
The System Power Management driver disables flushes of MCCIF and partition clients.

1. Set the following MC register bits to '0' to disable flush to XUSB
   - MC_CLIENT_HOTRESET_CTRL_0[XUSB_HOST_FLUSH_ENABLE] for host mode

Step 6
The xHCI PEP driver disables the USB 2.0 wakeup interrupts for ports assigned to XUSB.

1. Set the following USB PADCTL register bits to '0' to disable the interrupts.

- XUSB_PADCTL_ELPG_PROGRAM_0_0[USB2_PORT3_WAKE_INTERRUPT_ENABLE]
- XUSB_PADCTL_ELPG_PROGRAM_0_0[USB2_PORT2_WAKE_INTERRUPT_ENABLE]
- XUSB_PADCTL_ELPG_PROGRAM_0_0[USB2_PORT1_WAKE_INTERRUPT_ENABLE]
- XUSB_PADCTL_ELPG_PROGRAM_0_0[USB2_PORT0_WAKE_INTERRUPT_ENABLE]

Step 7

The xHCI PEP driver performs XUSB register initialization.

The xHCI PEP driver performs XUSB context restore operations.

Step 8

The xHCI PEP Driver loads XUSB firmware.

The xHCI PEP Driver notifies XUSB firmware whether context of SuperSpeed Partition should be restored.

Step 9

The xHCI driver performs context restore operation as described in the xHCI specification.

Step 10

The System Power Management driver programs the PMC USB 2.0 sleepwalk logic for ports assigned to XUSB host mode to disable the sleepwalk logic as described in the PMC Programming section.

The xHCI PEP driver clears the XUSB wakeup events.

1. Write '1' to the following USB PADCTL register bits to clear the events.
   - XUSB_PADCTL_ELPG_PROGRAM_0_0[USB2_PORT3_WAKE_EVENT]
   - XUSB_PADCTL_ELPG_PROGRAM_0_0[USB2_PORT2_WAKE_EVENT]
   - XUSB_PADCTL_ELPG_PROGRAM_0_0[USB2_PORT1_WAKE_EVENT]
   - XUSB_PADCTL_ELPG_PROGRAM_0_0[USB2_PORT0_WAKE_EVENT]

9.2.3.3.6 Device Mode Partition ELPG Exit

Step 1

The System Power Management driver enables the XUSB power rails.

1. Programming the following PMC register bits in a single write to disable the power rail to the XUSB device:
   - PMC_IMPL_PART_XUSBB_POWER_GATE_CONTROL_0[SRAM_SLEEP_EN] to 'OFF'
   - PMC_IMPL_PART_XUSBB_POWER_GATE_CONTROL_0[SRAM_RET_EN] to 'OFF'
   - PMC_IMPL_PART_XUSBB_POWER_GATE_CONTROL_0[LOGIC_SLEEP] to 'OFF'
   - PMC_IMPL_PART_XUSBB_POWER_GATE_CONTROL_0[START] to '1'
2. Poll the following PMC register bit to become '0' to ensure power rail is enabled
   - PMC_IMPL_PART_XUSBB_POWER_GATE_CONTROL_0[START]
3. Read the following PMC register bits to confirm the power gating status of XUSB Device
   - PMC_IMPL_PART_XUSBB_POWER_GATE_STATUS_0[LOGIC_SLEEP_STS] equals 'OFF'
   - PMC_IMPL_PART_XUSBB_POWER_GATE_STATUS_0[SRAM_SLEEP_STS] equals 'OFF'
   - PMC_IMPL_PART_XUSBB_POWER_GATE_STATUS_0[SRAM_RET_STS] equals 'OFF'

Step 2

The System Power Management driver enables XUSB clocks.

1. Set the following CAR register bit to '1' to enable the clocks to individual XUSB partitions.
   - CLK_RST_CONTROLLER_CLK_OUT_ENB_XUSB_SET_0[SET_CLK_ENB_XUSB_DEV] for device mode
2. Wait 2 µs

Step 3

The XUSB Device Mode driver initiates the signal sequence to disable the XUSB SS wake detection logic for the SuperSpeed ports assigned to the device as described in Step 3 of the All Partitions ELPG Exit section.

Step 4

XUSB Device Mode driver disables the XUSB SS wakeup interrupts for the ports assigned to device as described in Step 4 of the All Partitions ELPG Exit section.

Step 5

The System Power Management driver removes power clamps to XUSB partitions:

1. Set the following PMC register bit to '0' to remove the power clamp to the XUSB device:
   - PMC_IMPL_PART_XUSBB_CLAMP_CONTROL_0[CLAMP] for Device partition
2. Wait 200 ns

Step 6

The System Power Management driver deasserts reset to XUSB.

1. Set the following CAR register bit to '0' to deassert reset to XUSB:
   - CLK_RST_CONTROLLER_RST_DEV_XUSB_0[SWR_XUSB_DEV_RST] for device mode
2. Wait 1 µs

Step 7

The System Power Management driver programs the PMC USB 2.0 sleepwalk logic for ports assigned to XUSB Device Mode to disable the sleepwalk logic as described in the Disable PMC Sleepwalk Logic section.

The XUSB Device Mode driver clears the XUSB wakeup events:

1. Write '1' to the following USB PADCTL register bits to clear the events.
   - XUSB_PADCTL_ELPG_PROGRAM_0_0[USB2_PORT3_WAKE_EVENT]
   - XUSB_PADCTL_ELPG_PROGRAM_0_0[USB2_PORT2_WAKE_EVENT]
   - XUSB_PADCTL_ELPG_PROGRAM_0_0[USB2_PORT1_WAKE_EVENT]
   - XUSB_PADCTL_ELPG_PROGRAM_0_0[USB2_PORT0_WAKE_EVENT]

Step 8

The XUSB Device Mode driver performs XUSB_DEVICE context restore operations.

Before power is to be removed from USB3, software should set the Device Mode to stop, then read the contexts from Device Mode MMIO registers and put them in system memory if required. After

power is restored to USB3 and hardware reset to USB3 is deasserted, software should pull the contexts from system memory and write them to the Device Mode registers. Software should reprogram the Device Mode registers to re-initialize the data structures and transfer rings before setting the Device Mode to run.

The following Device Mode contexts should be saved and restored by the driver in Linux systems and are assumed to be saved and restored by the OS in Windows systems – if this is not the case, the Windows PEP filter driver has to perform the context save and restore:

- PCI config registers
  - Command and Status
  - BAR0/1
  - Interrupt Line
  - FLADJ
  - MSI Message Control
  - MSI Address
  - MSI Data
  - Power Management Control and Status

The following Device Mode contexts should be saved and restored by the Device Mode driver:

- MMIO registers:
  - Device Address
  - U2 Timeout
  - Port Link State and Port State
  - Event Ring enqueue pointer and PCS

9.2.3.4 USB AO Programming

Sleepwalk logic in the USB AO is used for wake event detection of USB 2.0 ports, including wake on connect, wake on disconnect, and remote wakeup for XUSB.

9.2.3.4.1 Initialize USB AO Sleepwalk Logic

Step 1

The xHCI PEP driver initializes the sleepwalk logic:

1. Set the following registers to '0' to ensure sleepwalk logics are disabled
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_0[MASTER_ENABLE]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_1[MASTER_ENABLE]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_3[MASTER_ENABLE]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_2[MASTER_ENABLE]

2. Set the following registers to '1' to ensure sleepwalk logics are in low power mode
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_0[MASTER_CFG_SEL]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_1[MASTER_CFG_SEL]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_2[MASTER_CFG_SEL]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_3[MASTER_CFG_SEL]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_2[MASTER_CFG_SEL]
3. Program the following registers to '0x1' to set debounce time
   - XUSB_AO_USB_DEBOUNCE_DEL_0[UTMIP_LINE_DEB_CNT]
4. Set the following registers to '0' to ensure fake events of sleepwalk logic are disabled
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_0[FAKE_USBOP_EN]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_0[FAKE_USBON_EN]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_0[FAKE_USBOP_VAL]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_0[FAKE_USBON_VAL]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_1[FAKE_USBOP_EN]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_1[FAKE_USBON_EN]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_1[FAKE_USBOP_VAL]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_1[FAKE_USBON_VAL]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_2[FAKE_USBOP_EN]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_2[FAKE_USBON_EN]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_2[FAKE_USBOP_VAL]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_2[FAKE_USBON_VAL]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_3[FAKE_USBOP_EN]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_3[FAKE_USBON_EN]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_3[FAKE_USBOP_VAL]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_3[FAKE_USBON_VAL]
5. Set the following registers to '0' to ensure wake events of sleepwalk logic are not latched
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_0[LINE_WAKEUP_EN]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_1[LINE_WAKEUP_EN]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_2[LINE_WAKEUP_EN]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_3[LINE_WAKEUP_EN]
6. Program the following registers to 'NONE' to disable wake event triggers of sleepwalk logic
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_0[WAKE_VAL]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_1[WAKE_VAL]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_2[WAKE_VAL]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_2[WAKE_VAL]
7. Set the following registers to '1' to power down the line state detectors of the pad
   - XUSB_AO_UTMIP_PAD_CFG_0[USBOP_VAL_PD]
   - XUSB_AO_UTMIP_PAD_CFG_0[USBON_VAL_PD]

- XUSB_AO_UTMIP_PAD_CFG_1[USBOP_VAL_PD]
- XUSB_AO_UTMIP_PAD_CFG_1[USBON_VAL_PD]
- XUSB_AO_UTMIP_PAD_CFG_2[USBOP_VAL_PD]
- XUSB_AO_UTMIP_PAD_CFG_2[USBON_VAL_PD]
- XUSB_AO_UTMIP_PAD_CFG_3[USBOP_VAL_PD]
- XUSB_AO_UTMIP_PAD_CFG_3[USBON_VAL_PD]

9.2.3.4.2 Enable USB AO Sleepwalk Logic

Step 1

The xHCI PEP driver re-initializes the sleepwalk logic as described in the Initialize USB AO Sleepwalk Logic section.

Step 2

The xHCI PEP driver sets up the sleepwalk logic.

1. Program the following registers to match the speed of the port:
   - XUSB_AO_UTMIP_SAVED_STATE_0[SPEED]
   - XUSB_AO_UTMIP_SAVED_STATE_1[SPEED]
   - XUSB_AO_UTMIP_SAVED_STATE_2[SPEED]
   - XUSB_AO_UTMIP_SAVED_STATE_3[SPEED]
2. Set the following registers to '1' to enable the trigger of the sleepwalk logic:
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_0[WAKE_WALK_EN]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_0[LINEVAL_WALK_EN]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_1[WAKE_WALK_EN]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_1[LINEVAL_WALK_EN]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_2[WAKE_WALK_EN]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_2[LINEVAL_WALK_EN]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_3[WAKE_WALK_EN]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_3[LINEVAL_WALK_EN]
3. Write '1' to the following registers to reset the walk pointer and clear the alarm of the sleepwalk logic, as well as capture the configuration of the USB 2.0 pad:
   - XUSB_AO_UTMIP_TRIGGERS_0[CLR_WALK_PTR]
   - XUSB_AO_UTMIP_TRIGGERS_0[CLR_WAKE_ALARM]
   - XUSB_AO_UTMIP_TRIGGERS_0[CAP_CFG]
   - XUSB_AO_UTMIP_TRIGGERS_1[CLR_WALK_PTR]
   - XUSB_AO_UTMIP_TRIGGERS_1[CLR_WAKE_ALARM]
   - XUSB_AO_UTMIP_TRIGGERS_1[CAP_CFG]
   - XUSB_AO_UTMIP_TRIGGERS_2[CLR_WALK_PTR]

- XUSB_AO_UTMIP_TRIGGERS_2[CLR_WAKE_ALARM]
- XUSB_AO_UTMIP_TRIGGERS_2[CAP_CFG]
- XUSB_AO_UTMIP_TRIGGERS_3[CLR_WALK_PTR]
- XUSB_AO_UTMIP_TRIGGERS_3[CLR_WAKE_ALARM]
- XUSB_AO_UTMIP_TRIGGERS_3[CAP_CFG]

4. Program the following registers to set up the pull-ups and pull-downs of the signals during the four stages of sleepwalk
   - XUSB_AO_UTMIP_SLEEPWALK_0[USBOP_RPD_{A/B/C/D}] to '1'
   - XUSB_AO_UTMIP_SLEEPWALK_0[USBON_RPD_{A/B/C/D}] to '1'
   - XUSB_AO_UTMIP_SLEEPWALK_0[USBOP_RPU_{A/B/C/D}] to '0'
   - XUSB_AO_UTMIP_SLEEPWALK_0[USBON_RPU_{A/B/C/D}] to '0'
   - XUSB_AO_UTMIP_SLEEPWALK_1[USBOP_RPD_{A/B/C/D}] to '1'
   - XUSB_AO_UTMIP_SLEEPWALK_1[USBON_RPD_{A/B/C/D}] to '1'
   - XUSB_AO_UTMIP_SLEEPWALK_1[USBOP_RPU_{A/B/C/D}] to '0'
   - XUSB_AO_UTMIP_SLEEPWALK_1[USBON_RPU_{A/B/C/D}] to '0'
   - XUSB_AO_UTMIP_SLEEPWALK_2[USBOP_RPD_{A/B/C/D}] to '1'
   - XUSB_AO_UTMIP_SLEEPWALK_2[USBON_RPD_{A/B/C/D}] to '1'
   - XUSB_AO_UTMIP_SLEEPWALK_2[USBOP_RPU_{A/B/C/D}] to '0'
   - XUSB_AO_UTMIP_SLEEPWALK_2[USBON_RPU_{A/B/C/D}] to '0'
   - XUSB_AO_UTMIP_SLEEPWALK_3[USBOP_RPD_{A/B/C/D}] to '1'
   - XUSB_AO_UTMIP_SLEEPWALK_3[USBON_RPD_{A/B/C/D}] to '1'
   - XUSB_AO_UTMIP_SLEEPWALK_3[USBOP_RPU_{A/B/C/D}] to '0'
   - XUSB_AO_UTMIP_SLEEPWALK_3[USBON_RPU_{A/B/C/D}] to '0'
5. Program the following registers to set up the driving values of the signals during the four stages of sleepwalk (if the port speed is not unknown)
   - XUSB_AO_UTMIP_SLEEPWALK_0[AP_A] to '1' for HS/FS and '0' for LS
   - XUSB_AO_UTMIP_SLEEPWALK_0[AN_A] to '0' for HS/FS and '1' for LS
   - XUSB_AO_UTMIP_SLEEPWALK_0[HIGNZ_A] to '1'
   - XUSB_AO_UTMIP_SLEEPWALK_0[AP_{B/C/D}] to '0' for HS/FS and '1' for LS
   - XUSB_AO_UTMIP_SLEEPWALK_0[AN_{B/C/D}] to '1' for HS/FS and '0' for LS
   - XUSB_AO_UTMIP_SLEEPWALK_0[HIGHZ_{B/C/D}] to '0'
   - XUSB_AO_UTMIP_SLEEPWALK_1[AP_A] to '1' for HS/FS and '0' for LS
   - XUSB_AO_UTMIP_SLEEPWALK_1[AN_A] to '0' for HS/FS and '1' for LS
   - XUSB_AO_UTMIP_SLEEPWALK_1[HIGNZ_A] to '1'
   - XUSB_AO_UTMIP_SLEEPWALK_1[AP_{B/C/D}] to '0' for HS/FS and '1' for LS
   - XUSB_AO_UTMIP_SLEEPWALK_1[AN_{B/C/D}] to '1' for HS/FS and '0' for LS
   - XUSB_AO_UTMIP_SLEEPWALK_1[HIGHZ_{B/C/D}] to '0'
   - XUSB_AO_UTMIP_SLEEPWALK_2[AP_A] to '1' for HS/FS and '0' for LS

- XUSB_AO_UTMIP_SLEEPWALK_2[AN_A] to '0' for HS/FS and '1' for LS
- XUSB_AO_UTMIP_SLEEPWALK_2[HIGNZ_A] to '1'
- XUSB_AO_UTMIP_SLEEPWALK_2[AP_{B/C/D}] to '0' for HS/FS and '1' for LS
- XUSB_AO_UTMIP_SLEEPWALK_2[AN_{B/C/D}] to '1' for HS/FS and '0' for LS
- XUSB_AO_UTMIP_SLEEPWALK_2[HIGHZ_{B/C/D}] to '0'
- XUSB_AO_UTMIP_SLEEPWALK_3[AP_A] to '1' for HS/FS and '0' for LS
- XUSB_AO_UTMIP_SLEEPWALK_3[AN_A] to '0' for HS/FS and '1' for LS
- XUSB_AO_UTMIP_SLEEPWALK_3[HIGNZ_A] to '1'
- XUSB_AO_UTMIP_SLEEPWALK_3[AP_{B/C/D}] to '0' for HS/FS and '1' for LS
- XUSB_AO_UTMIP_SLEEPWALK_3[AN_{B/C/D}] to '1' for HS/FS and '0' for LS
- XUSB_AO_UTMIP_SLEEPWALK_3[HIGHZ_{B/C/D}] to '0'

6. Program the following registers to set up the driving values of the signals during the four stages of sleepwalk (if port speed is not determined):
   - XUSB_AO_UTMIP_SLEEPWALK_0[AP_{A/B/C/D}] to '0'
   - XUSB_AO_UTMIP_SLEEPWALK_0[AN_{A/B/C/D}] to '0'
   - XUSB_AO_UTMIP_SLEEPWALK_0[HIGHZ_{A/B/C/D}] to '1'
   - XUSB_AO_UTMIP_SLEEPWALK_1[AP_{A/B/C/D}] to '0'
   - XUSB_AO_UTMIP_SLEEPWALK_1[AN_{A/B/C/D}] to '0'
   - XUSB_AO_UTMIP_SLEEPWALK_1[HIGHZ_{A/B/C/D}] to '1'
   - XUSB_AO_UTMIP_SLEEPWALK_2[AP_{A/B/C/D}] to '0'
   - XUSB_AO_UTMIP_SLEEPWALK_2[AN_{A/B/C/D}] to '0'
   - XUSB_AO_UTMIP_SLEEPWALK_2[HIGHZ_{A/B/C/D}] to '1'
   - XUSB_AO_UTMIP_SLEEPWALK_3[AP_{A/B/C/D}] to '0'
   - XUSB_AO_UTMIP_SLEEPWALK_3[AN_{A/B/C/D}] to '0'
   - XUSB_AO_UTMIP_SLEEPWALK_3[HIGHZ_{A/B/C/D}] to '1'

Step 3

The xHCI driver puts the ports in the U3 suspend state.

Step 4

The xHCI PEP driver enables the wake event detections.

1. Set the following registers to '0' to power up the line state detectors of the pad:
   - XUSB_AO_UTMIP_PAD_CFG_0[USBOP_VAL_PD]
   - XUSB_AO_UTMIP_PAD_CFG_0[USBON_VAL_PD]
   - XUSB_AO_UTMIP_PAD_CFG_1[USBOP_VAL_PD]
   - XUSB_AO_UTMIP_PAD_CFG_1[USBON_VAL_PD]
   - XUSB_AO_UTMIP_PAD_CFG_2[USBOP_VAL_PD]
   - XUSB_AO_UTMIP_PAD_CFG_2[USBON_VAL_PD]

- XUSB_AO_UTMIP_PAD_CFG_3[USBOP_VAL_PD]
- XUSB_AO_UTMIP_PAD_CFG_3[USBON_VAL_PD]

2. Wait 1 µs.
3. Set the following registers to '1' to switch the electric control of the USB 2.0 pad to XUSB_AO:
   - XUSB_AO_UTMIP_PAD_CFG_0[FSLS_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_0[TRK_CTRL_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_0[RPD_CTRL_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_0[RPU_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_0[VREG_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_1[FSLS_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_1[TRK_CTRL_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_1[RPD_CTRL_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_1[RPU_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_1[VREG_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_2[FSLS_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_2[TRK_CTRL_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_2[RPD_CTRL_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_2[RPU_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_2[VREG_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_3[FSLS_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_3[TRK_CTRL_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_3[RPD_CTRL_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_3[RPU_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_3[VREG_USE_XUSB_AO]
4. Program the following registers to set the wake signaling trigger events:
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_0[WAKE_VAL] to 'ANY'
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_1[WAKE_VAL] to 'ANY'
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_2[WAKE_VAL] to 'ANY'
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_3[WAKE_VAL] to 'ANY'
5. Set the following registers to '1' to enable the wake detection:
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_0[MASTER_ENABLE]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_0[LINE_WAKEUP_EN]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_1[MASTER_ENABLE]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_1[LINE_WAKEUP_EN]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_2[MASTER_ENABLE]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_2[LINE_WAKEUP_EN]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_3[MASTER_ENABLE]

- XUSB_AO_UTMIP_SLEEPWALK_CFG_3[LINE_WAKEUP_EN]

9.2.3.4.3 Disable USB AO Sleepwalk Logic

Step 1
The xHCI PEP driver disables the wake event detections.

1. Set the following registers to '0' to disable the wake detection
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_0[MASTER_ENABLE]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_0[LINE_WAKEUP_EN]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_1[MASTER_ENABLE]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_1[LINE_WAKEUP_EN]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_2[MASTER_ENABLE]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_2[LINE_WAKEUP_EN]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_3[MASTER_ENABLE]
   - XUSB_AO_UTMIP_SLEEPWALK_CFG_3[LINE_WAKEUP_EN]
2. Set the following registers to '0' to switch the electric control of the USB 2.0 pad to XUSB vcore logic
   - XUSB_AO_UTMIP_PAD_CFG_0[FSLS_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_0[TRK_CTRL_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_0[RPD_CTRL_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_0[RPU_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_0[VREG_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_1[FSLS_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_1[TRK_CTRL_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_1[RPD_CTRL_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_1[RPU_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_1[VREG_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_2[FSLS_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_2[TRK_CTRL_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_2[RPD_CTRL_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_2[RPU_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_2[VREG_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_3[FSLS_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_3[TRK_CTRL_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_3[RPD_CTRL_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_3[RPU_USE_XUSB_AO]
   - XUSB_AO_UTMIP_PAD_CFG_3[VREG_USE_XUSB_AO]
3. Program the following registers to 'NONE' to disable wake event triggers of sleepwalk logic

- XUSB_AO_UTMIP_SLEEPWALK_CFG_0[WAKE_VAL]
- XUSB_AO_UTMIP_SLEEPWALK_CFG_1[WAKE_VAL]
- XUSB_AO_UTMIP_SLEEPWALK_CFG_2[WAKE_VAL]
- XUSB_AO_UTMIP_SLEEPWALK_CFG_3[WAKE_VAL]

4. Set the following registers to '1' to power down the line state detectors of the pad
   - XUSB_AO_UTMIP_PAD_CFG_0[USBOP_VAL_PD]
   - XUSB_AO_UTMIP_PAD_CFG_0[USBON_VAL_PD]
   - XUSB_AO_UTMIP_PAD_CFG_1[USBOP_VAL_PD]
   - XUSB_AO_UTMIP_PAD_CFG_1[USBON_VAL_PD]
   - XUSB_AO_UTMIP_PAD_CFG_2[USBOP_VAL_PD]
   - XUSB_AO_UTMIP_PAD_CFG_2[USBON_VAL_PD]
   - XUSB_AO_UTMIP_PAD_CFG_3[USBOP_VAL_PD]
   - XUSB_AO_UTMIP_PAD_CFG_3[USBON_VAL_PD]
5. Write '1' to the following registers to clear alarm of the sleepwalk logic
   - XUSB_AO_UTMIP_TRIGGERS_0[CLR_WAKE_ALARM]
   - XUSB_AO_UTMIP_TRIGGERS_1[CLR_WAKE_ALARM]
   - XUSB_AO_UTMIP_TRIGGERS_2[CLR_WAKE_ALARM]
   - XUSB_AO_UTMIP_TRIGGERS_3[CLR_WAKE_ALARM]

9.2.3.5 USB 2 Pad Tracking Programming

USB 2 Pad tracking logic should be triggered once during cold boot and SC7 exit to measure and capture the electric parameters of USB 2.0 pads.

Step 1

Sets up and enables the tracking clocks.

1. Set the following CAR register bits to '1' to enable the tracking clocks
   - CLK_RST_CONTROLLER_CLK_OUT_ENB_USB2_HSIC_TRK_SET_0[SET_CLK_ENB_USB2_TRK]
2. Program the following CAR register bits to set the divisor of tracking clocks
   - CLK_RST_CONTROLLER_CLK_SOURCE_USB2_HSIC_TRK_0[USB2_HSIC_TRK_CLK_DIVISOR]

Step 2

Set up the timing parameters:

1. Program the following XUSB registers:
   - XUSB_PADCTL_USB2_BIAS_PAD_CTL_1_0[TRK_START_TIMER] to '0x1E'
   - XUSB_PADCTL_USB2_BIAS_PAD_CTL_1_0[TRK_DONE_RESET_TIMER] to '0xA'

Step 3
Power up the pads and the tracking circuit:

1. Set the following XUSB register bits to power up the pads:
   - XUSB_PADCTL_USB2_BIAS_PAD_CTL_0_0[PD] to '0'
2. Wait 1 µs.
3. Set the following XUSB register bits to power up the tracking circuits and enable the hardware state machine:
   - XUSB_PADCTL_USB2_BIAS_PAD_CTL_1_0[PD_TRK] to '0'
4. Wait 100 µs.

Step 4
Disable the tracking clocks after tracking completed.

1. Set the following CAR register bits to '1' to disable the tracking clocks:
   - CLK_RST_CONTROLLER_CLK_OUT_ENB_USB2_HSIC_TRK_CLR_0[CLR_CLK_ENB_USB2_TRK]

9.2.3.6 Battery Charging over XUSB

9.2.3.6.1 Initialization

The common steps below need to be done to ensure that the pads are in the correct mode of operation for battery charging operations

Step 1

Ensure pad power rails are ON, AVDD_USB, and VCLAMP_USB rails are needed

Step 2

1. Power up BIAS Pad by writing '0' to the following register bit in XUSB_PADCTL
   - XUSB_PADCTL_USB2_BIAS_PAD_CTL_0_0[PD]
2. Put the USB 2 Pad0 in correct power mode for battery charging operations
   - Clear XUSB_PADCTL_USB2_OTG_PAD0_CTL_0_0[PD].
   - Clear XUSB_PADCTL_USB2_OTG_PAD0_CTL_0_0[PD_ZI].
   - Set XUSB_PADCTL_USB2_OTG_PAD0_CTL_0_0[PD2_OVRD_EN] and XUSB_PADCTL_USB2_OTG_PAD0_CTL_0_0[PD2].
   - Clear XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL0_0[PD_CHG].

Step 3

1. Take software control over DP/DN, pull ups and pull downs (these should be cleared once all battery charging operation is completed)
   - Set XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL1_0[USBON_RPD_OVRD].

- Set XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL1_0[USBON_RPU_OVRD].
- Set XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL1_0[USBOP_RPD_OVRD].
- Set to XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL1_0[USBOP_RPU_OVRD].

2. Remove all pull ups and pull downs
   - Clear XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL1_0 [USBON_RPD_OVRD_VAL].
   - Clear XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL1_0 [USBON_RPU_OVRD_VAL].
   - Clear XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL1_0 [USBOP_RPD_OVRD_VAL].
   - Clear XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL1_0 [USBOP_RPU_OVRD_VAL].
3. Enable battery charging related filters
   - Write '0xa' to XUSB_PADCTL_USB2_BATTERY_CHRG_TDCD_DBNC_TIMER_0[TDCD_DBNC].
   - Set XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL0_0[VDCD_DET_FILTER_EN].
   - Set XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL0_0[VDAT_DET_FILTER_EN].
   - Set XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL0_0[ZIN_FILTER_EN].
   - Set XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL0_0[ZIP_FILTER_EN].

9.2.3.7 Data Contact Detection

Step 1

1. Clear stale status bits by writing '1' to the following register bits in XUSB_PADCTL (below can be done using a single register write)
   - XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL0_0[DCD_DETECTED]

Step 2

1. Set OP_I_SRC_EN and enable ONRPD
   - Set XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL1_0[USBON_RPD_OVRD_VAL].
   - Set XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL0_0[OP_I_SRC_EN].
2. Wait until data contact detection (DCD) reports data contact:
   - Set XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL0_0[DCD_INTR_EN] and wait until hardware asserts interrupt **or,**
   - Poll XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL0_0[DCD_DETECTED] **or,**
   - Wait at least 300 ms and at most 900 ms if data contact is not reported.
3. Clear OP_I_SRC_EN and USBON_RPD
   - Clear XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL0_0[OP_I_SRC_EN].
   - Clear XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL1_0 [USBON_RPD_OVRD_VAL].
4. Clear DCD bit if it is set
   - Write '1' to clear XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL0_0[DCD_DETECTED].

9.2.3.8 Primary Charger Detection

1. Set OP_SRC_EN and ON_SINK_EN
   - Set XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL0_0[OP_SRC_EN].
   - Set XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL0_0[ON_SINK_EN].
2. Wait at least 10 ms.
3. Check XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL0_0[VDAT_DET] and XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL0_0[ZIN].
   - D- is larger than VLGC when both VDAT_DET and ZIN are 1
   - D- is between VLGC and VDAT_REF when VDAT_DET is 1 and ZIN are 0
   - D- is smaller than VDAT_REF when both VDAT_DET and ZIN are 0
4. Clear OP_SRC_EN and ON_SINK_EN
   - Clear XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL0_0[OP_SRC_EN].
   - Clear XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL0_0[ON_SINK_EN].

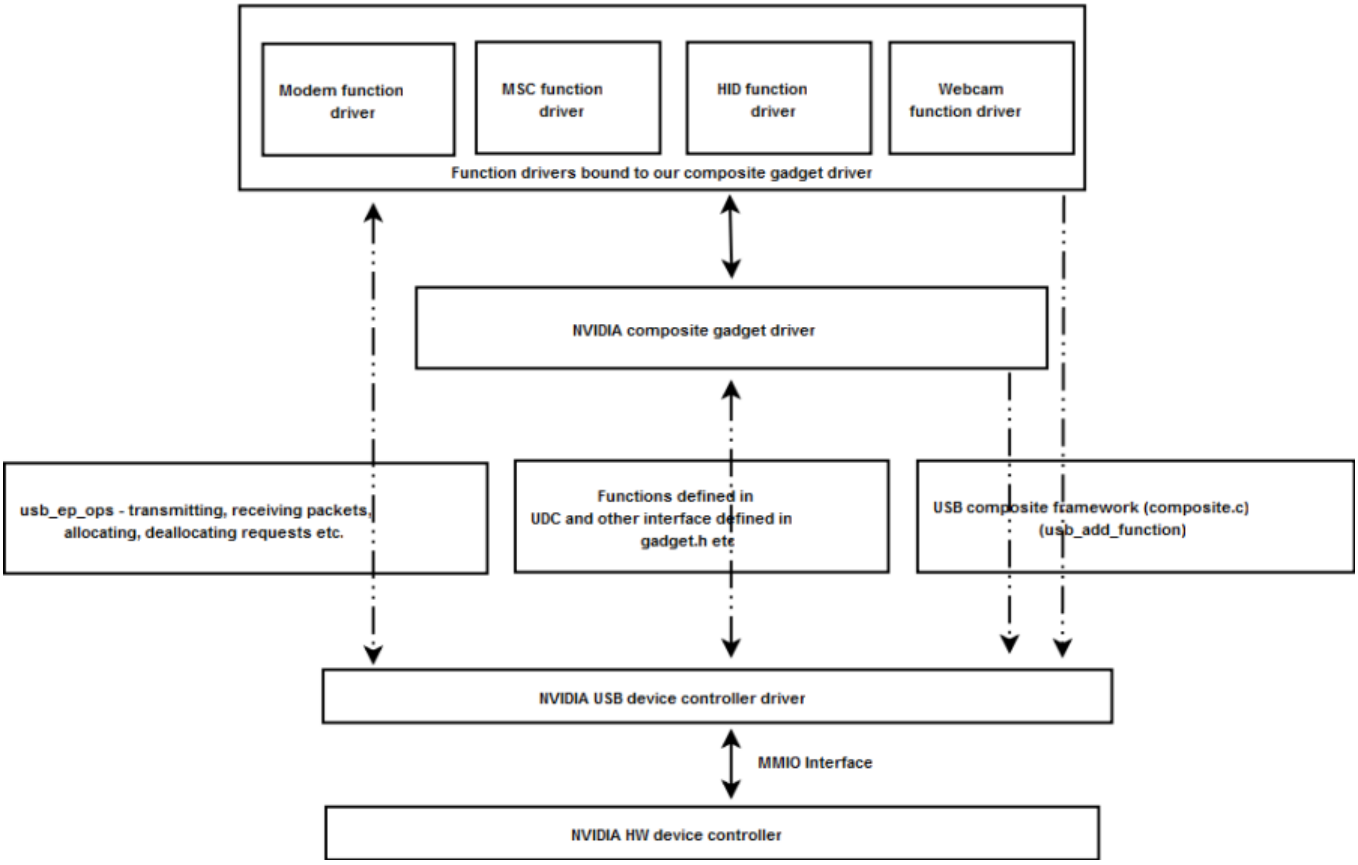
9.2.3.9 Secondary Charger Detection

1. Set ON_SRC_EN and OP_SINK_EN
   - Set XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL0_0[ON_SRC_EN].
   - Set XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL0_0[OP_SINK_EN].
2. Wait at least 10 ms.
3. Check XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL0_0[VDAT_DET] and XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL0_0[ZIP].
   - D- is larger than VLGC when both VDAT_DET and ZIP are 1
   - D- is between VLGC and VDAT_REF when VDAT_DET is 1 and ZIP are 0
   - D- is smaller than VDAT_REF when both VDAT_DET and ZIP are 0
4. Clear ON_SRC_EN and OP_SINK_EN
   - Clear XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL0_0[ON_SRC_EN].
   - Clear XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL0_0[OP_SINK_EN].

9.2.3.10 XUSB Programming Guidelines for Device Mode

The pictorial representation of the device mode software stack is shown in the figure below.

Figure 9.7 Device Mode Software Stack



9.2.3.10.1 Initialization

The sequence of steps to be performed to initialize the device controller is as follows:

Step 1

1. Allocate and initialize all device controller related memory data structures:
   - Device controller event ring segment table and the event ring segments that it points to.
   - Control endpoint contexts (one IN and one OUT) and the transfer rings that they point to.
2. Initialize the event ring segment table size register with number of entries in the event ring segment table.
3. Initialize the event ring segment table base address register with the physical memory address of the event ring segment table.
4. Initialize the event ring segment table dequeue pointer register with the physical memory address of the event ring segment pointed by event ring segment table entry 0.

Step 2

1. Initialize the endpoint context register with the physical memory address of the first entry in the array of endpoint contexts (control endpoints).

Step 3

1. Enable the device mode controller by writing '1' to enable bit (Enumeration mode) in the capability control register.
2. Update the 'Device Address' field in the control register upon successfully receiving the 'Set_Address' request from the host.

Step 4

1. Wait for Set_Configuration request from the host.
   - Initialize the required endpoint contexts and the transfer rings depending upon the configuration/interface selected.
     Software has to initialize all 32 EP contexts at this step.
   - Check for the correctness of the requested configuration with Set_Configuration as discussion in the subsequent sections.
2. Write '1' to 'Run' field (Run mode) in the control register as the last step in completing Set_Configuration request from the host.
   - The hardware fetches all 32 EP contexts before setting 'Run' bit to '1,' The valid EPs are identified by looking at EP.Type field in the endpoint context.

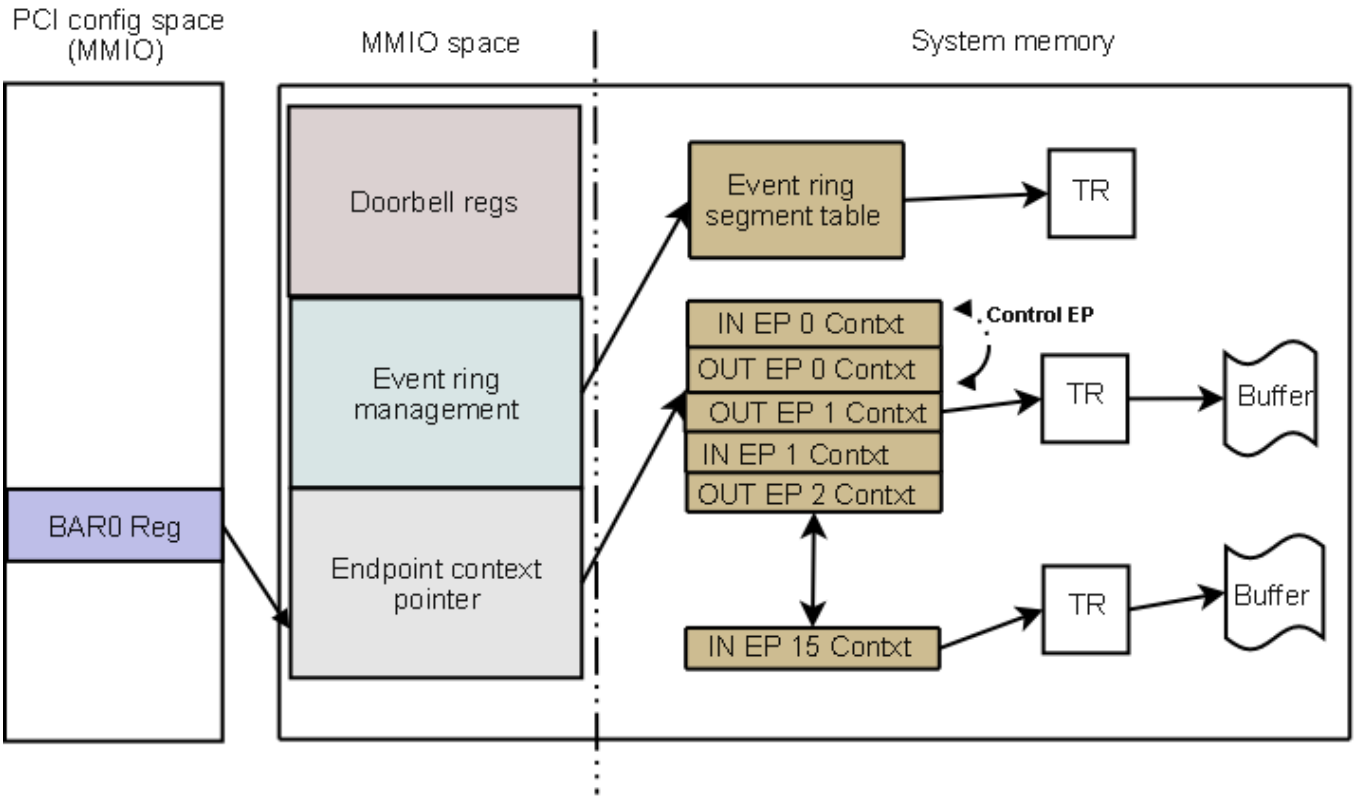
Note: All registers (PCI configuration space) are memory mapped in the Orin platform.

Note: The device controller has the option to generate an interrupt whenever an event is posted to the event ring (including SETUP).

9.2.3.10.2 Memory Map: Device Mode

The device mode software interacts with the device mode controller through a set of transfer rings, event rings, various data structures (contexts), and the register interface based on the xHCI specification.

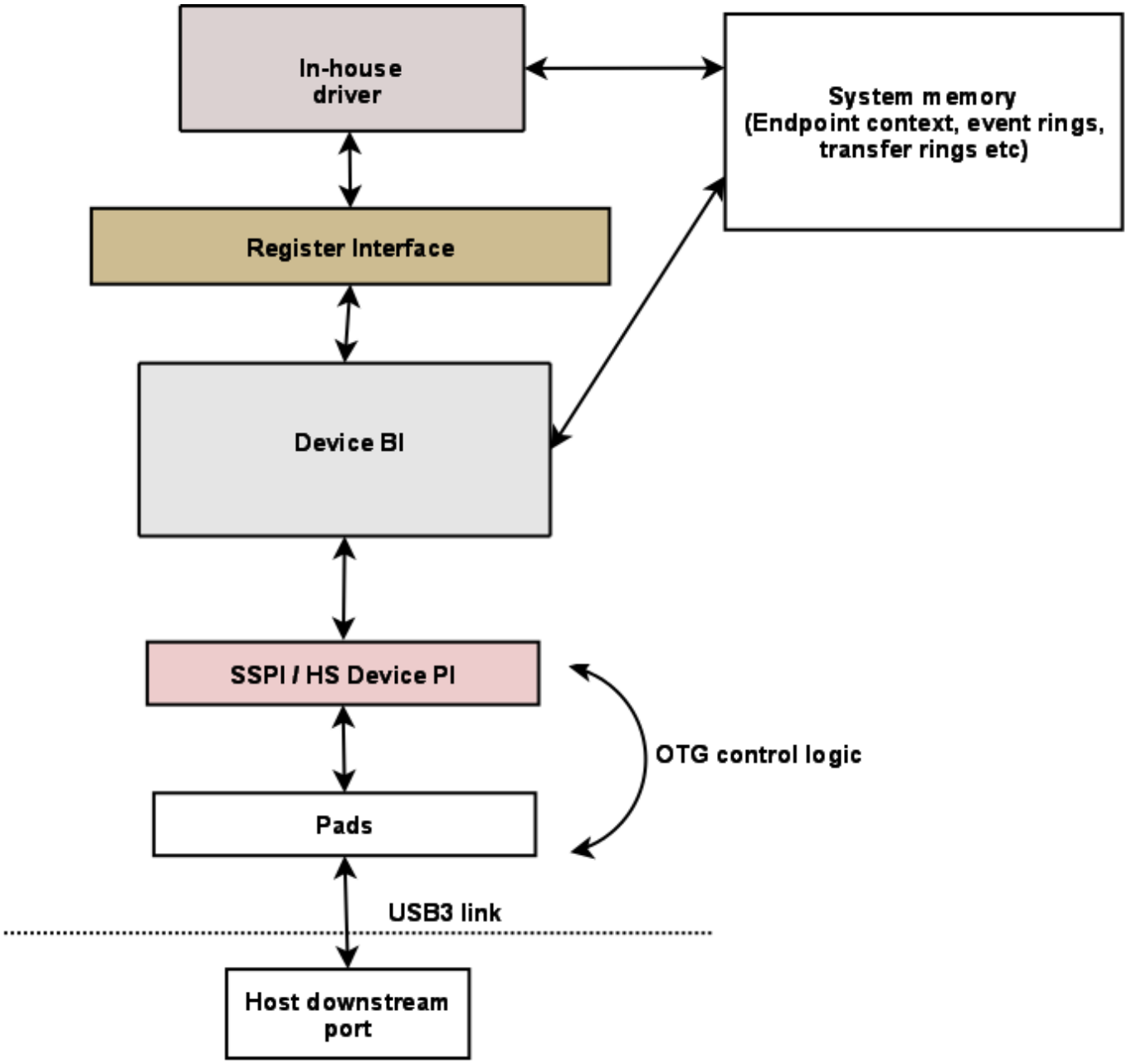
Figure 9.8 Memory Map for Device Mode



9.2.3.10.3 Hardware and Software Interaction

The device mode controller could operate either in super speed or high-speed mode.

Figure 9.9 Software/Hardware Interaction in Device Mode



9.2.3.10.4 Modes

Enumeration Mode

The device mode controller is in enumeration mode when the enable bit transitions from '0' to '1.' While in enumeration mode, the software services the standard USB enumeration related requests from the USB host through the default control endpoint. The standard requests that need to be supported by the software are:

- Get_Descriptor
- Get_Configuration
- Get_Interface
- Get_Status
- Synch_Frame
- Set_Address
- Set_Configuration
- Set_Feature
- Clear_Feature
- Set_Interface
- Set_Isoch_Delay
- Set_Sel

The software should also support OTG specific device descriptor fields as described in Chapter 6 of the USB3 OTG specification.

The firmware needs to handle the following device requests in the debug mode:

- Get_Configuration, Get_Descriptor, and Get_Status
- Set_Feature, Clear_Feature
- Set_configuration, Set_Address
- Set_Interface, Get_Interface
  - Only one interface support. Firmware returns STALL upon receiving the requests targeting the non-default interface.
- Set_Isoch_Delay, Set_Sel, Sync_Frame
  - No need to support these requests. Firmware returns STALL upon receiving these requests.

Set_configuration

The software behavior upon receiving this request varies depending upon the device state as discussed below.

Default state: The behavior of the device is unknown upon receiving this request in default state.

Addressed state: If the specified configuration is zero, then the device remains in the addressed state. If the specified configuration value matches the configuration value from a configuration descriptor, then that configuration is selected and the device enters the configured state. Otherwise, the software should respond with STALL by writing to Nth bit in the 'HALT' register.

Configured state: If the specified configuration value is zero, then the device enters the address state. If the specified configuration value matches the configuration value from a configuration descriptor, then that configuration is selected and the device remains in the configured state. Otherwise, the device responds with STALL by writing to Nth bit in the 'HALT' register.

Steps to Handle Set_configuration

In the Addressed State to set a new configuration

1. Program EP Context for each endpoint that is enabled by this configuration.
2. Set Reload bit for each Endpoint being enabled.
3. Set Run bit after setting Reload bit for first enabled Endpoint.
4. Wait for all Reload bits to be cleared.
5. Clear all Pause bits corresponding to the Reload bits that were set.
6. Wait for STCHG bits to be set to indicate that Pause has been cleared.
7. Clear all STCHG bits.
8. Clear Halt bits for all configured endpoints.
9. Clear all STCHG bits.
10. Complete Status Stage for the SET_CONFIG request.

In the Configured State to deconfigure the Device (SET_CONFIG 0)

- Pause all active Endpoints
- Program EP State Field in EP context to Disabled for all Endpoints except the default Control Endpoint
- Clear all Augmented EP Context fields to 0 for all endpoints except for the default Control Endpoint
- Clear Run bit in Control register
- Wait for all STCHG bits to be set
- Clear all STCHG bits
- Complete Status stage for SET_CONFIG request

In the Configured State to Select a Different Configuration

- Do all steps to deconfigure the device as mentioned in "In the Configured State to deconfigure the device (SET_CONFIG 0)" above
- Do all steps to select a new configuration as mentioned in "In the Addressed State to set a new configuration" above.

Removing an Endpoint as Part of the Select Alternate Interface

- Set Pause bit for the Endpoint to be disabled
- Wait for corresponding STCHG bit to be set
- Program EP State field in EP Context of the Endpoint to Disabled
- Clear all fields in Augmented area of EP context to 0
- Set Reload bit for the Endpoint to be disabled
- Wait for STCHG bit to be set
- Clear STCHG bit

Adding an Endpoint as Part of the Select Alternate Interface

- Initialize EP context for Endpoint to be enabled
- Set Reload bit for the Endpoint to be enabled
- Wait for STCHG bit to be set
- Clear STCHG bit

Modifying an Endpoint as Part of the Select Alternate Interface

- Do all steps in "Removing an Endpoint as part of the Select Alternate Interface" above to disable the endpoint
- Do all steps in "Adding an Endpoint as part of the Select Alternate Interface" above to enable the endpoint again with new parameters

Run Mode

The device mode switches from enumeration mode to run mode after receiving a "Set_Configuration" request from the host. It is the responsibility of the software to write '1' to 'Run bit' field in control register upon successfully receiving the 'Set_Configuration' request from the host. The software should also move the device state from addressed state to configured state.

Test Mode

The device mode switches from enumeration mode or run mode to test mode after receiving a "Set_Feature" request from the host with feature selector set to TEST_MODE. In the test mode, software should trigger the test mode by programming the USB 2.0 Port Test Control register to match the test mode selector 1 ms after the control transfer is completed.

Device Classes Supported

The device mode must support all device classes. However, the priority is given to BOT/UAS storage classes and mobile communication classes for driver development and verification.

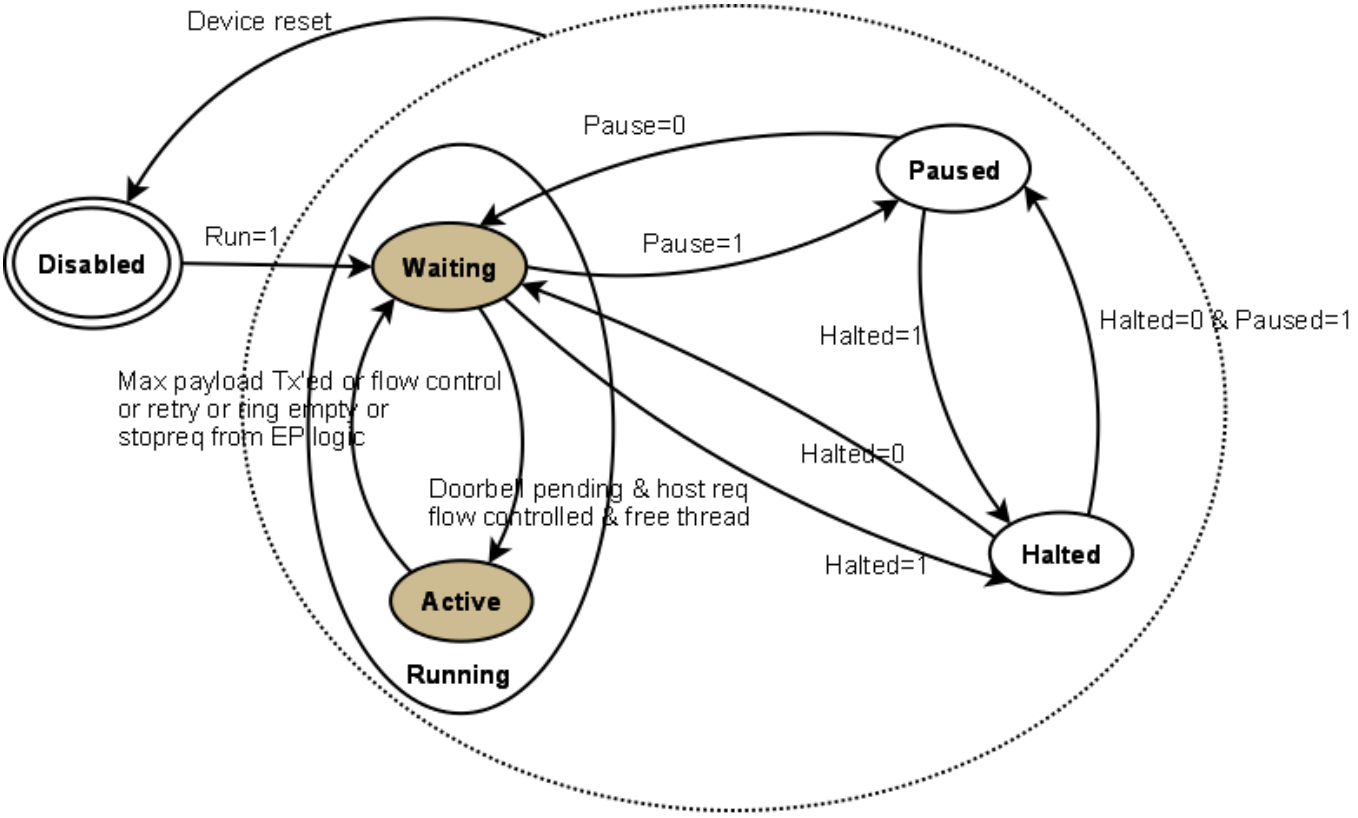
9.2.3.10.5 Endpoints

The device controller supports up to 32 endpoints (16 IN and 16 OUT). The endpoint 0 is the control endpoint which effectively reduces the number of non-control endpoints to 15 IN and 15 OUT. Each endpoint can be set up as IN and OUT direction and as Control, Bulk, or Interrupt transfer type. Isochronous endpoint is not supported.

Endpoint Contexts

The device uses 64-bytes version of the endpoint context. The data structure of the endpoint contexts for device is identical to host mode. However, the Endpoint context pointer points directly to Endpoint context 0 OUT (Control EP). The device controller owns the endpoint context as long as 'Run' bit is set 1.

Figure 9.10 EP State Transition



Halt an Endpoint

The software could put an endpoint into the halt state by programming HALT register. When in halt state, the device controller responds with STALL for any request from the host targeted at this endpoint. The device/debug controller never puts the endpoint into the halt state directly.

The sequence to halt an endpoint is as follows:

Step 1

If Run bit in the control register is set then proceed to Step 2; otherwise, exit.

Step 2

- To halt Nth IN endpoint, write '1' to 'Halt EP N IN' field in 'Halt Registers'
- To halt Nth OUT endpoint, write 'Halt EP N OUT' field in 'Halt Register'

Step 3

- Poll the Nth bit in 'State Change EP IN/OUT' register to make sure that EP has been halted by the hardware.

Pause an Endpoint

The endpoint could be put into pause state either by software or hardware in the device mode by writing to 'Pause EP N IN' or 'Pause EP N OUT' registers fields. When in paused state, the device mode controller responds with NRDY TPs for any host requests targeted at this endpoint.

The sequence to pause an Nth EP is as follows:

Step 1

If 'Run' bit in the control register is set, then proceed to Step 2; otherwise, exit.

Step 2

- To pause Nth IN endpoint, write '1' to 'Pause EP N IN' field in 'Pause Register'
- To halt Nth OUT endpoint, write '1' to 'Pause EP N OUT' field in 'Pause Register'

Step 3

- Poll the Nth bit in 'State Change EP IN/OUT' register to make sure that EP has been paused by the hardware.

Enabling ISO Endpoints

It is the software's responsibility to make sure that it pauses all non-ISO endpoints while enabling any ISO EP in device mode. This ensures that there are resources available in the device mode

controller while enabling any ISO endpoint. The non-ISO endpoints could be re-enabled once all ISO EPs are enabled.

Flow Control Threshold

This is purely for the use of the device controller. The software is responsible for enabling/disabling this feature and programming the threshold number for the consecutive IN/OUT request being flow controller.

9.2.3.10.6 Transfer Rings

The device mode controller makes use of transfer ring mechanism to pass the work from software to hardware. The operation/management of the transfer ring in device mode is similar to host mode. Refer to the xHCI specification to understand the transfer ring management. The software initializes the required number of transfer rings depending upon the configuration chosen. The device uses the TRBs and TDs to pass on the work to the hardware.

Add a TRB to the Transfer Ring

The sequence of steps to add a TRB to the transfer ring is as follows:

1. Make sure that a free entry is available to place the new TRB on to the transfer ring
2. Write the TRB to the transfer ring pointed by the enqueue pointer
   - Initialize the 'Cycle bit' field in the TRB to the PCS value.
3. Increment the enqueue pointer by the size of the TRB
4. Inform the hardware to fetch the TRB by writing to the doorbell register present in the MMIO space of the device controller.

Doorbell Registers

The software has to ring the doorbell upon adding a new TRB to a transfer ring. The device mode controller processes the transfer ring after the doorbell from the software.

Control Endpoint

There is one major difference while sending a doorbell register write to the control endpoint in the device mode compared to host mode. The difference is illustrated in the below sequence to handle the SETUP request.

1. A SETUP packet event TRB is received by the software/firmware.
   - The hardware passes a unique control sequence number (Say X) as part of the SETUP packet event TRB.
2. Software/firmware prepares the status and data stage TRB for the received SETUP request from the opposite side.

3. Software/firmware informs the hardware about the data and status stage TRB by writing 00h or 01h to 'Doorbell Target' and writing **X** to 'control sequence number' field in the doorbell registers

9.2.3.10.7 TRB Format

The TRB format of the various endpoints is similar to the format defined in the xHCI specification for the similar endpoints.

Bulk and Interrupt Endpoints

The bulk and interrupt endpoints use the normal TRB defined in the xHCI specification for their data transfers. Refer to Section 6.4.1.1 in the xHCI specification.

Control Endpoints

The control endpoints use the data and status stage TRB as defined in the xHCI specification. The data and status stage of the control transfer uses EP 0 context.

Isochronous Endpoints

The isochronous endpoints use the Isoc TRB defined in the xHCI specification. The debug mode does not support isochronous endpoints.

Micro-Frame Synchronizer

The device mode controller implements the local frame timer to track the micro-frames and in turn frame ID. The software could read the current Frame ID by accessing FrameID and uFrameID registers in MMIO space.

9.2.3.10.8 Event Ring

The operation and management of the event ring is similar to device debug capability mode described in xHCI specification. The device mode supports one event ring defined through event ring segment table, event ring segment table size, event ring segment table base address, and event ring dequeue pointer registers in the register interfaces. Hardware owns the event ring and the software/driver is the consumer of the event ring.

Note: Software should always program and use two segments for the event ring.

Event Ring Non-Empty

The hardware generates an interrupt upon posting an event TRB to the system memory. The 'Interrupt Pending' field in the status register (offset 0x24h) is set upon posting the event TRB. The

interrupt is generated depending upon whether the 'Interrupt Enable' field in the control register (offset 0x20h) is set or not.

The event TRBs are generated by the hardware for these transactions,

- IN DMA engine
- Out DMA engine
- Control (Setup), Flow control threshold
- Port status change

Setup Event TRB

The device controller posts a setup event TRB to the event ring upon receiving a setup transaction from the host. The hardware also generates an interrupt upon posting the event.

The setup data packet is sent as part of the setup event TRB posted to the event ring. The software uses Endpoint 0 (IN or OUT) to send data stage and status stage transactions in response to SETUP request from the host. The device controller reports errors if the sequence number or the direction of the data/status stage TRB does not match the control transfer.

Notes:

1. If a SETUP request is received when the control EP state is set to "Halted" by software then the hardware clears the halt condition.
2. A "serial number" field is included as part of the SETUP event TRB posted from the hardware. This is useful for the hardware to identify the valid SETUP transaction.
3. For example, a new SETUP request was received before the successful completion of the previous SETUP request of the same control endpoint.
4. An event TRB is posted by the hardware when there is a mismatch in the serial number stored locally (points to latest SETUP) in the hardware and the serial number present in the data/status TRBs fetched as part of the previous SETUP transaction.

Port Status Change Event Generation (PSCEG)

The hardware places an event TRB of type PSCEG whenever the following fields in the control register change:

- Connect status change (CSC)
- Port reset change (PRC)
- Port link state change (PLC)
- Port configuration link error (CEC)

It is the responsibility of the software to check and clear the above register fields upon receiving the PSCEG event TRBs. The format of the PSCEG event TRB is similar to xHCI defined event TRB with port ID set to 0x0.

Host Rejected Stream Selection Transfer Event TRB

This event TRB is posted to indicate the host rejection for a particular StreamID selected by the device.

Prime Pipe Received Transfer Event TRB

This event TRB is posted to indicate that host has issued a Prime Pipe transaction for a Bulk Stream Endpoint. Software can use this Event to reschedule a StreamID which has previously been rejected by host. Software should not ring Doorbell to a Stream Enabled Endpoint which had the Stream ID selection rejected by host, until it receives a Prime Pipe Received Transfer Event TRB.

Stream NumP Error Transfer Event TRB

This Event TRB is posted to indicate that host has violated NumP and PP fields in an ACK TP sent to a Stream enabled endpoint. Software should HALT the Stream Enabled endpoint on receiving this Transfer Event TRB.

Babble Detected Error Transfer Event TRB

This Event TRB is posted to indicate that host has sent excessive data payload that violates data length field in DP. Software should wait until hardware sets the EP State to Stopped, then clears the EP State and sets the Halt bit of the endpoint to trigger the endpoint to respond subsequent host request with STALL.

Interrupts

The device mode controller needs to support both MSI (one vector) and legacy interrupts. The hardware generates an interrupt upon posting an event TRB to the event ring. The interrupt generation is moderated (as in host mode) and there are dedicated registers to control the interrupt generation interval (Interrupt modulation interval) and to track the current interrupt counter value (Interrupt modulation counter).

9.2.3.10.9 Port

Generating a Device Notification Packet

The registers to generate the device notification packet are present in the MMIO space. However, these registers are not exposed to the debug software in debug mode and are implemented as internal registers accessible by the firmware. These registers valid are only for super speed mode and when the device controller is enabled.

The software sequence to generate a device notification packet is as follows:

1. Set the 'notification type' bit field in the device notification register

2. Set the 'notification type specific' bits field in the device notification register
3. Write '1' to 'device notification packet fire' bit to generate the packet. The bit is cleared by the hardware once the device notification packet has been sent on the USB link.

Host Connection

The device controller generates a Port Status Change Event upon detection of the host connection. It also sets the Interrupt pending bit in the status register whenever event ring is non-empty. An interrupt is generated to the software if Interrupt Enable bit is set.

Port Reset Handling

Hardware generates a Port Status Change event TRB upon detecting the port reset from the host. Upon receiving the event TRB, the software checks 'PR' bit in the control register for any port reset. The following sequence is executed upon port reset detection:

Step 1

1. Poll 'PRC' field in the control register to make sure that the reset sequence is complete.
2. Clear the 'PRC' field in the control registers once the reset sequence is completed.

Step 2

1. Make sure that the port is enabled by looking at 'Port Enabled/Disabled' field of the port status and control register.

Step 3

1. Reset the internal states of the software/driver.
2. Remove any pending TDs.
3. Restart transfer rings management.

Step 4

1. The device is in enumeration mode waiting to be configured from the host.

Step 5

1. Change the device state to 'addressed' once the 'Set_address' request is received from the host.
2. Write the address assigned to 'Device Address' field in the control register.

Step 6

1. Wait for 'Set_Configuration' request from the host.
   - Initialize the required endpoint contexts and the transfer rings depending upon the configuration/interface selected.
2. Clear 'Run Change' field in the control register.

3. Write '1' to run bit (Run mode) in the control register as the last step in completing Set_Configuration request from the host.

Link States

Connect

The OTG-B device generates a Port Status Change Event when Vbus presence or removal is detected. This is not true in the case of OTG A-device switching from host mode to device (Vbus always present).

The device mode asserts the PME when the Vbus presence is detected, and PME Enable bit and Wake on Connection bit are set to 0x1. The wake on signaling is controlled by PORTSC register in the case of debug mode.

Both device and debug modes asserts their PME when Vbus is removed, and the PME Enable bit and Wake on Disconnection bit are set to 0x1.

Link Power Management

The software can write to Port Link State field of Port Control register to manage the link states. The Port Link State field reflects the software initiated link state once the hardware completes the link state transition.

The device mode can initiate only SS U1 and SS U2 states. The software can initiate SS U1 and U2 entry/exit by programming the Port PM registers (With SS U2 timeout value, SS U1 Enable and SS Force Link PM Accept) as required. For high speed, the device mode responds to L1 depending upon the value programmed into PortPM registers (High speed L1 State). The U3/L2 state transition is always accepted by the device mode /debug mode (only U3).

Suspend

The software should not enable suspend, resume, and remote wakeup if the port is connected in high-speed and the port is acting either as a host in OTG B-Device or as a device in OTG A-Device. The resulting behavior is unknown for the above case. The software is responsible for EP states (disabled or not) when the links are in suspend state.

Resume: Device Initiated

The sequence of steps to generate the resume signaling is described below. The sequence remains the same for both device and debug mode.

1. The software decides to send the resume signaling to the opposite side.
2. Make sure Device Remote Wake Enable bit (USB 2.0) or Function Remote Wake Enable bit (USB 3.0) is set.
3. Transition the link to U0 by writing 'U0' to 'Port Link State' field in the port status and control register.

4. Write to the device notification register to generate a 'Function_WAKE' device notification packet (for USB 3.0 only).

Resume: Host initiated

The sequence upon detecting host initiated resume signaling is as follows,

1. A port status event TRB is posted upon detecting the resume signaling from the host.
2. The Port Link State field of the port control and status register is set to Resume
3. The software/debug-software should enable the link upon resume signaling by writing 'U0' to 'Port Link' State field.

9.2.3.10.10 Device Mode DVFS Support

Software should set the halt_ltssm and ltssm_state_change_pme_en bits for all SSPI before the SSPI clocks are lowered from their operation frequencies. Software should clear the halt_ltssm and ltssm_state_change_pme_en bits for all SSPI after the SSPI clocks are increased to their operation frequencies.

Software should increase the frequencies of SSPI clocks before clearing the halt_ltssm bit to allow LTSSM state transitions for the following LTSSM state transitions requests:

- Rx.Detect to Polling
- U3 to Recovery
- U2 to Recovery (note, this might not be enabled due to the shorter U2 exit time)

Software should clear the halt_ltssm bit to allow the following state transitions requested by SSPI, since they don't require SSPI clocks to be in operating frequency, and then set the halt_ltssm bit again to prevent further LTSSM state transitions:

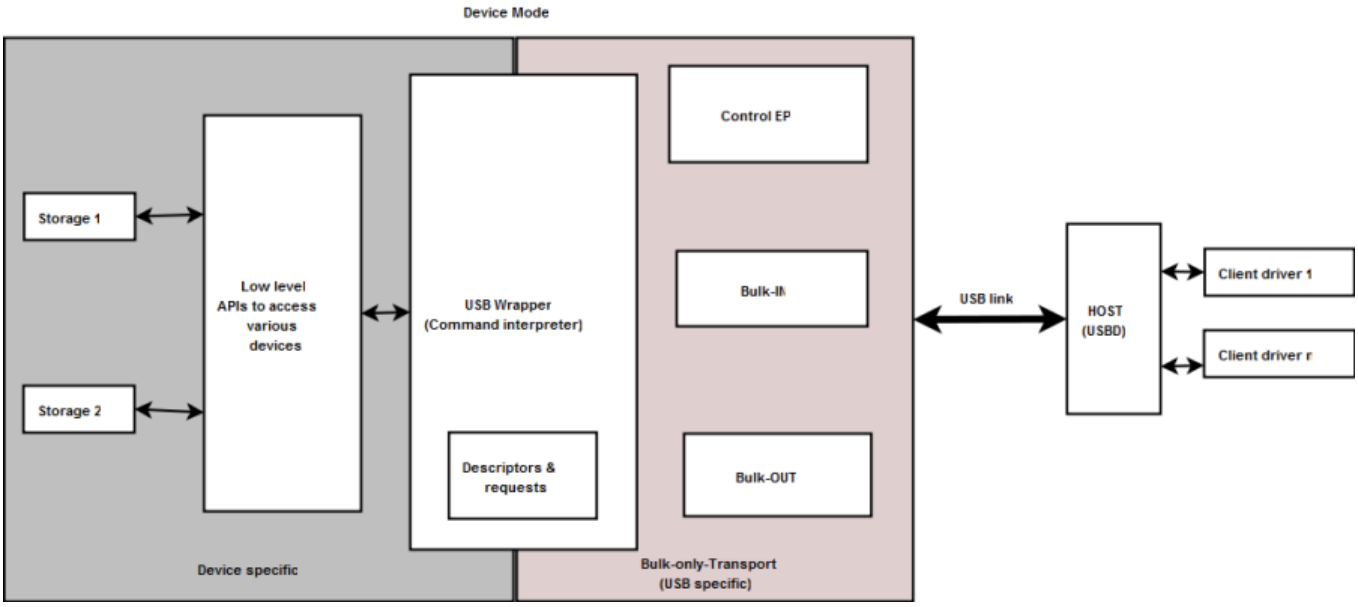
- Rx.Detect to SS.Disabled
- SS.Disabled to Rx.Detect
- U3 to Rx.Detect
- U2 to Rx.Detect (note, this might not be enabled due to the shorter U2 exit time)

9.2.3.10.11 Mass Storage Class

Bulk-Only Transport (BOT)

It makes use of one control EP, one bulk IN endpoint, and one bulk OUT endpoint to communicate with the opposite side (host port). However, the throughput achieved using BOT is less because this protocol does not support command queuing. In BOT, every transaction must be completed before the next transaction could be initiated by the host. For backward compatibility, BOT interface should always be declared as first interface.

Figure 9.11 BOT Diagram



BOT requests to be supported by the device driver are listed below.

Bulk-Only Mass Storage Reset (BOMSR)

This class-specific request shall ready the device for the next command from the host. This is sent through control endpoint.

bmRequestType	bRequest	wValue	wIndex	wLength	Data
00100001b	11111111b	0000h	Interface	0000h	none

The sequences of actions to be taken by the driver are as follows:

Step 1

- The device driver receives the BOMSR command through the SETUP packet.

Step 2

- Decode the type of command requested.

Step 3

- Place the corresponding request on to storage controller.

Step 4

- Wait for the completion of the storage controller reset.

Step 5

- Clear the internal variable associated with all interfaces enabled for the chosen configuration. (e.g., reset the transfer rings.)

Step 6

- Prepare and place the Data/status TRB on to the transfer ring once the storage reset is over. The hardware keeps on sending the NRDY until the status TRB is placed on the transfer ring.

Notes:

- Only BOMSR and Get Max Len requests are received through the control endpoints. All other requests are through bulk-endpoints. BOMSR is issued to clear the stall condition.
- The host sends CLEAR_HALT request for bulk-IN and bulk-OUT endpoints.

Get Max LUN

The host sends this command to find out the number of logical storages supported by the device. The BOT protocol supports up to 15 LUNs (0 to 15). Most likely, the multiple LUN is not supported with BOT.

The SETUP packet fields for the Get Max LUN are:

- bmRequestType: Class, Interface, device to host
  - bRequest field set to 254 (FEh)
  - wValue field set to 0
  - wIndex field set to the interface number
  - wLength field set to 1

bmRequestType	bRequest	wValue	wIndex	wLength	Data
10100001b	11111110b	0000h	Interface	0001h	1 byte

The device returns the number of LUNs supported as part of the data stage. The device could choose to send STALL response if multiple LUNs are not supported or could send zero as part of the data stage.

The driver sequence upon receiving the Get Max LUN is as follows,

Step 1

- Get Max LUN request is received as part of the SETUP request

Step 2

- If the command is not supported
  - Then HALT the endpoint by programming the HALT EP xx register.
- Else
  - Return the number of LUNs supported as part of the data stage.

Host Commands to the Mass Storage Device

Command Block Wrapper (CBW)

The host sends the CBW to the device's bulk OUT endpoint. The CBW contains a command block and other information about the command. The size of CBW is 31 bytes. The driver steps, upon receiving the CBW, is as follows:

Step 1

- Check the validity of the CBW structure
  - CBW is received after a CSW or reset.
  - The CBW is 31 bytes.
  - The dCBWSignature field has the correct value.

Step 2

- Check that the contents are valid.
  - All of the reserved bits are zero.
  - The bCBWLUN field contains a supported LUN value.
  - The bCBWCBLength and CBWCB fields are valid for the interface's subclass.

Step 3

- Prepare to receive or send the data to host.

Step 4

- If sending data then get the data from the internal storage device and place the corresponding TRBs on the transfer ring.

Step 5

- If receiving the data then get the data from the host and write it to the internal storage.

Data Phase

There is no data transfer between CBW phase and CSW phase if dCBWDataTransferLength is zero.

Common Status Write (CSW)

This is sent by the device to the host once the data transfer or the command phase is completed. The size of the CSW is 13 bytes. This is sent through the bulk-IN endpoint.

Notes:
The specification does not specify the timeout for host initiated requests targeted to mass storage device. The windows operation system implements around 20-30 seconds (some source, need to check)

STALL

The bulk-IN endpoints should be halted upon sending a STALL response to the opposite side. The STALL response is sent in the following scenarios,

- The device sends the less data than the requested amount during bulk-IN data phase.
- If the received CBW request is invalid.

Bulk-OUT could be halted or accept and discard any received data on the endpoint (preferred).

Notes:

- There is no support for bidirectional data transfer in a single command.
- BOT protocol does not support more than one outstanding CBW request to the device (command queuing)

Standard Descriptors

The descriptor data returned by the device to the hose is speed dependent. Depending upon the current operating mode (High speed or Superspeed), the device driver returns the configuration information corresponding to that mode only.

Table 9.2 Device Descriptor (one per device)

Offset	Field	Size	Value	Description
0	bLength	1	0x12	Size of this descriptor
1	bDescriptorType	1	0x1	
2	bcdUSB	2	0x0300h	
4	bDeviceClass	1	0	
5	bDeviceSubClass	1	0	
6	bDeviceProtocol	1	0	
7	bMaxPacketSize0	1	HS – 0x40 SS – 0x9	
8	idVendor	2		
10	idProduct	2		
12	bcdDevice	2		
14	iManufacturer	1		
15	iProduct	1		
16	iSerialNumber	1		
17	bNumConfigurations	1		

Configuration Descriptor

offset	Field	Size	Value	Description
0	bLength	1	0x9	
1	bDescriptorType	1	0x2	
2	wTotalLength	2		
4	bNumInterfaces	1		
5	bConfigurationValue	1		
6	iConfiguration	1		
7	bmAttributes	1	8b'01100000	Self powered – Yes Remote wakeup – Yes
8	bMaxPower	1		Expressed in units of 8mA for super speed.

Interface Configuration (both SS and HS)

Offset	Field	Size	Value	Description
0	bLength	1	0x9	
1	bDescriptorType	1	0x4	
2	bInterfaceNumber	1	1	
3	bAlternateSetting	1	0	UASP is secondary setting for backward compatibility. BOT is the primary alternate setting (0).
4	bNumEndpoints	1	2	
5	bInterfaceClass	1	0x8	
6	bInterfaceSubClass	1		
7	bInterfaceProtocol	1	0x50	BOT
8	iInterface	1		

Endpoint Configuration (both HS/SS)

Bulk-IN

Offset	Field	Size	Value	Description
0	bLength	1	0x7	
1	bDescriptorType	1	0x5	
2	bEndpointAddress	1	0x81	
3	bmAttributes	1	0x2	
4	wMaxPacketSize	2	HS – 512 SS – 1024	
6	bInterval	1	00h	

Bulk-OUT

Offset	Field	Size	Value	Description
0	bLength	1	0x7	
1	bDescriptorType	1	0x5	
2	bEndpointAddress	1	0x01	
3	bmAttributes	1	0x2	
4	wMaxPacketSize	2	HS – 512 SS – 1024	
6	bInterval	1	00h	

Notes:

- The values of some fields change as per the device of the speed.
- Refer to BOT specification to know more about the descriptor details.

Table 9.3 String Descriptor

Offset	Field	Size	Value	Description
0	bLength	1		
1	bDescriptor	1		
2	wString1	2		Serial number character 1
4	WString2	2		Serial number character 2
6				
8				

Offset	Field	Size	Value	Description
10	sString12	2		Serial number is at least 12 characters long

9.2.3.11 Recommended PHY Settings

Contact your local NVIDIA representative for PHY settings specific to your design.

9.2.3.12 BIAS PAD Configuration

The UTMIP BIAS pad is shared across USB_OTG and XUSB controllers and the below common bias pad settings need to be configured from the USB1 controller only.

USB1_UTMIP_BIAS_CFG0_0:

- UTMIP_BIASPD
- UTMIP_HSCHIRP_LEVEL
- UTMIP_HSSQUELCH_LEVEL
- UTMIP_HSDISCON_LEVEL_MSB
- UTMIP_HSDISCON_LEVEL
- UTMIP_ACTIVE_TERM_OFFSET
- UTMIP_ACTIVE_PULLUP_OFFSET
- UTMIP_VBUS_LEVEL_LEVEL
- UTMIP_SESS_LEVEL_LEVEL

USB1_UTMIP_BIAS_CFG1_0:

- UTMIP_FORCE_PDTRK_POWERUP
- UTMIP_FORCE_PDTRK_POWERDOWN

Following is the sequence to configure above parameters of the BIAS pad:

1. Enable the clock to the USB1 controller
2. Set any of above parameters as required.
3. Disable the clock to the USB1 controller if it is not used.

9.2.3.13 Boot ROM Initialization Sequence for USB Recovery

1. Program PLL_U.
   - Set the PLLU_BASE register fields, PLLU_DIVM, PLLU_DIVN, PLLU_VCO_FREQ, PLLU_BYPASS and PLLU_ENABLE fields as described in this document.
2. Configure USB_OTG

- Bring up USB_OTG clocks by writing 1 to CLK_ENB_USBD in the CLK_OUT_ENB_L register.
- Assert and deassert the master USBD reset in the CAR block (SWR_USBD_RST in the RST_DEVICES_L register) to bring USB_OTG out of reset.

  Stop the crystal clock by setting UTMIP_PHY_XTAL_CLOCKEN in the UTMIP_MISC_CFG1 register to 0. This only stops the crystal clocks in the UTMIP units.
  - The default value of USB1_UTMIP_PHY_XTAL_CLOCKEN (= 1) now changes to 0.
  - To use the A Session Valid for cable detection logic, set the USB1_VBUS_SENSE_CTL field in the USB1_LEGACY_CTRL register to A_SESS_VLD (2'b11).
- Program the automatic PLL start times.
- Set USB1_IF_UTMIP_PLLU_ENABLE_DLY_COUNT, UTMIP_PLLU_STABLE_COUNT, UTMIP_PLL_ACTIVE_DLY_COUNT and UTMIP_XTAL_FREQ_COUNT as per the values given in this document.
- Program the tracking duration.
- Set USB1_IF_UTMIP_BIAS_CFG1.UTMIP_BIAS_PDTRK_COUNT as per the values given in this document.
- Program the debouncer length times.
- Set UTMIP_DEBOUNCE_CFG0.UTMIP_BIAS_DEBOUNCE_A field.

  Program various static parameters of the USB_OTG UTMIP1.
  - Set UTMIP_TX_CFG0.UTMIP_FS_PREAMBLE_J to 0x1.
  - Set UTMIP_BAT_CHRG_CFG0.UTMIP_PD_CHRG to 1.
  - Set UTMIP_XCVR_CFG0.UTMIP_XCVR_LSBIAS_SEL to 0.
  - Set the third bit of UTMIP_SPARE_CFG0 to 1. i.e., UTMIP_SPARE_CFG0[3] to 1.
  - Set UTMIP_HSRX_CFG0. UTMIP_IDLE_WAIT as per the values given in this document.
  - Set UTMIP_HSRX_CFG0.UTMIP_ELASTIC_LIMIT to 16.
  - Set UTMIP_HSRX_CFG1.UTMIP_HS_SYNC_START_DLY to 9
- Restart the crystal clock by setting UTMIP_PHY_XTAL_CLOCKEN in the UTMIP_MISC_CFG1 register to 1 for USB_OTG.

3. Wait for cable connect on the USB_OTG UTMIP1 port. When cable is connected on USB_OTG UTMIP1 port, continue to the next step.
4. Bring UTMIP1 out of reset by writing 0 to the UTMIP_RESET bit of the USB1_IF_SUSP_CTRL register.
5. Wait until USB1_IF_USB_SUSP_CTRL.PHY_CLK_VALID is set to 1.
6. Then perform USB controller initialization.
   - Reset the bus.
   - Wait until the bus comes out of reset.
   - Set the controller in Device Mode.
   - Perform USB operations.

9.2.3.14 Performance Settings for USB Controllers

To meet USB's strict bandwidth/latency requirements, some AHB programming needs to be done. The following gives a guideline on the programming requirements to achieve maximum performance from USB:

- The burst size for the USB controller should be programmed to 8 in the USB2D_BURSTSIZE register. Both TXPBURST and RXPBURST fields should be programmed to the same value of 8.
- The ENB_FAST_REARBITRATE field for AHB_MEM gizmo should be set to 1 in the AHB_GIZMO_AHB_MEM register.
- The IMMEDIATE field for USB gizmos should be set to 1 in the AHB_GIZMO_USB, or AHB_GIZMO_USB2 register, depending on the controller in use.
- USB controllers should be set as high-priority masters on AHB by setting the bits corresponding to each USB controller to 1 in AHB_PRIORITY_SELECT field in the register AHB_ARBITRATION_PRIORITY_CTRL and setting the priority weight to 7 by setting the AHB_PRIORITY_WEIGHT field in the same register. USB master numbers are 6 for USB_OTG and 18 for USB 2. The priority weight could be relaxed depending on requirements from other AHB masters in the system as required for different use cases.
- The prefetch engine needs to be set up correctly to enable prefetching of transmit data packets for USB masters. Each USB master needs one channel on the prefetch engine. There are four channels on the prefetch engine. If all three USB masters enable one channel at the same time, it would leave one more channel for another AHB master. Each prefetch channel is controlled by the AHB_AHB_MEM_PREFETCH_CFG[NO] register, where NO = 1,2,3,4. To enable prefetch for a USB controller on a channel, program AHB_MST_ID_USB, or AHB_MST_ID_USB2 in the AHB_MST_ID field for the AHB_AHB_MEM_PREFETCH_CFG[NO] register. The ADDR_BNDRY field should be set to log2 (buffer size) according to the buffer size required for the corresponding USB master. The SPEC_THROTTLE field should be set to 0, and the INACTIVITY_TIMEOUT field should be set to 0x800.
- When a particular USB controller is in Host mode, the TXFIFOTHRES field in the USB2D_TXFILLTUNING register (offset 0x154) should be set to 0x10.

All this programming needs to be done before the RS bit in USB2D_USBCMD is set to RUN (1) for the corresponding USB controller.

9.2.3.15 USB Controller Handling of USB Resume Sequence

The USB controller handles a USB resume while the port is under PMC control:

1. The PMC maintains suspend mode on the bus.
2. When auto resume happens, the system starts restoring the USB controller.
3. Set the RUN bit from the USB controller to start SOFs.
4. The USB controller is brought back to suspend state.

5. Switch the USB bus from the PMC to the USB controller.
6. Wait until a resume complete notification from the USB controller. This is handled by the EHCI driver.
7. Further USB communication can start from here.

9.2.4 USB Registers

9.2.4.1 USB 3.0 Device Controller Memory-Mapped I/O Registers

T_XUSB_DEV_XHCI_SPARAM_0

Offset: 0x0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00XXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx))

Bit	Reset	Description
20:16	X	**ERSTMAX**: Event Ring Segment Table Max: Indicates the maximum value supported in the Event Ring Segment Table Base Size, where the maximum size of the Event Ring Segment Table is 2ERST Max **2** = VALUE
15:0	X	**RSVD0**: **1** = INIT

T_XUSB_DEV_XHCI_DB_0

Offset: 0x4
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000000XX (0b0000,0000,0000,0000,0000,0000,xxxx,xxxx)

Bit	R/W	Reset	Description
31:16	WO	0x0	**STREAMID**: Only significant for control EPs and is used to represent the control sequence number. Software is responsible for ensuring that the value written to this field matches the Sequence Number of a SETUP Packet Event STREAMID is not used for stream EPs. **0** = _00

Bit	R/W	Reset	Description
15:8	WO	0x0	**TARGET**: The target field represents the Endpoint ID to which the doorbell is targeted 00h: EP 0 Enqueue Pointer Update (Control EP 0) even values: OUT Enqueue Pointer Update odd values: IN Enqueue Pointer Update **0** = INIT
7:0	RO	X	**RSVD0**: **0** = _00

T_XUSB_DEV_XHCI_ERSTSZ_0

Event Ring Segment Table Size Register

Specifies the size of each event ring segment in terms of number of event TRB slots (each slot is 16 bytes) the min value of size in each register is 16 and the max is 4096 both the segments should be initialized with valid values.

Offset: 0x8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**ERST1SZ**: **0** = INIT
15:0	0x0	**ERST0SZ**: **0** = INIT

T_XUSB_DEV_XHCI_RSVD0_0

Offset: 0xc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RSVD0**: **0** = _00

T_XUSB_DEV_XHCI_ERST0BALO_0

Offset: 0x10
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0b0000,0000,0000,0000,0000,0000,0000,xxxx)

Bit	R/W	Reset	Description
31:4	RW	0x0	**ADDRLO**: **0** = INIT
3:0	RO	X	**RSVD0**: **0** = _00

T_XUSB_DEV_XHCI_ERST0BAHI_0

Offset: 0x14
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ADDRHI**: **0** = INIT

T_XUSB_DEV_XHCI_ERST1BALO_0

Offset: 0x18
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0b0000,0000,0000,0000,0000,0000,0000,xxxx)

Bit	R/W	Reset	Description
31:4	RW	0x0	**ADDRLO**: **0** = INIT
3:0	RO	X	**RSVD0**: **0** = _00

T_XUSB_DEV_XHCI_ERST1BAHI_0

Offset: 0x1c
Read/Write: R/W

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ADDRHI**: **0** = INIT

T_XUSB_DEV_XHCI_ERDPLO_0

Reflects the current value of ERDP, software updates this register after it has popped events.

Offset: 0x20
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0b0000,0000,0000,0000,0000,0000,0000,0xxx)

Bit	R/W	Reset	Description
31:4	RW	0x0	**ADDRLO**: **0** = INIT
3	RW	0x0	**EHB**: EHB is set by hardware when an event is posted and is cleared by software after it has processed all events **0** = INIT
2:0	RO	X	**RSVD0**: **0** = _00

T_XUSB_DEV_XHCI_ERDPHI_0

Offset: 0x24
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ADDRHI**: **0** = INIT

T_XUSB_DEV_XHCI_EREPLO_0

Reflects the current value of Event Ring Enqueue Pointer, it is updated by hardware after posting an event software can write to the register when Control Enable bit is cleared, software also specifies the segment index of the segment to be used and hardware samples the value written by software on a positive edge of control enable.

Offset: 0x28
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0b0000,0000,0000,0000,0000,0000,0000,xx00)

Bit	R/W	Reset	Description
31:4	RW	0x0	**ADDRLO**: **0** = INIT
3:2	RO	X	**RSVD0**: **0** = _00
1	RW	0x0	**SEGI**: **0** = INIT
0	RW	0x0	**ECS**: **0** = INIT

T_XUSB_DEV_XHCI_EREPHI_0

Offset: 0x2c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ADDRHI**: **0** = INIT

T_XUSB_DEV_XHCI_CTRL_0

Offset: 0x30
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00XXX20X (0b0000,0000,xxxx,xxxx,xxxx,0010,0000,xx00)
PROD: 0x80000012 (0b1xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1,xx1x)

Bit	R/W	Reset	Description
31	RW	0x0	**ENABLE**: Device control Enable used to enable device mode operation **0** = DIS **1** = EN
30:24	RW	0x0	**DEVADR**: Address assigned to the device DUT **0** = INIT
23:12	RO	X	**RSVD0**: **0** = _00
11	RW	0x0	**FORCE_FAULT**: **0** = FALSE **1** = TRUE
10	RW	0x0	**FAULT_MASK_DSE**: **0** = FALSE **1** = TRUE
9	RW	0x1	**FAULT_EN**: Enable fault reporting from device controller fault control for device system error **0** = FALSE **1** = TRUE
8	RW	0x0	**EVENT_RING_HALT**: Halt posting events, can be used to relocate the event ring data structures **0** = FALSE **1** = TRUE
7	RW	0x0	**EWE**: Enable event for MFINDEX rollover from 3FFF to 0 **0** = FALSE **1** = TRUE
6	RW	0x0	**SMI_DSE**: Enable SMI interrupt for device system errors **0** = FALSE **1** = TRUE
5	RW	0x0	**SMI_EVT**: **0** = FALSE **1** = TRUE
4	RW	0x0	**IE**: Enable legacy/smi interrupt for pending events **0** = FALSE **1** = TRUE
3:2	RO	X	**RSVD1**: **0** = _00
1	RW	0x0	**LSE**: Generate Event on PORTSC change **0** = DIS **1** = EN

Bit	R/W	Reset	Description
0	RW	0x0	**RUN**: Device Mode Run/Stop bit **0** = STOP **1** = RUN

T_XUSB_DEV_XHCI_ST_0

Status Register Run Change - set when run bit transitions from 1 to 0

Interrupt Pending is set by hardware when an event is added to the event ring. Device System Error bit set when error is received on FPCI wrapper.

Offset: 0x34
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xxx0)

Bit	R/W	Reset	Description
31:6	RO	X	**RSVD1**: **0** = _00
5	RW	0x0	**DSE**: **0** = NOTPENDING **1** = PENDING
4	RW	0x0	**IP**: **0** = NOTPENDING **1** = PENDING
3:1	RO	X	**RSVD0**: **0** = _00
0	RW	0x0	**RC**: **0** = NOTPENDING **1** = PENDING

T_XUSB_DEV_XHCI_RT_IMOD_0

Offset: 0x38
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0fa00fa0 (0b0000,1111,1010,0000,0000,1111,1010,0000)

Bit	Reset	Description
31:16	0xfa0	**IMODC**: **4000** = INIT

Bit	Reset	Description
15:0	0xfa0	**IMODI**: **4000** = INIT

T_XUSB_DEV_XHCI_PORTSC_0

Port Status and Control Register

Offset: 0x3c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xX00XX080 (0bx0xx,xxxx,000x,0x00,xx00,0000,1000,0000)

Bit	R/W	Reset	Description
30	RO	0x0	**WPR**: **0** = NORST **1** = RST
29:28	RO	X	**RSVD4**: **0** = _00
23	RW	0x0	**CEC**: **0** = NOT_PENDING **1** = PENDING
22	RW	0x0	**PLC**: **0** = NOT_PENDING **1** = PENDING
21	RW	0x0	**PRC**: **0** = NOT_PENDING **1** = PENDING
19	RW	0x0	**WRC**: **0** = NOT_PENDING **1** = PENDING
18	RO	X	**RSVD3**: **0** = _00
17	RW	0x0	**CSC**: **0** = NOT_PENDING **1** = PENDING
16	RW	0x0	**LWS**: **0** = INIT **1** = SET
14	RO	X	**RSVD2**: **0** = _00

Bit	R/W	Reset	Description
13:10	RO	0x0	**PS**: **0** = UNDEFINED **1** = FS **2** = LS **3** = HS **4** = SS **5** = SSP
9	RO	0x0	**LANE_POLARITY_VALUE**: **0** = INIT
8:5	RW	0x4	**PLS**: **0** = U0 **1** = U1 **2** = U2 **3** = U3 **4** = DISABLED **5** = RXDETECT **6** = INACTIVE **7** = POLLING **8** = RECOVERY **9** = HOTRESET **10** = COMPLIANCE **11** = LOOPBACK **15** = RESUME
4	RO	0x0	**PR**: **0** = NORST **1** = RST
3	RW	0x0	**LANE_POLARITY_OVRD_VALUE**: **0** = INIT
2	RW	0x0	**LANE_POLARITY_OVRD**: **0** = INIT
1	RW	0x0	**PED**: **0** = DIS **1** = EN
0	RW	0x0	**CCS**: **0** = NOCON **1** = CON

T_XUSB_DEV_XHCI_ECPLO_0

Endpoint Context Pointer - points to the default control EP, there is one empty slot after the default control EP and the remaining EP contexts start from offset 2.

Offset: 0x40
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000000XX (0b0000,0000,0000,0000,0000,0000,00xx,xxxx)

Bit	R/W	Reset	Description
31:6	RW	0x0	**ADDRLO**: **0** = INIT
5:0	RO	X	**RSVD0**: **0** = _00

T_XUSB_DEV_XHCI_ECPHI_0

Offset: 0x44
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ADDRHI**: **0** = INIT

T_XUSB_DEV_XHCI_MFINDEX_0

Offset: 0x48
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000XXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
13:3	X	**FRAME**: **0** = INIT
2:0	X	**UFRAME**: **0** = INIT

T_XUSB_DEV_XHCI_PORTPM_0

Port PM Status and Control Register

Offset: 0x4c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0Xffff01 (0b0000,11x0,1111,1111,1111,1111,0000,0x01)

Bit	R/W	Reset	Description
31	RW	0x0	**PNG_CYA**: **0** = INIT
30	RW	0x0	**FRWE**: Enables the port to generate resume signaling. This bit reflects the Function Remote Wake Enable bit captured in the SET_FEATURE device request. This field shall be set to 0x0 by the transition of Port Reset from 0x0 to 0x1. **0** = INIT
29	RW	0x0	**U2E**: Enables the port to initiate U2 entry or accept U2/L1 entry requests. Software sets this bit when U2 is enabled by the SET_FEATURE device request and software clears this bit when U2 is disabled by the CLEAR_FEATURE device request. This field shall be set to 0x0 by the transition of Port Reset from 0x0 to 0x1. **0** = INIT
28	RW	0x0	**U1E**: Enables the port to initiate U1 entry or accept U1 entry requests. Software sets this bit when U1 is enabled by the SET_FEATURE device request and software clears this bit when U1 is disabled by the CLEAR_FEATURE device request. This field shall be set to 0x0 by the transition of Port Reset from 0x0 to 0x1. **0** = INIT
27	RW	0x1	**WOD**: Wake on Disconnection - Writing this bit to 0x1 enables the port to be sensitive to Vbus disconnects as system wake-up events. **1** = INIT
26	RW	0x1	**WOC**: Writing this bit to 0x1 enables the port to be sensitive to Vbus connects as system wake-up events. This bit is not valid for the Debug Capability. The host PORTSC register is used to control the wake on connect for host ports. **1** = INIT
25	RO	X	**VBA**: Indicates whether Vbus is detected at the port. 0x1 indicates Vbus is present and 0x0 indicates Vbus is not detected. **0** = INIT
24	RW	0x0	**FLA**: **0** = INIT
23:16	RW	0xff	**U1TIMEOUT**: **255** = INIT
15:8	RW	0xff	**U2TIMEOUT**: **255** = INIT
7:4	RW	0x0	**HIRD**: **0** = INIT
3	RW	0x0	**RWE**: **0** = DISABLED **1** = ENABLED
1:0	RW	0x1	**L1S**: **0** = DROP **1** = ACCEPT **2** = NYET **3** = STALL

T_XUSB_DEV_XHCI_EP_HALT_0

EP HALT register used by software to program an EP to respond to host request with STALL transaction. When this bit is set, hardware preserves certain EP parameters but software can modify the transfer ring after setting this bit.

Offset: 0x50
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DCI**: **0** = NO **1** = YES

T_XUSB_DEV_XHCI_EP_PAUSE_0

EP Pause register used by software to program the DUT to stop processing the EP and respond host requests with a NRDY/NAK. When this bit is set hardware preserves certain EP parameters, but software can modify the transfer ring after setting this bit.

Offset: 0x54
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DCI**: **0** = NO **1** = YES

T_XUSB_DEV_XHCI_EP_RELOAD_0

EP Reload bits are used by software to direct the Device BI to reload the EP context for that EP and update its internal EP related variables like EP state.

Offset: 0x58
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DCI**: **0** = NO **1** = YES

T_XUSB_DEV_XHCI_EP_STCHG_0

EP STCHG is set by hardware whenever halt/pause/run bit is cleared by software, to indicate completion of processing of the software request. Software should clear the EP_STCHG bit by writing 1 to it, there is 1 bit for each EP.

Offset: 0x5c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**IN15**: **0** = NO **1** = YES
30	0x0	**OUT15**: **0** = NO **1** = YES
29	0x0	**IN14**: **0** = NO **1** = YES
28	0x0	**OUT14**: **0** = NO **1** = YES
27	0x0	**IN13**: **0** = NO **1** = YES
26	0x0	**OUT13**: **0** = NO **1** = YES
25	0x0	**IN12**: **0** = NO **1** = YES
24	0x0	**OUT12**: **0** = NO **1** = YES
23	0x0	**IN11**: **0** = NO **1** = YES

Bit	Reset	Description
22	0x0	**OUT11**: **0** = NO **1** = YES
21	0x0	**IN10**: **0** = NO **1** = YES
20	0x0	**OUT10**: **0** = NO **1** = YES
19	0x0	**IN9**: **0** = NO **1** = YES
18	0x0	**OUT9**: **0** = NO **1** = YES
17	0x0	**IN8**: **0** = NO **1** = YES
16	0x0	**OUT8**: **0** = NO **1** = YES
15	0x0	**IN7**: **0** = NO **1** = YES
14	0x0	**OUT7**: **0** = NO **1** = YES
13	0x0	**IN6**: **0** = NO **1** = YES
12	0x0	**OUT6**: **0** = NO **1** = YES
11	0x0	**IN5**: **0** = NO **1** = YES
10	0x0	**OUT5**: **0** = NO **1** = YES
9	0x0	**IN4**: **0** = NO **1** = YES
8	0x0	**OUT4**: **0** = NO **1** = YES

Bit	Reset	Description
7	0x0	**IN3**: **0** = NO **1** = YES
6	0x0	**OUT3**: **0** = NO **1** = YES
5	0x0	**IN2**: **0** = NO **1** = YES
4	0x0	**OUT2**: **0** = NO **1** = YES
3	0x0	**IN1**: **0** = NO **1** = YES
2	0x0	**OUT1**: **0** = NO **1** = YES
1	0x0	**IN0**: **0** = NO **1** = YES
0	0x0	**OUT0**: **0** = NO **1** = YES

T_XUSB_DEV_XHCI_FLOWCNTRL_0

Flow control threshold values for HSFS NAK. Software needs to enable this if more than four IN or four OUT EPs are configured. IDLE timeout for SS - number of micro frames an async EP is in idle state in a thread before it is evicted automatically.

Offset: 0x60
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXX080000 (0bxxxx,xxxx,0000,1000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:24	RO	X	**RSVD0**: **0** = _00
23:16	RW	0x8	**IDLE_MITS**: **8** = INIT
15	RW	0x0	**OUT_EN**: **0** = INIT

Bit	R/W	Reset	Description
14:8	RW	0x0	**OUT_THRESH**: **0** = INIT
7	RW	0x0	**IN_EN**: **0** = INIT
6:0	RW	0x0	**IN_THRESH**: **0** = INIT

T_XUSB_DEV_XHCI_DEVNOTIF_LO_0

Device notification registers for software to direct hardware to send a device notification TP. Data should be written and updated before writing 1 to the trigger.

Offset: 0x64
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0b0000,0000,0000,0000,0000,0000,0000,xxx0)

Bit	R/W	Reset	Description
31:8	RW	0x0	**DATA**: **0** = INIT
7:4	RW	0x0	**TYPE**: **0** = INIT
3:1	RO	X	**RSVD0**: **0** = _00
0	RW	0x0	**TRIG**: **0** = INIT **1** = SET

T_XUSB_DEV_XHCI_DEVNOTIF_HI_0

Offset: 0x68
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DATA**: **0** = INIT

T_XUSB_DEV_XHCI_PORTHALT_0

Offset: 0x6c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x01000001 (0bxxxx,x001,xxx0,0000,xxxx,xxxx,xxxx,xx01)

Bit	R/W	Reset	Description
26	RW	0x0	**STCHG_SMI_EN**: **0** = INIT
25	RW	0x0	**STCHG_PME_EN**: **0** = INIT
24	RW	0x1	**STCHG_INTR_EN**: **1** = INIT
20	RW	0x0	**STCHG_REQ**: **0** = NOT_PENDING **1** = PENDING
19:16	RO	0x0	**STCHG_STATE**: **0** = U0
1	RW	0x0	**HALT_REJECT**: **0** = FALSE **1** = TRUE
0	RW	0x1	**HALT_LTSSM**: **1** = INIT

T_XUSB_DEV_XHCI_PORT_TM_0

Offset: 0x70
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3:0	0x0	**CTRL**: **0** = DISABLED **1** = TESTJ **2** = TESTK **3** = SE0_NAK **4** = TEST_PKT **5** = TEST_FORCEEN

T_XUSB_DEV_XHCI_EP_THREAD_ACTIVE_0

Offset: 0x74
Read/Write: RO

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DCI**: **0** = NO **1** = YES

T_XUSB_DEV_XHCI_EP_STOPPED_0

Write 1 to Clear bit to indicate that the Endpoint has hit an error condition which needs to be cleared to 0 to stop the endpoint from being processed again.

Offset: 0x78
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**IN15**: **0** = NO **1** = YES
30	0x0	**OUT15**: **0** = NO **1** = YES
29	0x0	**IN14**: **0** = NO **1** = YES
28	0x0	**OUT14**: **0** = NO **1** = YES
27	0x0	**IN13**: **0** = NO **1** = YES
26	0x0	**OUT13**: **0** = NO **1** = YES
25	0x0	**IN12**: **0** = NO **1** = YES
24	0x0	**OUT12**: **0** = NO **1** = YES

Bit	Reset	Description
23	0x0	**IN11**: **0** = NO **1** = YES
22	0x0	**OUT11**: **0** = NO **1** = YES
21	0x0	**IN10**: **0** = NO **1** = YES
20	0x0	**OUT10**: **0** = NO **1** = YES
19	0x0	**IN9**: **0** = NO **1** = YES
18	0x0	**OUT9**: **0** = NO **1** = YES
17	0x0	**IN8**: **0** = NO **1** = YES
16	0x0	**OUT8**: **0** = NO **1** = YES
15	0x0	**IN7**: **0** = NO **1** = YES
14	0x0	**OUT7**: **0** = NO **1** = YES
13	0x0	**IN6**: **0** = NO **1** = YES
12	0x0	**OUT6**: **0** = NO **1** = YES
11	0x0	**IN5**: **0** = NO **1** = YES
10	0x0	**OUT5**: **0** = NO **1** = YES
9	0x0	**IN4**: **0** = NO **1** = YES

Bit	Reset	Description
8	0x0	**OUT4**: **0** = NO **1** = YES
7	0x0	**IN3**: **0** = NO **1** = YES
6	0x0	**OUT3**: **0** = NO **1** = YES
5	0x0	**IN2**: **0** = NO **1** = YES
4	0x0	**OUT2**: **0** = NO **1** = YES
3	0x0	**IN1**: **0** = NO **1** = YES
2	0x0	**OUT1**: **0** = NO **1** = YES
1	0x0	**IN0**: **0** = NO **1** = YES
0	0x0	**OUT0**: **0** = NO **1** = YES

T_XUSB_DEV_XHCI_STREAMID_CFG_0

Offset: 0x7c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXX00 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
31:16	X	**STREAMID**:
12:8	X	**DCI**:
0	0x0	**TRIGGER**: **0** = NOT_PENDING **1** = PENDING **1** = SET

T_XUSB_DEV_XHCI_DEV_SOFTRST_0

Offset: 0x84
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00008000 (0bxxxx,xxxx,xxxx,xxxx,1000,0000,xxxx,xx00)

Bit	Reset	Description
15:8	0x80	**RSTCNT**: **128** = INIT
1	0x0	**CYA_DMAIDLE**: **0** = INIT
0	0x0	**RESET**: **0** = NOT_PENDING **1** = PENDING **1** = SET

T_XUSB_DEV_XHCI_HSFSPI_COUNT0_0

Offset: 0x100
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000012c (0bxx00,0000,0000,0000,0000,0001,0010,1100)

Bit	Reset	Description
29:0	0x12c	**FS_RESET_MIN**: **300** = INIT

T_XUSB_DEV_XHCI_HSFSPI_COUNT1_0

Offset: 0x104
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00003395 (0bxx00,0000,0000,0000,0011,0011,1001,0101)

Bit	Reset	Description
29:0	0x3395	**HS_RESET_MIN**: **13205** = INIT

T_XUSB_DEV_XHCI_HSFSPI_COUNT2_0

Offset: 0x108
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00002eef (0bxx00,0000,0000,0000,0010,1110,1110,1111)

Bit	Reset	Description
29:0	0x2eef	**HS_RESET_ST_RESET_MIN**: **12015** = INIT

T_XUSB_DEV_XHCI_HSFSPI_COUNT3_0

Offset: 0x10c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0003a980 (0bxx00,0000,0000,0011,1010,1001,1000,0000)

Bit	Reset	Description
29:0	0x3a980	**TX_CHIRP_MID**: **240000** = INIT

T_XUSB_DEV_XHCI_HSFSPI_COUNT4_0

Offset: 0x110
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000012c0 (0bxx00,0000,0000,0000,0001,0010,1100,0000)

Bit	Reset	Description
29:0	0x12c0	**CHIRP_MIN**: **4800** = INIT

T_XUSB_DEV_XHCI_HSFSPI_COUNT5_0

Offset: 0x114
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00001c20 (0bxx00,0000,0000,0000,0001,1100,0010,0000)

Bit	Reset	Description
29:0	0x1c20	**CHIRP_MAX**: **7200** = INIT

T_XUSB_DEV_XHCI_HSFSPI_COUNT6_0

Offset: 0x118
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00057ed0 (0bxx00,0000,0000,0101,0111,1110,1101,0000)

Bit	Reset	Description
29:0	0x57ed0	**INACTIVITY_TIMEOUT**: **360144** = INIT

T_XUSB_DEV_XHCI_HSFSPI_COUNT7_0

Offset: 0x11c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000000e0 (0bxxxx,xxxx,xx00,0000,0000,0000,1110,0000)

Bit	Reset	Description
21:0	0xe0	**HS_FSM_TIMEOUT**: **224** = INIT

T_XUSB_DEV_XHCI_HSFSPI_COUNT8_0

Offset: 0x120
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000000b0 (0bxxxx,xxxx,xx00,0000,0000,0000,1011,0000)

Bit	Reset	Description
21:0	0xb0	**FS_FSM_TIMEOUT**: **176** = INIT

T_XUSB_DEV_XHCI_HSFSPI_COUNT9_0

Offset: 0x124
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00124f80 (0bxx00,0000,0001,0010,0100,1111,1000,0000)

Bit	Reset	Description
29:0	0x124f80	**U3_RESUME_K_DURATION**: **1200000** = INIT

T_XUSB_DEV_XHCI_HSFSPI_COUNT10_0

Offset: 0x128
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00003a9e (0bxx00,0000,0000,0000,0011,1010,1001,1110)

Bit	Reset	Description
29:0	0x3a9e	**U3_ENTRY_DELAY**: **15006** = INIT

T_XUSB_DEV_XHCI_HSFSPI_COUNT11_0

Offset: 0x12c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0001d4ff (0bxx00,0000,0000,0001,1101,0100,1111,1111)

Bit	Reset	Description
29:0	0x1d4ff	**FS_RESET_ST_RESET_MIN**: **120063** = INIT

T_XUSB_DEV_XHCI_HSFSPI_COUNT12_0

Offset: 0x130
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000003d4 (0bxx00,0000,0000,0000,0000,0011,1101,0100)

Bit	Reset	Description
29:0	0x3d4	**U2_ENTRY_DELAY**: **980** = INIT

T_XUSB_DEV_XHCI_HSFSPI_COUNT13_0

Offset: 0x134
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00001772 (0bxx00,0000,0000,0000,0001,0111,0111,0010)
PROD: 0x00002c88 (0bxx00,0000,0000,0000,0010,1100,1000,1000)

Bit	Reset	Description
29:0	0x1772	**U2_RESUME_K_DURATION**: **6002** = INIT

T_XUSB_DEV_XHCI_HSFSPI_CTRL_0

Offset: 0x138
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x7cc00c60 (0bx111,1100,1100,0000,0000,1100,0110,0000)

Bit	Reset	Description
30:26	0x1f	**HS_IDLE_BIT_TIME**: **31** = INIT
25:22	0x3	**FS_SE0_WIDTH**: **3** = INIT
21	0x0	**BITSTUFF_DISABLE**: **0** = INIT
20	0x0	**NRZI_DISABLE**: **0** = INIT
19:10	0x3	**FS_INTERPKT_DELAY**: **3** = INIT
9:0	0x60	**HS_INTERPKT_DELAY**: **96** = INIT

T_XUSB_DEV_XHCI_HSFSPI_TESTMODE_CTRL_0

Offset: 0x138
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x7cc00c60 (0bx111,1100,1100,0000,0000,1100,0110,0000)

Bit	Reset	Description
1	0x0	**CYA_ADDR_MATCH**: **0** = INIT
0	0x0	**PATTERN_SELECT**: **0** = INIT

T_XUSB_DEV_XHCI_HSFSPI_TESTMODE_PATTERN0_0

Offset: 0x140
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**BYTE3**: **0** = INIT
23:16	0x0	**BYTE2**: **0** = INIT
15:8	0x0	**BYTE1**: **0** = INIT
7:0	0x0	**BYTE0**: **0** = INIT

T_XUSB_DEV_XHCI_HSFSPI_TESTMODE_PATTERN1_0

Offset: 0x144
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**BYTE3**: **0** = INIT

Bit	Reset	Description
23:16	0x0	**BYTE2**: **0** = INIT
15:8	0x0	**BYTE1**: **0** = INIT
7:0	0x0	**BYTE0**: **0** = INIT

T_XUSB_DEV_XHCI_HSFSPI_TESTMODE_PATTERN2_0

Offset: 0x148
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xaaaaaa00 (0b1010,1010,1010,1010,1010,1010,0000,0000)

Bit	Reset	Description
31:24	0xaa	**BYTE3**: **170** = INIT
23:16	0xaa	**BYTE2**: **170** = INIT
15:8	0xaa	**BYTE1**: **170** = INIT
7:0	0x0	**BYTE0**: **0** = INIT

T_XUSB_DEV_XHCI_HSFSPI_TESTMODE_PATTERN3_0

Offset: 0x14c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xaaaaaaaa (0b1010,1010,1010,1010,1010,1010,1010,1010)

Bit	Reset	Description
31:24	0xaa	**BYTE3**: **170** = INIT
23:16	0xaa	**BYTE2**: **170** = INIT
15:8	0xaa	**BYTE1**: **170** = INIT

Bit	Reset	Description
7:0	0xaa	**BYTE0**: **170** = INIT

T_XUSB_DEV_XHCI_HSFSPI_TESTMODE_PATTERN4_0

Offset: 0x150
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xeeeeeeaa (0b1110,1110,1110,1110,1110,1110,1010,1010)

Bit	Reset	Description
31:24	0xee	**BYTE3**: **238** = INIT
23:16	0xee	**BYTE2**: **238** = INIT
15:8	0xee	**BYTE1**: **238** = INIT
7:0	0xaa	**BYTE0**: **170** = INIT

T_XUSB_DEV_XHCI_HSFSPI_TESTMODE_PATTERN5_0

Offset: 0x154
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xeeeeeeee (0b1110,1110,1110,1110,1110,1110,1110,1110)

Bit	Reset	Description
31:24	0xee	**BYTE3**: **238** = INIT
23:16	0xee	**BYTE2**: **238** = INIT
15:8	0xee	**BYTE1**: **238** = INIT
7:0	0xee	**BYTE0**: **238** = INIT

T_XUSB_DEV_XHCI_HSFSPI_TESTMODE_PATTERN6_0

Offset: 0x158
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xfffffeee (0b1111,1111,1111,1111,1111,1110,1110,1110)

Bit	Reset	Description
31:24	0xff	**BYTE3**: **255** = INIT
23:16	0xff	**BYTE2**: **255** = INIT
15:8	0xfe	**BYTE1**: **254** = INIT
7:0	0xee	**BYTE0**: **238** = INIT

T_XUSB_DEV_XHCI_HSFSPI_TESTMODE_PATTERN7_0

Offset: 0x15c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:24	0xff	**BYTE3**: **255** = INIT
23:16	0xff	**BYTE2**: **255** = INIT
15:8	0xff	**BYTE1**: **255** = INIT
7:0	0xff	**BYTE0**: **255** = INIT

T_XUSB_DEV_XHCI_HSFSPI_TESTMODE_PATTERN8_0

Offset: 0x160
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:24	0xff	**BYTE3**: **255** = INIT
23:16	0xff	**BYTE2**: **255** = INIT
15:8	0xff	**BYTE1**: **255** = INIT
7:0	0xff	**BYTE0**: **255** = INIT

T_XUSB_DEV_XHCI_HSFSPI_TESTMODE_PATTERN9_0

Offset: 0x164
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xdfbf7fff (0b1101,1111,1011,1111,0111,1111,1111,1111)

Bit	Reset	Description
31:24	0xdf	**BYTE3**: **223** = INIT
23:16	0xbf	**BYTE2**: **191** = INIT
15:8	0x7f	**BYTE1**: **127** = INIT
7:0	0xff	**BYTE0**: **255** = INIT

T_XUSB_DEV_XHCI_HSFSPI_TESTMODE_PATTERN10_0

Offset: 0x168
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xfdfbf7ef (0b1111,1101,1111,1011,1111,0111,1110,1111)

Bit	Reset	Description
31:24	0xfd	**BYTE3**: **253** = INIT
23:16	0xfb	**BYTE2**: **251** = INIT

Bit	Reset	Description
15:8	0xf7	**BYTE1**: **247** = INIT
7:0	0xef	**BYTE0**: **239** = INIT

T_XUSB_DEV_XHCI_HSFSPI_TESTMODE_PATTERN11_0

Offset: 0x16c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xdfbf7efc (0b1101,1111,1011,1111,0111,1110,1111,1100)

Bit	Reset	Description
31:24	0xdf	**BYTE3**: **223** = INIT
23:16	0xbf	**BYTE2**: **191** = INIT
15:8	0x7e	**BYTE1**: **126** = INIT
7:0	0xfc	**BYTE0**: **252** = INIT

T_XUSB_DEV_XHCI_HSFSPI_TESTMODE_PATTERN12_0

Offset: 0x170
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xfdfbf7ef (0b1111,1101,1111,1011,1111,0111,1110,1111)

Bit	Reset	Description
31:24	0xfd	**BYTE3**: **253** = INIT
23:16	0xfb	**BYTE2**: **251** = INIT
15:8	0xf7	**BYTE1**: **247** = INIT
7:0	0xef	**BYTE0**: **239** = INIT

T_XUSB_DEV_XHCI_HSFSPI_TESTMODE_PATTERN13_0

Offset: 0x174
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x7f7f7f7e (0b0111,1111,0111,1111,0111,1111,0111,1110)

Bit	Reset	Description
31:24	0x7f	**BYTE3**: **127** = INIT
23:16	0x7f	**BYTE2**: **127** = INIT
15:8	0x7f	**BYTE1**: **127** = INIT
7:0	0x7e	**BYTE0**: **126** = INIT

T_XUSB_DEV_XHCI_HSFSPI_TESTMODE_PATTERN14_0

Offset: 0x178
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**BYTE3**: **0** = INIT
23:16	0x0	**BYTE2**: **0** = INIT
15:8	0x0	**BYTE1**: **0** = INIT
7:0	0x0	**BYTE0**: **0** = INIT

T_XUSB_DEV_XHCI_HSFSPI_TESTMODE_PATTERN15_0

Offset: 0x17c
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**BYTE3**: **0** = INIT
23:16	0x0	**BYTE2**: **0** = INIT
15:8	0x0	**BYTE1**: **0** = INIT
7:0	0x0	**BYTE0**: **0** = INIT

T_XUSB_DEV_XHCI_HSFSPI_PVTPORTDBG_CTRL_0

Register for helping in debug of PSM state transitions

Offset: 0x180
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,xxxx,xx00)

Bit	Reset	Description
23:20	0x0	**END_MINOR**: **0** = INIT
19:16	0x0	**END_MAJOR**: **0** = INIT
15:12	0x0	**START_MINOR**: **0** = INIT
11:8	0x0	**START_MAJOR**: **0** = INIT
1	0x0	**CLEAR**: **0** = NOT_PENDING **1** = PENDING **1** = TRIGGER
0	0x0	**RUN**: **0** = STOP **1** = RUN

T_XUSB_DEV_XHCI_HSFSPI_PVTPORTDBG_STS_0

Offset: 0x184
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
19	X	**TESTMODE_HS_PKT**:
18	X	**TESTMODE_HS_NAK**:
17	X	**TESTMODE_HS_KSTATE**:
16	X	**TESTMODE_HS_JSTATE**:
15	X	**ENABLED_FS_RESUME**:
14	X	**ENABLED_FS_U3**:
13	X	**ENABLED_FS_U2**:
12	X	**ENABLED_FS_U0**:
11	X	**ENABLED_HS_RESUME**:
10	X	**ENABLED_HS_U3**:
9	X	**ENABLED_HS_U2**:
8	X	**ENABLED_HS_U0**:
7	X	**RESET_HS_PROLOG**:
6	X	**RESET_FS**:
5	X	**RESET_HS_RXCHIRPJ**:
4	X	**RESET_HS_RXCHIRPK**:
3	X	**RESET_HS_TXCHIRPK**:
2	X	**DISABLED_NONE**:
1	X	**DISCONNECTED_NONE**:
0	X	**START_NONE**:

T_XUSB_DEV_XHCI_HSFSPI_COUNT14_0

Offset: 0x18c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000040 (0bxxxx,xxxx,xx00,0000,0000,0000,0100,0000)

Bit	Reset	Description
21:0	0x40	**HS_ISO_FSM_TIMEOUT**: **64** = INIT

T_XUSB_DEV_XHCI_HSFSPI_COUNT15_0

Offset: 0x190
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000000b0 (0bxxxx,xxxx,xx00,0000,0000,0000,1011,0000)

Bit	Reset	Description
21:0	0xb0	**FS_ISO_FSM_TIMEOUT**: **176** = INIT

T_XUSB_DEV_XHCI_HSFSPI_NVWRAP_DESER_0

Offset: 0x194
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x01568c03 (0bxxxx,xxx1,0101,0110,1000,1100,0000,0011)

Bit	Reset	Description
24	0x1	**USECOMMONUPDNCNT**: **1** = INIT
23	0x0	**THREECYCLEEDGEDETCFG**: **0** = INIT
22	0x1	**USEWEIGHTEDEDGESCFG**: **1** = INIT
21	0x0	**HSRXLATESQUELCHCFG**: **0** = INIT
20:19	0x2	**KEEPPATTERNONACTIVECFG**: **2** = INIT
18	0x1	**ALLOWCONSECUTIVEUPDOWNCFG**: **1** = INIT
17	0x1	**REALIGNONNEWPACKETCFG**: **1** = INIT
16:13	0x4	**PCOUNTUPDOWNDIVCFG**: **4** = INIT

Bit	Reset	Description
12:10	0x3	**SQUELCHEOPDELAYCFG**: **3** = INIT
9	0x0	**PASSCHIRPCFG**: **0** = INIT
8	0x0	**PASSFEEDBACKCFG**: **0** = INIT
7:6	0x0	**PCOUNTINERTIACFG**: **0** = INIT
5:4	0x0	**PHASEADJUSTCFG**: **0** = INIT
3	0x0	**THREESYNCBITSCFG**: **0** = INIT
2	0x0	**USE4SYNCTRANCFG**: **0** = INIT
1	0x1	**T210_A02_MODE**: **1** = INIT
0	0x1	**MCP_MODE**: **1** = INIT

T_XUSB_DEV_XHCI_HSFSPI_CTRL1_0

Offset: 0x198
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000b1710 (0bxxxx,xxxx,xxx0,1011,xx01,0111,0001,0000)

Bit	Reset	Description
16	0x1	**FILTERED_BUS_IDLE**: **1** = INIT
13:9	0x9	**FS_IDLE_BITTIMES**: **9** = INIT
8	0x1	**BABBLE_TIMEOUT_CYA**: **1** = INIT
7:0	0x10	**BABBLE_TIMEOUT_VAL**: **16** = INIT

T_XUSB_DEV_XHCI_HSFSPI_COUNT16_0

Offset: 0x19c
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x000927c0 (0bxx00,0000,0000,1001,0010,0111,1100,0000)

Bit	Reset	Description
29:0	0x927c0	**CHIRP_FAIL**: **600000** = INIT

T_XUSB_DEV_XHCI_SSPI_HOSTCFG_START_0

Offset: 0x600
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00001000 (0b0000,0000,0000,0000,0001,0000,0000,0000)

Bit	Reset	Description
31:0	0x1000	**FIELD**: **4096** = VALUE

T_XUSB_DEV_XHCI_SSPI_HOSTCFG_END_0

Offset: 0x7fc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**FIELD**: **0** = VALUE

T_XUSB_DEV_XHCI_PERFMON_READ_CUMLATENCY_REG0_0

Offset: 0x800
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: **0** = INIT

Bit	Reset	Description
30:16	0x0	**PKTS**: **0** = INIT
15:0	0x0	**CYCLESHI**: **0** = INIT

T_XUSB_DEV_XHCI_PERFMON_READ_CUMLATENCY_REG1_0

Offset: 0x804
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CYCLESLO**: **0** = INIT

T_XUSB_DEV_XHCI_PERFMON_READ_LATENCY_0

Offset: 0x808
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000014 (0b0000,0000,0000,0000,0000,0000,0001,0100)

Bit	Reset	Description
31:16	0x0	**MINCYCLES**: **0** = INIT
15:0	0x14	**MAXCYCLES**: **20** = INIT

T_XUSB_DEV_XHCI_PERFMON_READ_HISTOGRAM_BOUNDARY_REG0_0

Offset: 0x80c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000014 (0b0000,0000,0000,0000,0000,0000,0001,0100)

Bit	Reset	Description
31:16	0x0	**B**: **0** = INIT
15:0	0x14	**A**: **20** = INIT

T_XUSB_DEV_XHCI_PERFMON_READ_HISTOGRAM_BOUNDARY_REG1_0

Offset: 0x810
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000014 (0b0000,0000,0000,0000,0000,0000,0001,0100)

Bit	Reset	Description
31:16	0x0	**D**: **0** = INIT
15:0	0x14	**C**: **20** = INIT

T_XUSB_DEV_XHCI_PERFMON_READ_HISTOGRAM_BUCKET_REG0_0

Offset: 0x814
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000014 (0b0000,0000,0000,0000,0000,0000,0001,0100)

Bit	Reset	Description
31:16	0x0	**_1**: **0** = INIT
15:0	0x14	**_0**: **20** = INIT

T_XUSB_DEV_XHCI_PERFMON_READ_HISTOGRAM_BUCKET_REG1_0

Offset: 0x818
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000014 (0b0000,0000,0000,0000,0000,0000,0001,0100)

Bit	Reset	Description
31:16	0x0	**_3**: **0** = INIT
15:0	0x14	**_2**: **20** = INIT

T_XUSB_DEV_XHCI_PERFMON_READ_HISTOGRAM_BUCKET_REG2_0

Offset: 0x81c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000014 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0001,0100)

Bit	Reset	Description
15:0	0x14	**_4**: **20** = INIT

T_XUSB_DEV_XHCI_PERFMON_BI_CTRL_0

Offset: 0x820
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**EN**: **0** = INIT

T_XUSB_DEV_XHCI_PERFMON_BI_EPTRB_0

Offset: 0x824
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,0000,0000,xxxx,xxxx)

Bit	Reset	Description
24:16	0x0	**TRB**: **0** = INIT

Bit	Reset	Description
15:8	0x0	**EPSYSMEM**: **0** = INIT

T_XUSB_DEV_XHCI_PERFMON_BI_DATA_OUT_0

Offset: 0x828
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**REQCOUNT**: **0** = INIT
23:0	0x0	**SIZE**: **0** = INIT

T_XUSB_DEV_XHCI_PERFMON_BI_DATA_IN_0

Offset: 0x82c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**REQCOUNT**: **0** = INIT
23:0	0x0	**SIZE**: **0** = INIT

T_XUSB_DEV_XHCI_BLCG_0

Clock gating control status register

Offset: 0x840
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00007ffd (0bx000,0000,0000,00x0,x111,1111,1111,11x1)
PROD: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,xxxx,xxxx)

Bit	Reset	Description
30	0x0	**OVRD_SS_PI_500M**: **0** = DISABLED **1** = ENABLED
29	0x0	**OVRD_SS_PI**: **0** = DISABLED **1** = ENABLED
28	0x0	**OVRD_IOPLL_3_PWRDN**: **0** = DISABLED **1** = ENABLED
27	0x0	**OVRD_IOPLL_2_PWRDN**: **0** = DISABLED **1** = ENABLED
26	0x0	**OVRD_IOPLL_1_PWRDN**: **0** = DISABLED **1** = ENABLED
25	0x0	**OVRD_IOPLL_0_PWRDN**: **0** = DISABLED **1** = ENABLED
24	0x0	**OVRD_COREPLL_PWRDN**: **0** = DISABLED **1** = ENABLED
23	0x0	**OVRD_NVWRAP_48M**: **0** = DISABLED **1** = ENABLED
22	0x0	**OVRD_NVWRAP_480M**: **0** = DISABLED **1** = ENABLED
21	0x0	**OVRD_HSFS_PI**: **0** = DISABLED **1** = ENABLED
20	0x0	**OVRD_PICLK_BI**: **0** = DISABLED **1** = ENABLED
19	0x0	**OVRD_CORE_BI**: **0** = DISABLED **1** = ENABLED
18	0x0	**OVRD_FE**: **0** = DISABLED **1** = ENABLED
16	0x0	**OVRD_AXIAPB**: **0** = DISABLED **1** = ENABLED
14	0x1	**SS_PI_500M**: **0** = DISABLED **1** = ENABLED

Bit	Reset	Description
13	0x1	**SS_PI**: **0** = DISABLED **1** = ENABLED
12	0x1	**IOPLL_3_PWRDN**: **0** = DISABLED **1** = ENABLED
11	0x1	**IOPLL_2_PWRDN**: **0** = DISABLED **1** = ENABLED
10	0x1	**IOPLL_1_PWRDN**: **0** = DISABLED **1** = ENABLED
9	0x1	**IOPLL_0_PWRDN**: **0** = DISABLED **1** = ENABLED
8	0x1	**COREPLL_PWRDN**: **0** = DISABLED **1** = ENABLED
7	0x1	**NVWRAP_48M**: **0** = DISABLED **1** = ENABLED
6	0x1	**NVWRAP_480M**: **0** = DISABLED **1** = ENABLED
5	0x1	**HSFS_PI**: **0** = DISABLED **1** = ENABLED
4	0x1	**PICLK_BI**: **0** = DISABLED **1** = ENABLED
3	0x1	**CORE_BI**: **0** = DISABLED **1** = ENABLED
2	0x1	**FE**: **0** = DISABLED **1** = ENABLED
0	0x1	**AXIAPB**: **0** = DISABLED **1** = ENABLED

T_XUSB_DEV_XHCI_BLCG_STS_0

Offset: 0x844
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0xxx)

Bit	Reset	Description
14	0x0	**SS_PI_500M**: **0** = DISABLED **1** = ENABLED
13	0x0	**SS_PI**: **0** = DISABLED **1** = ENABLED
12	0x0	**IOPLL_3_PWRDN**: **0** = DISABLED **1** = ENABLED
11	0x0	**IOPLL_2_PWRDN**: **0** = DISABLED **1** = ENABLED
10	0x0	**IOPLL_1_PWRDN**: **0** = DISABLED **1** = ENABLED
9	0x0	**IOPLL_0_PWRDN**: **0** = DISABLED **1** = ENABLED
8	0x0	**COREPLL_PWRDN**: **0** = DISABLED **1** = ENABLED
7	0x0	**NVWRAP_48M**: **0** = DISABLED **1** = ENABLED
6	0x0	**NVWRAP_480M**: **0** = DISABLED **1** = ENABLED
5	0x0	**HSFS_PI**: **0** = DISABLED **1** = ENABLED
4	0x0	**PICLK_BI**: **0** = DISABLED **1** = ENABLED
3	0x0	**CORE_BI**: **0** = DISABLED **1** = ENABLED

T_XUSB_DEV_XHCI_BLCG_INTR_0

Offset: 0x848
Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0bxxx0,0000,xxxx,xxxx,xxx0,0000,xxxx,xxx0)

Bit	R/W	Reset	Description
28	RO	0x0	**STS_IOPLL_3_PWRDN**: **0** = DISABLED **1** = ENABLED
27	RO	0x0	**STS_IOPLL_2_PWRDN**: **0** = DISABLED **1** = ENABLED
26	RO	0x0	**STS_IOPLL_1_PWRDN**: **0** = DISABLED **1** = ENABLED
25	RO	0x0	**STS_IOPLL_0_PWRDN**: **0** = DISABLED **1** = ENABLED
24	RO	0x0	**STS_COREPLL_PWRDN**: **0** = DISABLED **1** = ENABLED
12	RW	0x0	**IOPLL_3_PWRDN**: **0** = DISABLED **1** = ENABLED
11	RW	0x0	**IOPLL_2_PWRDN**: **0** = DISABLED **1** = ENABLED
10	RW	0x0	**IOPLL_1_PWRDN**: **0** = DISABLED **1** = ENABLED
9	RW	0x0	**IOPLL_0_PWRDN**: **0** = DISABLED **1** = ENABLED
8	RW	0x0	**COREPLL_PWRDN**: **0** = DISABLED **1** = ENABLED
0	RW	0x0	**TGT**: **0** = SMI **1** = PME

T_XUSB_DEV_XHCI_CFG_DEVBI_0

Offset: 0x850
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x3026b8c3 (0bx011,0000,0010,0110,1011,1xx0,1100,0011)

Bit	Reset	Description
30	0x0	**OVERRUN_EVT_DEQPTR0**: **0** = INIT
29	0x1	**ISOCH_SKIP_SIA**: If there is no host request for an EP which has TD pending with SIA=1, BI will generate a missed service if this configuration is 1. **1** = INIT
28:24	0x10	**DMA_RD_MAX_ALOM**: Controls the number of back to back accesses that DMA Engines issue with ALOM bit (At-Least One More) set **16** = INIT
23:16	0x26	**CNT_250NS**: 250NS counter duration for Interrupt Moderation **38** = INIT
14	0x0	**TRBFETCH_RDPASSPW**: **0** = INIT
13	0x1	**ASYNC_EP_IDLE**: Check to include Async EPs Idle condition for U1/U2. If this bit is 1, Async EPs Idle is based on PP bit indication by host. If it is 0, Async EPs always flags idle. **1** = INIT
12	0x1	**LOCAL_ROTATE**: Enabled Performance mode in Thread logic where it does not evict and Endpoint if there are no other endpoints waiting **0** = DIS **1** = EN **1** = INIT
11	0x1	**TRBFETCH_RINGEND_CHK**: Setting this to 1, makes TRB Fetch always fetch one extra TRB to check for End of Ring. **1** = EN
8	0x0	**TRBFETCH_IDT_IN**: Check to allow IDT TRBs for an OUT Endpoint **0** = INIT
7	0x1	**DMA_WR_UPSTREAM_RO**: Controls PCIE Attributes for upstream requests **1** = INIT
6	0x1	**DMA_RD_UPSTREAM_RO**: Controls PCIE Attributes for upstream requests **1** = INIT
5	0x0	**DMA_RD_UPSTREAM_RDPASSPW**: **0** = INIT
4:0	0x3	**DMA_WR_MAX_ALOM**: **3** = INIT

T_XUSB_DEV_XHCI_CFG_DEVBI_UPSTREAM_0

Offset: 0x854
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,xxx0,0000,xxxx,x000)

Bit	Reset	Description
31:24	0x0	**DMA_WR_LIMIT**: **0** = INIT
23:16	0x0	**DMA_RD_LIMIT**: **0** = INIT
12	0x0	**DMA_WR_USE_WR_ACKS**: **0** = INIT
11	0x0	**EPLOGIC_RDPASSPW**: **0** = INIT
10	0x0	**EPLOGIC_RO**: **0** = INIT
9	0x0	**EPLOGIC_NS**: **0** = INIT
8	0x0	**EPLOGIC_TC**: **0** = INIT
2	0x0	**EVENTQ_RO**: **0** = INIT
1	0x0	**EVENTQ_NS**: **0** = INIT
0	0x0	**EVENTQ_TC**: **0** = INIT

T_XUSB_DEV_XHCI_CFG_DEV_SSPI_XFER_0

Offset: 0x858
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000f00 (0b0000,0000,0000,0000,0000,1111,0000,0000)
PROD: 0x0000f000 (0b0000,0000,0000,0000,1111,0000,0000,0000)

Bit	Reset	Description
31:0	0xf00	**ACKTIMEOUT**: **3840** = INIT

T_XUSB_DEV_XHCI_CFG_DEV_FE_0

Offset: 0x85c
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x1a000030 (0bxx01,1010,xxxx,xxxx,xxxx,xxxx,xx11,0000)

Bit	Reset	Description
29	0x0	**INFINITE_SS_RETRY**: **0** = DIS **1** = EN
28	0x1	**EN_PRIME_EVENT**: **1** = INIT
27	0x1	**FEATURE_LPM**: **0** = DIS **1** = EN
26	0x0	**EN_STALL_EVENT**: **0** = INIT
25	0x1	**PORTDISCON_RST_HW**: **1** = INIT
24	0x0	**CTX_RESTORE**: **0** = INIT
5	0x1	**PORTRST_HW**: **1** = INIT
4	0x1	**SEQNUM_INIT**: **1** = INIT
3:0	0x0	**PORTREGSEL**: **0** = INIT **1** = SS_AUTO **2** = HSFS **4** = SSPI_0 **5** = SSPI_1 **6** = SSPI_2 **7** = SSPI_3

T_XUSB_DEV_XHCI_CFG_IDLE_0

Offset: 0x860
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
4	X	**DEV_SS_PI**:
3	X	**DEV_FS_NVWRAP**:
2	X	**DEV_HS_NVWRAP**:
1	X	**DEV_HSFS_PI**:

Bit	Reset	Description
0	X	**DEV_BI**:

T_XUSB_DEV_XHCI_CFG_SSXFER_0

Offset: 0x864
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x03938700 (0b0000,0011,1001,0011,1000,0111,0000,0000)

Bit	Reset	Description
31:0	0x3938700	**ERDYTIMER**: **60000000** = INIT

T_XUSB_DEV_XHCI_CFG_SSXFER1_0

Offset: 0x868
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000002 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0010)

Bit	Reset	Description
15:0	0x2	**PINGTIMER**: **2** = INIT

T_XUSB_DEV_XHCI_CFG_DEVBI1_0

Offset: 0x86c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0001)

Bit	Reset	Description
7:0	0x1	**TRBFETCH_NUMSKIP**: **1** = INIT

T_XUSB_DEV_XHCI_CFG_SPARE0_0

Offset: 0x870
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**REG**: **0** = INIT

T_XUSB_DEV_XHCI_CFG_SPARE1_0

Offset: 0x874
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffffffff	**REG**: **4294967295** = INIT

T_XUSB_DEV_XHCI_CFG_DEVBI_DMA_CTRL_0

Offset: 0x878
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
18:16	0x0	**WR_UPSTREAM_NUM_REQ**: **0** = INIT
2:0	0x0	**RD_UPSTREAM_NUM_REQ**: **0** = INIT

T_XUSB_DEV_XHCI_CFG_DEV_MFINDEX_0

Offset: 0x87c
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x000004b0 (0bxxxx,xxxx,xxxx,0000,0000,0100,1011,0000)

Bit	Reset	Description
19:0	0x4b0	**MFCOUNT_MIN**: **1200** = INIT

9.2.4.2 USB PADCTL and AO Registers

Refer to "Reading Register Tables" in the Introduction chapter for the register table protocol as well as recommendations for accessing registers.

The Base Addresses of the registers related to USB are specified in the Address Map section of the Orin TRM.

9.2.4.2.1 XUSB PADCTL Registers

XUSB_PADCTL_BOOT_MEDIA_0
Offset: 0x0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4:1	0x0	**BOOT_PORT**: **0** = USB2_OTG0 **1** = USB2_OTG1 **2** = USB2_OTG2 **3** = USB2_OTG3 **4** = USB2_OTG4 **5** = USB2_OTG5 **6** = USB2_OTG6
0	0x0	**BOOT_MEDIA_ENABLE**: **0** = NO **1** = YES

XUSB_PADCTL_USB2_PAD_MUX_0

Offset: 0x4
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000055 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0101,0101)

Bit	Reset	Description
7:6	0x1	**USB2_OTG_PAD_PORT3**: **1** = XUSB **2** = SWD **3** = UART
5:4	0x1	**USB2_OTG_PAD_PORT2**: **1** = XUSB **2** = SWD **3** = UART
3:2	0x1	**USB2_OTG_PAD_PORT1**: **1** = XUSB **2** = SWD **3** = UART
1:0	0x1	**USB2_OTG_PAD_PORT0**: **1** = XUSB **2** = SWD **3** = UART

XUSB_PADCTL_USB2_PORT_CAP_0

Offset: 0x8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15	0x0	**PORT3_REVERSE_ID**: **0** = NO **1** = YES
14	0x0	**PORT3_INTERNAL**: **0** = NO **1** = YES
13:12	0x0	**PORT3_CAP**: **0** = DISABLED **1** = HOST_ONLY **2** = DEVICE_ONLY **3** = OTG_CAP
11	0x0	**PORT2_REVERSE_ID**: **0** = NO **1** = YES
10	0x0	**PORT2_INTERNAL**: **0** = NO **1** = YES

Bit	Reset	Description
9:8	0x0	**PORT2_CAP**: **0** = DISABLED **1** = HOST_ONLY **2** = DEVICE_ONLY **3** = OTG_CAP
7	0x0	**PORT1_REVERSE_ID**: **0** = NO **1** = YES
6	0x0	**PORT1_INTERNAL**: **0** = NO **1** = YES
5:4	0x0	**PORT1_CAP**: **0** = DISABLED **1** = HOST_ONLY **2** = DEVICE_ONLY **3** = OTG_CAP
3	0x0	**PORT0_REVERSE_ID**: **0** = NO **1** = YES
2	0x0	**PORT0_INTERNAL**: **0** = NO **1** = YES
1:0	0x0	**PORT0_CAP**: **0** = DISABLED **1** = HOST_ONLY **2** = DEVICE_ONLY **3** = OTG_CAP

XUSB_PADCTL_SS_PORT_CAP_0

Offset: 0xc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15	0x0	**PORT3_REVERSE_ID**: **0** = NO **1** = YES
14	0x0	**PORT3_INTERNAL**: **0** = NO **1** = YES

Bit	Reset	Description
13:12	0x0	**PORT3_CAP**: **0** = DISABLED **1** = HOST_ONLY **2** = DEVICE_ONLY **3** = OTG_CAP
11	0x0	**PORT2_REVERSE_ID**: **0** = NO **1** = YES
10	0x0	**PORT2_INTERNAL**: **0** = NO **1** = YES
9:8	0x0	**PORT2_CAP**: **0** = DISABLED **1** = HOST_ONLY **2** = DEVICE_ONLY **3** = OTG_CAP
7	0x0	**PORT1_REVERSE_ID**: **0** = NO **1** = YES
6	0x0	**PORT1_INTERNAL**: **0** = NO **1** = YES
5:4	0x0	**PORT1_CAP**: **0** = DISABLED **1** = HOST_ONLY **2** = DEVICE_ONLY **3** = OTG_CAP
3	0x0	**PORT0_REVERSE_ID**: **0** = NO **1** = YES
2	0x0	**PORT0_INTERNAL**: **0** = NO **1** = YES
1:0	0x0	**PORT0_CAP**: **0** = DISABLED **1** = HOST_ONLY **2** = DEVICE_ONLY **3** = OTG_CAP

XUSB_PADCTL_USB2_OC_MAP_0

Offset: 0x10
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000ffff (0bxxxx,xxxx,xxxx,xxxx,1111,1111,1111,1111)

Bit	Reset	Description
15:12	0xf	**PORT3_OC_PIN**: **0** = OC_DETECTED0 **1** = OC_DETECTED1 **2** = OC_DETECTED2 **3** = OC_DETECTED3 **4** = OC_DETECTED_VBUS_PAD0 **5** = OC_DETECTED_VBUS_PAD1 **6** = OC_DETECTED_VBUS_PAD2 **7** = OC_DETECTED_VBUS_PAD3 **15** = OC_DETECTION_DISABLED
11:8	0xf	**PORT2_OC_PIN**: **0** = OC_DETECTED0 **1** = OC_DETECTED1 **2** = OC_DETECTED2 **3** = OC_DETECTED3 **4** = OC_DETECTED_VBUS_PAD0 **5** = OC_DETECTED_VBUS_PAD1 **6** = OC_DETECTED_VBUS_PAD2 **7** = OC_DETECTED_VBUS_PAD3 **15** = OC_DETECTION_DISABLED
7:4	0xf	**PORT1_OC_PIN**: **0** = OC_DETECTED0 **1** = OC_DETECTED1 **2** = OC_DETECTED2 **3** = OC_DETECTED3 **4** = OC_DETECTED_VBUS_PAD0 **5** = OC_DETECTED_VBUS_PAD1 **6** = OC_DETECTED_VBUS_PAD2 **7** = OC_DETECTED_VBUS_PAD3 **15** = OC_DETECTION_DISABLED
3:0	0xf	**PORT0_OC_PIN**: **0** = OC_DETECTED0 **1** = OC_DETECTED1 **2** = OC_DETECTED2 **3** = OC_DETECTED3 **4** = OC_DETECTED_VBUS_PAD0 **5** = OC_DETECTED_VBUS_PAD1 **6** = OC_DETECTED_VBUS_PAD2 **7** = OC_DETECTED_VBUS_PAD3 **15** = OC_DETECTION_DISABLED

XUSB_PADCTL_SS_OC_MAP_0

Offset: 0x14
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000ffff (0bxxxx,xxxx,xxxx,xxxx,1111,1111,1111,1111)

Bit	Reset	Description
15:12	0xf	**PORT3_OC_PIN**: **0** = OC_DETECTED0 **1** = OC_DETECTED1 **2** = OC_DETECTED2 **3** = OC_DETECTED3 **4** = OC_DETECTED_VBUS_PAD0 **5** = OC_DETECTED_VBUS_PAD1 **6** = OC_DETECTED_VBUS_PAD2 **7** = OC_DETECTED_VBUS_PAD3 **15** = OC_DETECTION_DISABLED
11:8	0xf	**PORT2_OC_PIN**: **0** = OC_DETECTED0 **1** = OC_DETECTED1 **2** = OC_DETECTED2 **3** = OC_DETECTED3 **4** = OC_DETECTED_VBUS_PAD0 **5** = OC_DETECTED_VBUS_PAD1 **6** = OC_DETECTED_VBUS_PAD2 **7** = OC_DETECTED_VBUS_PAD3 **15** = OC_DETECTION_DISABLED
7:4	0xf	**PORT1_OC_PIN**: **0** = OC_DETECTED0 **1** = OC_DETECTED1 **2** = OC_DETECTED2 **3** = OC_DETECTED3 **4** = OC_DETECTED_VBUS_PAD0 **5** = OC_DETECTED_VBUS_PAD1 **6** = OC_DETECTED_VBUS_PAD2 **7** = OC_DETECTED_VBUS_PAD3 **15** = OC_DETECTION_DISABLED
3:0	0xf	**PORT0_OC_PIN**: **0** = OC_DETECTED0 **1** = OC_DETECTED1 **2** = OC_DETECTED2 **3** = OC_DETECTED3 **4** = OC_DETECTED_VBUS_PAD0 **5** = OC_DETECTED_VBUS_PAD1 **6** = OC_DETECTED_VBUS_PAD2 **7** = OC_DETECTED_VBUS_PAD3 **15** = OC_DETECTION_DISABLED

XUSB_PADCTL_VBUS_OC_MAP_0

Offset: 0x18
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000f7bde (0bxxxx,xxxx,xxxx,1111,0111,1011,1101,1110)

Bit	Reset	Description
19:16	0xf	**VBUS_ENABLE3_OC_MAP**: **0** = OC_DETECTED0 **1** = OC_DETECTED1 **2** = OC_DETECTED2 **3** = OC_DETECTED3 **4** = OC_DETECTED_VBUS_PAD0 **5** = OC_DETECTED_VBUS_PAD1 **6** = OC_DETECTED_VBUS_PAD2 **7** = OC_DETECTED_VBUS_PAD3 **15** = OC_DETECTION_DISABLED
15	0x0	**VBUS_ENABLE3**: **0** = NO **1** = YES
14:11	0xf	**VBUS_ENABLE2_OC_MAP**: **0** = OC_DETECTED0 **1** = OC_DETECTED1 **2** = OC_DETECTED2 **3** = OC_DETECTED3 **4** = OC_DETECTED_VBUS_PAD0 **5** = OC_DETECTED_VBUS_PAD1 **6** = OC_DETECTED_VBUS_PAD2 **7** = OC_DETECTED_VBUS_PAD3 **15** = OC_DETECTION_DISABLED
10	0x0	**VBUS_ENABLE2**: **0** = NO **1** = YES
9:6	0xf	**VBUS_ENABLE1_OC_MAP**: **0** = OC_DETECTED0 **1** = OC_DETECTED1 **2** = OC_DETECTED2 **3** = OC_DETECTED3 **4** = OC_DETECTED_VBUS_PAD0 **5** = OC_DETECTED_VBUS_PAD1 **6** = OC_DETECTED_VBUS_PAD2 **7** = OC_DETECTED_VBUS_PAD3 **15** = OC_DETECTION_DISABLED
5	0x0	**VBUS_ENABLE1**: **0** = NO **1** = YES
4:1	0xf	**VBUS_ENABLE0_OC_MAP**: **0** = OC_DETECTED0 **1** = OC_DETECTED1 **2** = OC_DETECTED2 **3** = OC_DETECTED3 **4** = OC_DETECTED_VBUS_PAD0 **5** = OC_DETECTED_VBUS_PAD1 **6** = OC_DETECTED_VBUS_PAD2 **7** = OC_DETECTED_VBUS_PAD3 **15** = OC_DETECTION_DISABLED
0	0x0	**VBUS_ENABLE0**: **0** = NO **1** = YES

XUSB_PADCTL_OC_DET_0

Offset: 0x1c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,0000,0000,xxxx,0000,0000,xxxx,0000)

Bit	Reset	Description
27	0x0	**OC_DETECTED_INTERRUPT_ENABLE_VBUSPAD3**: **0** = NO **1** = YES
26	0x0	**OC_DETECTED_INTERRUPT_ENABLE_VBUSPAD2**: **0** = NO **1** = YES
25	0x0	**OC_DETECTED_INTERRUPT_ENABLE_VBUSPAD1**: **0** = NO **1** = YES
24	0x0	**OC_DETECTED_INTERRUPT_ENABLE_VBUSPAD0**: **0** = NO **1** = YES
23	0x0	**OC_DETECTED_INTERRUPT_ENABLE3**: **0** = NO **1** = YES
22	0x0	**OC_DETECTED_INTERRUPT_ENABLE2**: **0** = NO **1** = YES
21	0x0	**OC_DETECTED_INTERRUPT_ENABLE1**: **0** = NO **1** = YES
20	0x0	**OC_DETECTED_INTERRUPT_ENABLE0**: **0** = NO **1** = YES
15	0x0	**OC_DETECTED_VBUS_PAD3**: **0** = NO **1** = YES
14	0x0	**OC_DETECTED_VBUS_PAD2**: **0** = NO **1** = YES
13	0x0	**OC_DETECTED_VBUS_PAD1**: **0** = NO **1** = YES
12	0x0	**OC_DETECTED_VBUS_PAD0**: **0** = NO **1** = YES
11	0x0	**OC_DETECTED3**: **0** = NO **1** = YES

Bit	Reset	Description
10	0x0	**OC_DETECTED2**: **0** = NO **1** = YES
9	0x0	**OC_DETECTED1**: **0** = NO **1** = YES
8	0x0	**OC_DETECTED0**: **0** = NO **1** = YES
3	0x0	**SET_OC_DETECTED3**: **0** = NO **1** = YES
2	0x0	**SET_OC_DETECTED2**: **0** = NO **1** = YES
1	0x0	**SET_OC_DETECTED1**: **0** = NO **1** = YES
0	0x0	**SET_OC_DETECTED0**: **0** = NO **1** = YES

XUSB_PADCTL_ELPG_PROGRAM_0_0

Offset: 0x20
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,000x,xx00,00xx,x000,0xxx,0000)

Bit	Reset	Description
24	0x0	**SS_PORT3_WAKEUP_EVENT**: **0** = NO **1** = YES
23	0x0	**SS_PORT2_WAKEUP_EVENT**: **0** = NO **1** = YES
22	0x0	**SS_PORT1_WAKEUP_EVENT**: **0** = NO **1** = YES
21	0x0	**SS_PORT0_WAKEUP_EVENT**: **0** = NO **1** = YES

Bit	Reset	Description
17	0x0	**SS_PORT3_WAKE_INTERRUPT_ENABLE**: **0** = NO **1** = YES
16	0x0	**SS_PORT2_WAKE_INTERRUPT_ENABLE**: **0** = NO **1** = YES
15	0x0	**SS_PORT1_WAKE_INTERRUPT_ENABLE**: **0** = NO **1** = YES
14	0x0	**SS_PORT0_WAKE_INTERRUPT_ENABLE**: **0** = NO **1** = YES
10	0x0	**USB2_PORT3_WAKEUP_EVENT**: **0** = NO **1** = YES
9	0x0	**USB2_PORT2_WAKEUP_EVENT**: **0** = NO **1** = YES
8	0x0	**USB2_PORT1_WAKEUP_EVENT**: **0** = NO **1** = YES
7	0x0	**USB2_PORT0_WAKEUP_EVENT**: **0** = NO **1** = YES
3	0x0	**USB2_PORT3_WAKE_INTERRUPT_ENABLE**: **0** = NO **1** = YES
2	0x0	**USB2_PORT2_WAKE_INTERRUPT_ENABLE**: **0** = NO **1** = YES
1	0x0	**USB2_PORT1_WAKE_INTERRUPT_ENABLE**: **0** = NO **1** = YES
0	0x0	**USB2_PORT0_WAKE_INTERRUPT_ENABLE**: **0** = NO **1** = YES

XUSB_PADCTL_ELPG_PROGRAM_1_0

Offset: 0x24
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000fff (0bxxxx,xxxx,xxxx,xxxx,xxxx,1111,1111,1111)

Bit	Reset	Description
11	0x1	**SSP3_ELPG_VCORE_DOWN**: **0** = NO **1** = YES
10	0x1	**SSP3_ELPG_CLAMP_EN_EARLY**: **0** = NO **1** = YES
9	0x1	**SSP3_ELPG_CLAMP_EN**: **0** = NO **1** = YES
8	0x1	**SSP2_ELPG_VCORE_DOWN**: **0** = NO **1** = YES
7	0x1	**SSP2_ELPG_CLAMP_EN_EARLY**: **0** = NO **1** = YES
6	0x1	**SSP2_ELPG_CLAMP_EN**: **0** = NO **1** = YES
5	0x1	**SSP1_ELPG_VCORE_DOWN**: **0** = NO **1** = YES
4	0x1	**SSP1_ELPG_CLAMP_EN_EARLY**: **0** = NO **1** = YES
3	0x1	**SSP1_ELPG_CLAMP_EN**: **0** = NO **1** = YES
2	0x1	**SSP0_ELPG_VCORE_DOWN**: **0** = NO **1** = YES
1	0x1	**SSP0_ELPG_CLAMP_EN_EARLY**: **0** = NO **1** = YES
0	0x1	**SSP0_ELPG_CLAMP_EN**: **0** = NO **1** = YES

XUSB_PADCTL_PM_SPARE_0

Offset: 0x28
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	0x0	**OTG_PM_SPARE_BIT3**
2	0x0	**OTG_PM_SPARE_BIT2**
1	0x0	**OTG_PM_SPARE_BIT1**
0	0x0	**OTG_PM_SPARE_BIT0**

XUSB_PADCTL_SS_PORT_CFG_0

Offset: 0x2c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00001111 (0bxx00,xx00,xx00,xx00,xx01,xx01,xx01,xx01)

Bit	Reset	Description
29:28	0x0	**PORT3_DEV_SPEED_SUPPORT**: **0** = GEN1_5G **1** = GEN2_10G
25:24	0x0	**PORT2_DEV_SPEED_SUPPORT**: **0** = GEN1_5G **1** = GEN2_10G
21:20	0x0	**PORT1_DEV_SPEED_SUPPORT**: **0** = GEN1_5G **1** = GEN2_10G
17:16	0x0	**PORT0_DEV_SPEED_SUPPORT**: **0** = GEN1_5G **1** = GEN2_10G
13:12	0x1	**PORT3_SPEED_SUPPORT**: **0** = GEN1_5G **1** = GEN2_10G
9:8	0x1	**PORT2_SPEED_SUPPORT**: **0** = GEN1_5G **1** = GEN2_10G
5:4	0x1	**PORT1_SPEED_SUPPORT**: **0** = GEN1_5G **1** = GEN2_10G
1:0	0x1	**PORT0_SPEED_SUPPORT**: **0** = GEN1_5G **1** = GEN2_10G

XUSB_PADCTL_USB2_VBUS_ID_MAP_0

Offset: 0x30
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,x000,x000,x000)

Bit	Reset	Description
14:12	0x0	**PORT3_VBUS_ID_NUM**
10:8	0x0	**PORT2_VBUS_ID_NUM**
6:4	0x0	**PORT1_VBUS_ID_NUM**
2:0	0x0	**PORT0_VBUS_ID_NUM**

XUSB_PADCTL_SS_VBUS_ID_MAP_0

Offset: 0x34
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,x000,x000,x000)

Bit	Reset	Description
14:12	0x0	**PORT3_VBUS_ID_NUM**
10:8	0x0	**PORT2_VBUS_ID_NUM**
6:4	0x0	**PORT1_VBUS_ID_NUM**
2:0	0x0	**PORT0_VBUS_ID_NUM**

XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL0_0

Offset: 0x80
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000001 (0b0000,0000,0000,00xx,xx00,0000,0000,0001)

Bit	R/W	Reset	Description
31	RW	0x0	**GENERATE_SRP**: **0** = NO **1** = YES

Bit	R/W	Reset	Description
30	RW	0x0	**SRP_INTR_EN**: **0** = NO **1** = YES
29	RW	0x0	**SRP_DETECTED**: **0** = NO **1** = YES
28	RW	0x0	**SRP_DETECT_EN**: **0** = NO **1** = YES
27	RW	0x0	**DCD_INTR_EN**: **0** = NO **1** = YES
26	RW	0x0	**DCD_DETECTED**: **0** = NO **1** = YES
25	RW	0x0	**ZIN_FILTER_EN**: **0** = NO **1** = YES
24	RW	0x0	**ZIN_CHNG_INTR_EN**: **0** = NO **1** = YES
23	RW	0x0	**ZIN_ST_CHNG**: **0** = NO **1** = YES
22	RO	0x0	**ZIN**: **0** = NO **1** = YES
21	RW	0x0	**ZIP_FILTER_EN**: **0** = NO **1** = YES
20	RW	0x0	**ZIP_CHNG_INTR_EN**: **0** = NO **1** = YES
19	RW	0x0	**ZIP_ST_CHNG**: **0** = NO **1** = YES
18	RO	0x0	**ZIP**: **0** = NO **1** = YES
13	RW	0x0	**OP_I_SRC_EN**: **0** = NO **1** = YES
12	RW	0x0	**ON_SRC_EN**: **0** = NO **1** = YES

Bit	R/W	Reset	Description
11	RW	0x0	**ON_SINK_EN**: **0** = NO **1** = YES
10	RW	0x0	**OP_SRC_EN**: **0** = NO **1** = YES
9	RW	0x0	**OP_SINK_EN**: **0** = NO **1** = YES
8	RW	0x0	**VDAT_DET_FILTER_EN**: **0** = NO **1** = YES
7	RW	0x0	**VDAT_DET_CHNG_INTR_EN**: **0** = NO **1** = YES
6	RW	0x0	**VDAT_DET_ST_CHNG**: **0** = NO **1** = YES
5	RO	0x0	**VDAT_DET**: **0** = NO **1** = YES
4	RW	0x0	**VDCD_DET_FILTER_EN**: **0** = NO **1** = YES
3	RW	0x0	**VDCD_DET_CHNG_INTR_EN**: **0** = NO **1** = YES
2	RW	0x0	**VDCD_DET_ST_CHNG**: **0** = NO **1** = YES
1	RO	0x0	**VDCD_DET**: **0** = NO **1** = YES
0	RW	0x1	**PD_CHG**: **0** = NO **1** = YES

XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD0_CTL1_0

Offset: 0x84
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000040 (0bxxxx,xxxx,0000,0000,xxx0,0000,01x0,0000)

Bit	R/W	Reset	Description
23	RW	0x0	**USBON_RPU_OVRD_VAL**: **0** = NO **1** = YES
22	RW	0x0	**USBON_RPU_OVRD**: **0** = NO **1** = YES
21	RW	0x0	**USBON_RPD_OVRD_VAL**: **0** = NO **1** = YES
20	RW	0x0	**USBON_RPD_OVRD**: **0** = NO **1** = YES
19	RW	0x0	**USBOP_RPU_OVRD_VAL**: **0** = NO **1** = YES
18	RW	0x0	**USBOP_RPU_OVRD**: **0** = NO **1** = YES
17	RW	0x0	**USBOP_RPD_OVRD_VAL**: **0** = NO **1** = YES
16	RW	0x0	**USBOP_RPD_OVRD**: **0** = NO **1** = YES
12:11	RW	0x0	**VREG_DIR**
10:9	RW	0x0	**VREG_DYN_DLY**
8:7	RW	0x0	**VREG_LEV**
6	RW	0x1	**PD_VREG**
4	RW	0x0	**DIV_DET_EN**: **0** = NO **1** = YES
3	RO	0x0	**VOP_DIV2P7_DET**: **0** = NO **1** = YES
2	RO	0x0	**VOP_DIV2P0_DET**: **0** = NO **1** = YES
1	RO	0x0	**VON_DIV2P7_DET**: **0** = NO **1** = YES
0	RO	0x0	**VON_DIV2P0_DET**: **0** = NO **1** = YES

XUSB_PADCTL_USB2_OTG_PAD0_CTL_0_0

Offset: 0x88
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x26cc88e0 (0bxx10,0110,1100,1100,1000,1000,1110,0000)
PROD: 0x00000000 (0bxx0x,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	PROD	Description
29	0x1	0x0	**PD_ZI**
28	0x0	_NONE_	**PD2_OVRD_EN**
27	0x0	_NONE_	**PD2**
26	0x1	_NONE_	**PD**
25	0x1	_NONE_	**TERM_SEL**
24:21	0x6	_NONE_	**LS_FSLEW**
20:17	0x6	_NONE_	**LS_RSLEW**
16:13	0x4	_NONE_	**FS_FSLEW**
12:9	0x4	_NONE_	**FS_RSLEW**
8:6	0x3	_NONE_	**HS_SLEW**
5:0	0x20	_NONE_	**HS_CURR_LEVEL**

XUSB_PADCTL_USB2_OTG_PAD0_CTL_1_0

Offset: 0x8c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x20101044 (0bx010,0000,0001,0000,0xx1,0000,0100,0100)

Bit	R/W	Reset	Description
30:26	RW	0x8	**RPD_CTRL**
25	RO	0x0	**RPU_STATUS_HIGH**
24	RW	0x0	**RPU_SWITCH_LOW**
23	RW	0x0	**RPU_SWITCH_OVRD**

Bit	R/W	Reset	Description
22	RW	0x0	**HS_LOOPBACK_OVRD_VAL**
21	RW	0x0	**HS_LOOPBACK_OVRD_EN**
20:17	RW	0x8	**PTERM_RANGE_ADJ**
16	RW	0x0	**PD_DISC_OVRD_VAL**
15	RW	0x0	**PD_CHRP_OVRD_VAL**
12:11	RW	0x2	**HS_COUP_EN**
10:7	RW	0x0	**SPARE**
6:3	RW	0x8	**TERM_RANGE_ADJ**
2	RW	0x1	**PD_DR**
1	RW	0x0	**PD_DISC_OVRD**
0	RW	0x0	**PD_CHRP_OVRD**

XUSB_PADCTL_USB2_OTG_PAD0_CTL_2_0

Offset: 0x90
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x01a00001 (0b0000,xx01,1010,xx00,0000,0000,0000,xx01)

Bit	Reset	Description
31:28	0x0	**SPARE_AO**
25:20	0x1a	**TCTRL_SW**
17	0x0	**TCTRL_TRK_OVRD**
16	0x0	**PCTRL_TRK_OVRD**
15:10	0x0	**TCTRL_TRK**
9:4	0x0	**PCTRL_TRK**
1	0x0	**RPU_HIGH_FIXED**
0	0x1	**RPU_AUTO_SWITCH_EN**

XUSB_PADCTL_USB2_OTG_PAD0_CTL_3_0

Offset: 0x94
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,00xx,0000)

Bit	Reset	Description
8:6	0x0	**HS_RXEQ**
3:1	0x0	**HS_TXEQ**
0	0x0	**HS_DIN_DLY_SEL**

XUSB_PADCTL_USB2_OTG_PAD0_CTL_4_0

Offset: 0x98
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00001001 (0bxxxx,xxxx,0000,0000,0001,0000,0000,0001)

Bit	Reset	Description
23:20	0x0	**TCTRL_HS_STEP_SEL**
19	0x0	**TCTRL_HS_ADD_SUB**: **0** = ADD **1** = SUBTRACT
18	0x0	**TCTRL_HS_CYA_OFFSET_ADJ_EN**: **0** = DISABLE **1** = ENABLE
17:14	0x0	**TCTRL_FS_STEP_SEL**
13	0x0	**TCTRL_FS_ADD_SUB**: **0** = ADD **1** = SUBTRACT
12	0x1	**TCTRL_FS_CYA_OFFSET_ADJ_EN**: **0** = DISABLE **1** = ENABLE
11:8	0x0	**PCTRL_HS_STEP_SEL**
7	0x0	**PCTRL_HS_ADD_SUB**: **0** = ADD **1** = SUBTRACT

Bit	Reset	Description
6	0x0	**PCTRL_HS_CYA_OFFSET_ADJ_EN**: **0** = DISABLE **1** = ENABLE
5:2	0x0	**PCTRL_FS_STEP_SEL**
1	0x0	**PCTRL_FS_ADD_SUB**: **0** = ADD **1** = SUBTRACT
0	0x1	**PCTRL_FS_CYA_OFFSET_ADJ_EN**: **0** = DISABLE **1** = ENABLE

XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD1_CTL0_0

Offset: 0xc0
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000001 (0b0000,0000,0000,00xx,xx00,0000,0000,0001)

Bit	R/W	Reset	Description
31	RW	0x0	**GENERATE_SRP**: **0** = NO **1** = YES
30	RW	0x0	**SRP_INTR_EN**: **0** = NO **1** = YES
29	RW	0x0	**SRP_DETECTED**: **0** = NO **1** = YES
28	RW	0x0	**SRP_DETECT_EN**: **0** = NO **1** = YES
27	RW	0x0	**DCD_INTR_EN**: **0** = NO **1** = YES
26	RW	0x0	**DCD_DETECTED**: **0** = NO **1** = YES
25	RW	0x0	**ZIN_FILTER_EN**: **0** = NO **1** = YES
24	RW	0x0	**ZIN_CHNG_INTR_EN**: **0** = NO **1** = YES

Bit	R/W	Reset	Description
23	RW	0x0	**ZIN_ST_CHNG**: **0** = NO **1** = YES
22	RO	0x0	**ZIN**: **0** = NO **1** = YES
21	RW	0x0	**ZIP_FILTER_EN**: **0** = NO **1** = YES
20	RW	0x0	**ZIP_CHNG_INTR_EN**: **0** = NO **1** = YES
19	RW	0x0	**ZIP_ST_CHNG**: **0** = NO **1** = YES
18	RO	0x0	**ZIP**: **0** = NO **1** = YES
13	RW	0x0	**OP_I_SRC_EN**: **0** = NO **1** = YES
12	RW	0x0	**ON_SRC_EN**: **0** = NO **1** = YES
11	RW	0x0	**ON_SINK_EN**: **0** = NO **1** = YES
10	RW	0x0	**OP_SRC_EN**: **0** = NO **1** = YES
9	RW	0x0	**OP_SINK_EN**: **0** = NO **1** = YES
8	RW	0x0	**VDAT_DET_FILTER_EN**: **0** = NO **1** = YES
7	RW	0x0	**VDAT_DET_CHNG_INTR_EN**: **0** = NO **1** = YES
6	RW	0x0	**VDAT_DET_ST_CHNG**: **0** = NO **1** = YES
5	RO	0x0	**VDAT_DET**: **0** = NO **1** = YES

Bit	R/W	Reset	Description
4	RW	0x0	**VDCD_DET_FILTER_EN**: **0** = NO **1** = YES
3	RW	0x0	**VDCD_DET_CHNG_INTR_EN**: **0** = NO **1** = YES
2	RW	0x0	**VDCD_DET_ST_CHNG**: **0** = NO **1** = YES
1	RO	0x0	**VDCD_DET**: **0** = NO **1** = YES
0	RW	0x1	**PD_CHG**: **0** = NO **1** = YES

XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD1_CTL1_0

Offset: 0xc4
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000040 (0bxxxx,xxxx,0000,0000,xxx0,0000,01x0,0000)

Bit	R/W	Reset	Description
23	RW	0x0	**USBON_RPU_OVRD_VAL**: **0** = NO **1** = YES
22	RW	0x0	**USBON_RPU_OVRD**: **0** = NO **1** = YES
21	RW	0x0	**USBON_RPD_OVRD_VAL**: **0** = NO **1** = YES
20	RW	0x0	**USBON_RPD_OVRD**: **0** = NO **1** = YES
19	RW	0x0	**USBOP_RPU_OVRD_VAL**: **0** = NO **1** = YES
18	RW	0x0	**USBOP_RPU_OVRD**: **0** = NO **1** = YES

Bit	R/W	Reset	Description
17	RW	0x0	**USBOP_RPD_OVRD_VAL**: **0** = NO **1** = YES
16	RW	0x0	**USBOP_RPD_OVRD**: **0** = NO **1** = YES
12:11	RW	0x0	**VREG_DIR**
10:9	RW	0x0	**VREG_DYN_DLY**
8:7	RW	0x0	**VREG_LEV**
6	RW	0x1	**PD_VREG**
4	RW	0x0	**DIV_DET_EN**: **0** = NO **1** = YES
3	RO	0x0	**VOP_DIV2P7_DET**: **0** = NO **1** = YES
2	RO	0x0	**VOP_DIV2P0_DET**: **0** = NO **1** = YES
1	RO	0x0	**VON_DIV2P7_DET**: **0** = NO **1** = YES
0	RO	0x0	**VON_DIV2P0_DET**: **0** = NO **1** = YES

XUSB_PADCTL_USB2_OTG_PAD1_CTL_0_0

Offset: 0xc8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x26cc88e0 (0bxx10,0110,1100,1100,1000,1000,1110,0000)
PROD: 0x00000000 (0bxx0x,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	PROD	Description
29	0x1	0x0	**PD_ZI**
28	0x0	_NONE_	**PD2_OVRD_EN**
27	0x0	_NONE_	**PD2**
26	0x1	_NONE_	**PD**

Bit	Reset	PROD	Description
25	0x1	_NONE_	**TERM_SEL**
24:21	0x6	_NONE_	**LS_FSLEW**
20:17	0x6	_NONE_	**LS_RSLEW**
16:13	0x4	_NONE_	**FS_FSLEW**
12:9	0x4	_NONE_	**FS_RSLEW**
8:6	0x3	_NONE_	**HS_SLEW**
5:0	0x20	_NONE_	**HS_CURR_LEVEL**

XUSB_PADCTL_USB2_OTG_PAD1_CTL_1_0

Offset: 0xcc
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x20101044 (0bx010,0000,0001,0000,0xx1,0000,0100,0100)

Bit	R/W	Reset	Description
30:26	RW	0x8	**RPD_CTRL**
25	RO	0x0	**RPU_STATUS_HIGH**
24	RW	0x0	**RPU_SWITCH_LOW**
23	RW	0x0	**RPU_SWITCH_OVRD**
22	RW	0x0	**HS_LOOPBACK_OVRD_VAL**
21	RW	0x0	**HS_LOOPBACK_OVRD_EN**
20:17	RW	0x8	**PTERM_RANGE_ADJ**
16	RW	0x0	**PD_DISC_OVRD_VAL**
15	RW	0x0	**PD_CHRP_OVRD_VAL**
12:11	RW	0x2	**HS_COUP_EN**
10:7	RW	0x0	**SPARE**
6:3	RW	0x8	**TERM_RANGE_ADJ**
2	RW	0x1	**PD_DR**
1	RW	0x0	**PD_DISC_OVRD**

Bit	R/W	Reset	Description
0	RW	0x0	**PD_CHRP_OVRD**

XUSB_PADCTL_USB2_OTG_PAD1_CTL_2_0

Offset: 0xd0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000001 (0b0000,xx00,0000,xx00,0000,0000,0000,xx01)

Bit	Reset	Description
31:28	0x0	**SPARE_AO**
25:20	0x0	**TCTRL_SW**
17	0x0	**TCTRL_TRK_OVRD**
16	0x0	**PCTRL_TRK_OVRD**
15:10	0x0	**TCTRL_TRK**
9:4	0x0	**PCTRL_TRK**
1	0x0	**RPU_HIGH_FIXED**
0	0x1	**RPU_AUTO_SWITCH_EN**

XUSB_PADCTL_USB2_OTG_PAD1_CTL_3_0

Offset: 0xd4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,00xx,0000)

Bit	Reset	Description
8:6	0x0	**HS_RXEQ**
3:1	0x0	**HS_TXEQ**
0	0x0	**HS_DIN_DLY_SEL**

XUSB_PADCTL_USB2_OTG_PAD1_CTL_4_0

Offset: 0xd8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00001001 (0bxxxx,xxxx,0000,0000,0001,0000,0000,0001)

Bit	Reset	Description
23:20	0x0	**TCTRL_HS_STEP_SEL**
19	0x0	**TCTRL_HS_ADD_SUB**: **0** = ADD **1** = SUBTRACT
18	0x0	**TCTRL_HS_CYA_OFFSET_ADJ_EN**: **0** = DISABLE **1** = ENABLE
17:14	0x0	**TCTRL_FS_STEP_SEL**
13	0x0	**TCTRL_FS_ADD_SUB**: **0** = ADD **1** = SUBTRACT
12	0x1	**TCTRL_FS_CYA_OFFSET_ADJ_EN**: **0** = DISABLE **1** = ENABLE
11:8	0x0	**PCTRL_HS_STEP_SEL**
7	0x0	**PCTRL_HS_ADD_SUB**: **0** = ADD **1** = SUBTRACT
6	0x0	**PCTRL_HS_CYA_OFFSET_ADJ_EN**: **0** = DISABLE **1** = ENABLE
5:2	0x0	**PCTRL_FS_STEP_SEL**
1	0x0	**PCTRL_FS_ADD_SUB**: **0** = ADD **1** = SUBTRACT
0	0x1	**PCTRL_FS_CYA_OFFSET_ADJ_EN**: **0** = DISABLE **1** = ENABLE

XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD2_CTL0_0

Offset: 0x100
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000001 (0b0000,0000,0000,00xx,xx00,0000,0000,0001)

Bit	R/W	Reset	Description
31	RW	0x0	**GENERATE_SRP**: **0** = NO **1** = YES
30	RW	0x0	**SRP_INTR_EN**: **0** = NO **1** = YES
29	RW	0x0	**SRP_DETECTED**: **0** = NO **1** = YES
28	RW	0x0	**SRP_DETECT_EN**: **0** = NO **1** = YES
27	RW	0x0	**DCD_INTR_EN**: **0** = NO **1** = YES
26	RW	0x0	**DCD_DETECTED**: **0** = NO **1** = YES
25	RW	0x0	**ZIN_FILTER_EN**: **0** = NO **1** = YES
24	RW	0x0	**ZIN_CHNG_INTR_EN**: **0** = NO **1** = YES
23	RW	0x0	**ZIN_ST_CHNG**: **0** = NO **1** = YES
22	RO	0x0	**ZIN**: **0** = NO **1** = YES
21	RW	0x0	**ZIP_FILTER_EN**: **0** = NO **1** = YES
20	RW	0x0	**ZIP_CHNG_INTR_EN**: **0** = NO **1** = YES
19	RW	0x0	**ZIP_ST_CHNG**: **0** = NO **1** = YES
18	RO	0x0	**ZIP**: **0** = NO **1** = YES
13	RW	0x0	**OP_I_SRC_EN**: **0** = NO **1** = YES

Bit	R/W	Reset	Description
12	RW	0x0	**ON_SRC_EN**: **0** = NO **1** = YES
11	RW	0x0	**ON_SINK_EN**: **0** = NO **1** = YES
10	RW	0x0	**OP_SRC_EN**: **0** = NO **1** = YES
9	RW	0x0	**OP_SINK_EN**: **0** = NO **1** = YES
8	RW	0x0	**VDAT_DET_FILTER_EN**: **0** = NO **1** = YES
7	RW	0x0	**VDAT_DET_CHNG_INTR_EN**: **0** = NO **1** = YES
6	RW	0x0	**VDAT_DET_ST_CHNG**: **0** = NO **1** = YES
5	RO	0x0	**VDAT_DET**: **0** = NO **1** = YES
4	RW	0x0	**VDCD_DET_FILTER_EN**: **0** = NO **1** = YES
3	RW	0x0	**VDCD_DET_CHNG_INTR_EN**: **0** = NO **1** = YES
2	RW	0x0	**VDCD_DET_ST_CHNG**: **0** = NO **1** = YES
1	RO	0x0	**VDCD_DET**: **0** = NO **1** = YES
0	RW	0x1	**PD_CHG**: **0** = NO **1** = YES

XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD2_CTL1_0

Offset: 0x104
Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000040 (0bxxxx,xxxx,0000,0000,xxx0,0000,01x0,0000)

Bit	R/W	Reset	Description
23	RW	0x0	**USBON_RPU_OVRD_VAL**: **0** = NO **1** = YES
22	RW	0x0	**USBON_RPU_OVRD**: **0** = NO **1** = YES
21	RW	0x0	**USBON_RPD_OVRD_VAL**: **0** = NO **1** = YES
20	RW	0x0	**USBON_RPD_OVRD**: **0** = NO **1** = YES
19	RW	0x0	**USBOP_RPU_OVRD_VAL**: **0** = NO **1** = YES
18	RW	0x0	**USBOP_RPU_OVRD**: **0** = NO **1** = YES
17	RW	0x0	**USBOP_RPD_OVRD_VAL**: **0** = NO **1** = YES
16	RW	0x0	**USBOP_RPD_OVRD**: **0** = NO **1** = YES
12:11	RW	0x0	**VREG_DIR**
10:9	RW	0x0	**VREG_DYN_DLY**
8:7	RW	0x0	**VREG_LEV**
6	RW	0x1	**PD_VREG**
4	RW	0x0	**DIV_DET_EN**: **0** = NO **1** = YES
3	RO	0x0	**VOP_DIV2P7_DET**: **0** = NO **1** = YES
2	RO	0x0	**VOP_DIV2P0_DET**: **0** = NO **1** = YES
1	RO	0x0	**VON_DIV2P7_DET**: **0** = NO **1** = YES

Bit	R/W	Reset	Description
0	RO	0x0	**VON_DIV2P0_DET**: **0** = NO **1** = YES

XUSB_PADCTL_USB2_OTG_PAD2_CTL_0_0

Offset: 0x108
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x26cc88e0 (0bxx10,0110,1100,1100,1000,1000,1110,0000)
PROD: 0x00000000 (0bxx0x,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	PROD	Description
29	0x1	0x0	**PD_ZI**
28	0x0	_NONE_	**PD2_OVRD_EN**
27	0x0	_NONE_	**PD2**
26	0x1	_NONE_	**PD**
25	0x1	_NONE_	**TERM_SEL**
24:21	0x6	_NONE_	**LS_FSLEW**
20:17	0x6	_NONE_	**LS_RSLEW**
16:13	0x4	_NONE_	**FS_FSLEW**
12:9	0x4	_NONE_	**FS_RSLEW**
8:6	0x3	_NONE_	**HS_SLEW**
5:0	0x20	_NONE_	**HS_CURR_LEVEL**

XUSB_PADCTL_USB2_OTG_PAD2_CTL_1_0

Offset: 0x10c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x20101044 (0bx010,0000,0001,0000,0xx1,0000,0100,0100)

Bit	R/W	Reset	Description
30:26	RW	0x8	**RPD_CTRL**
25	RO	0x0	**RPU_STATUS_HIGH**
24	RW	0x0	**RPU_SWITCH_LOW**
23	RW	0x0	**RPU_SWITCH_OVRD**
22	RW	0x0	**HS_LOOPBACK_OVRD_VAL**
21	RW	0x0	**HS_LOOPBACK_OVRD_EN**
20:17	RW	0x8	**PTERM_RANGE_ADJ**
16	RW	0x0	**PD_DISC_OVRD_VAL**
15	RW	0x0	**PD_CHRP_OVRD_VAL**
12:11	RW	0x2	**HS_COUP_EN**
10:7	RW	0x0	**SPARE**
6:3	RW	0x8	**TERM_RANGE_ADJ**
2	RW	0x1	**PD_DR**
1	RW	0x0	**PD_DISC_OVRD**
0	RW	0x0	**PD_CHRP_OVRD**

XUSB_PADCTL_USB2_OTG_PAD2_CTL_2_0

Offset: 0x110
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000001 (0b0000,xx00,0000,xx00,0000,0000,0000,xx01)

Bit	Reset	Description
31:28	0x0	**SPARE_AO**
25:20	0x0	**TCTRL_SW**
17	0x0	**TCTRL_TRK_OVRD**
16	0x0	**PCTRL_TRK_OVRD**
15:10	0x0	**TCTRL_TRK**
9:4	0x0	**PCTRL_TRK**

Bit	Reset	Description
1	0x0	**RPU_HIGH_FIXED**
0	0x1	**RPU_AUTO_SWITCH_EN**

XUSB_PADCTL_USB2_OTG_PAD2_CTL_3_0

Offset: 0x114
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,00xx,0000)

Bit	Reset	Description
8:6	0x0	**HS_RXEQ**
3:1	0x0	**HS_TXEQ**
0	0x0	**HS_DIN_DLY_SEL**

XUSB_PADCTL_USB2_OTG_PAD2_CTL_4_0

Offset: 0x118
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00001001 (0bxxxx,xxxx,0000,0000,0001,0000,0000,0001)

Bit	Reset	Description
23:20	0x0	**TCTRL_HS_STEP_SEL**
19	0x0	**TCTRL_HS_ADD_SUB**: **0** = ADD **1** = SUBTRACT
18	0x0	**TCTRL_HS_CYA_OFFSET_ADJ_EN**: **0** = DISABLE **1** = ENABLE
17:14	0x0	**TCTRL_FS_STEP_SEL**
13	0x0	**TCTRL_FS_ADD_SUB**: **0** = ADD **1** = SUBTRACT

Bit	Reset	Description
12	0x1	**TCTRL_FS_CYA_OFFSET_ADJ_EN**: **0** = DISABLE **1** = ENABLE
11:8	0x0	**PCTRL_HS_STEP_SEL**
7	0x0	**PCTRL_HS_ADD_SUB**: **0** = ADD **1** = SUBTRACT
6	0x0	**PCTRL_HS_CYA_OFFSET_ADJ_EN**: **0** = DISABLE **1** = ENABLE
5:2	0x0	**PCTRL_FS_STEP_SEL**
1	0x0	**PCTRL_FS_ADD_SUB**: **0** = ADD **1** = SUBTRACT
0	0x1	**PCTRL_FS_CYA_OFFSET_ADJ_EN**: **0** = DISABLE **1** = ENABLE

XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD3_CTL0_0

Offset: 0x140
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000001 (0b0000,0000,0000,00xx,xx00,0000,0000,0001)

Bit	R/W	Reset	Description
31	RW	0x0	**GENERATE_SRP**: **0** = NO **1** = YES
30	RW	0x0	**SRP_INTR_EN**: **0** = NO **1** = YES
29	RW	0x0	**SRP_DETECTED**: **0** = NO **1** = YES
28	RW	0x0	**SRP_DETECT_EN**: **0** = NO **1** = YES
27	RW	0x0	**DCD_INTR_EN**: **0** = NO **1** = YES

Bit	R/W	Reset	Description
26	RW	0x0	**DCD_DETECTED**: **0** = NO **1** = YES
25	RW	0x0	**ZIN_FILTER_EN**: **0** = NO **1** = YES
24	RW	0x0	**ZIN_CHNG_INTR_EN**: **0** = NO **1** = YES
23	RW	0x0	**ZIN_ST_CHNG**: **0** = NO **1** = YES
22	RO	0x0	**ZIN**: **0** = NO **1** = YES
21	RW	0x0	**ZIP_FILTER_EN**: **0** = NO **1** = YES
20	RW	0x0	**ZIP_CHNG_INTR_EN**: **0** = NO **1** = YES
19	RW	0x0	**ZIP_ST_CHNG**: **0** = NO **1** = YES
18	RO	0x0	**ZIP**: **0** = NO **1** = YES
13	RW	0x0	**OP_I_SRC_EN**: **0** = NO **1** = YES
12	RW	0x0	**ON_SRC_EN**: **0** = NO **1** = YES
11	RW	0x0	**ON_SINK_EN**: **0** = NO **1** = YES
10	RW	0x0	**OP_SRC_EN**: **0** = NO **1** = YES
9	RW	0x0	**OP_SINK_EN**: **0** = NO **1** = YES
8	RW	0x0	**VDAT_DET_FILTER_EN**: **0** = NO **1** = YES

Bit	R/W	Reset	Description
7	RW	0x0	**VDAT_DET_CHNG_INTR_EN**: **0** = NO **1** = YES
6	RW	0x0	**VDAT_DET_ST_CHNG**: **0** = NO **1** = YES
5	RO	0x0	**VDAT_DET**: **0** = NO **1** = YES
4	RW	0x0	**VDCD_DET_FILTER_EN**: **0** = NO **1** = YES
3	RW	0x0	**VDCD_DET_CHNG_INTR_EN**: **0** = NO **1** = YES
2	RW	0x0	**VDCD_DET_ST_CHNG**: **0** = NO **1** = YES
1	RO	0x0	**VDCD_DET**: **0** = NO **1** = YES
0	RW	0x1	**PD_CHG**: **0** = NO **1** = YES

XUSB_PADCTL_USB2_BATTERY_CHRG_OTGPAD3_CTL1_0

Offset: 0x144
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000040 (0bxxxx,xxxx,0000,0000,xxx0,0000,01x0,0000)

Bit	R/W	Reset	Description
23	RW	0x0	**USBON_RPU_OVRD_VAL**: **0** = NO **1** = YES
22	RW	0x0	**USBON_RPU_OVRD**: **0** = NO **1** = YES
21	RW	0x0	**USBON_RPD_OVRD_VAL**: **0** = NO **1** = YES

Bit	R/W	Reset	Description
20	RW	0x0	**USBON_RPD_OVRD**: **0** = NO **1** = YES
19	RW	0x0	**USBOP_RPU_OVRD_VAL**: **0** = NO **1** = YES
18	RW	0x0	**USBOP_RPU_OVRD**: **0** = NO **1** = YES
17	RW	0x0	**USBOP_RPD_OVRD_VAL**: **0** = NO **1** = YES
16	RW	0x0	**USBOP_RPD_OVRD**: **0** = NO **1** = YES
12:11	RW	0x0	**VREG_DIR**
10:9	RW	0x0	**VREG_DYN_DLY**
8:7	RW	0x0	**VREG_LEV**
6	RW	0x1	**PD_VREG**
4	RW	0x0	**DIV_DET_EN**: **0** = NO **1** = YES
3	RO	0x0	**VOP_DIV2P7_DET**: **0** = NO **1** = YES
2	RO	0x0	**VOP_DIV2P0_DET**: **0** = NO **1** = YES
1	RO	0x0	**VON_DIV2P7_DET**: **0** = NO **1** = YES
0	RO	0x0	**VON_DIV2P0_DET**: **0** = NO **1** = YES

XUSB_PADCTL_USB2_OTG_PAD3_CTL_0_0

Offset: 0x148
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

Reset: 0x26cc88e0 (0bxx10,0110,1100,1100,1000,1000,1110,0000)
PROD: 0x00000000 (0bxx0x,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	PROD	Description
29	0x1	0x0	**PD_ZI**
28	0x0	_NONE_	**PD2_OVRD_EN**
27	0x0	_NONE_	**PD2**
26	0x1	_NONE_	**PD**
25	0x1	_NONE_	**TERM_SEL**
24:21	0x6	_NONE_	**LS_FSLEW**
20:17	0x6	_NONE_	**LS_RSLEW**
16:13	0x4	_NONE_	**FS_FSLEW**
12:9	0x4	_NONE_	**FS_RSLEW**
8:6	0x3	_NONE_	**HS_SLEW**
5:0	0x20	_NONE_	**HS_CURR_LEVEL**

XUSB_PADCTL_USB2_OTG_PAD3_CTL_1_0

Offset: 0x14c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x20101044 (0bx010,0000,0001,0000,0xx1,0000,0100,0100)

Bit	R/W	Reset	Description
30:26	RW	0x8	**RPD_CTRL**
25	RO	0x0	**RPU_STATUS_HIGH**
24	RW	0x0	**RPU_SWITCH_LOW**
23	RW	0x0	**RPU_SWITCH_OVRD**
22	RW	0x0	**HS_LOOPBACK_OVRD_VAL**
21	RW	0x0	**HS_LOOPBACK_OVRD_EN**
20:17	RW	0x8	**PTERM_RANGE_ADJ**
16	RW	0x0	**PD_DISC_OVRD_VAL**
15	RW	0x0	**PD_CHRP_OVRD_VAL**

Bit	R/W	Reset	Description
12:11	RW	0x2	**HS_COUP_EN**
10:7	RW	0x0	**SPARE**
6:3	RW	0x8	**TERM_RANGE_ADJ**
2	RW	0x1	**PD_DR**
1	RW	0x0	**PD_DISC_OVRD**
0	RW	0x0	**PD_CHRP_OVRD**

XUSB_PADCTL_USB2_OTG_PAD3_CTL_2_0

Offset: 0x150
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000001 (0b0000,xx00,0000,xx00,0000,0000,0000,xx01)

Bit	Reset	Description
31:28	0x0	**SPARE_AO**
25:20	0x0	**TCTRL_SW**
17	0x0	**TCTRL_TRK_OVRD**
16	0x0	**PCTRL_TRK_OVRD**
15:10	0x0	**TCTRL_TRK**
9:4	0x0	**PCTRL_TRK**
1	0x0	**RPU_HIGH_FIXED**
0	0x1	**RPU_AUTO_SWITCH_EN**

XUSB_PADCTL_USB2_OTG_PAD3_CTL_3_0

Offset: 0x154
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,00xx,0000)

Bit	Reset	Description
8:6	0x0	**HS_RXEQ**
3:1	0x0	**HS_TXEQ**
0	0x0	**HS_DIN_DLY_SEL**

XUSB_PADCTL_USB2_OTG_PAD3_CTL_4_0

Offset: 0x158
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00001001 (0bxxxx,xxxx,0000,0000,0001,0000,0000,0001)

Bit	Reset	Description
23:20	0x0	**TCTRL_HS_STEP_SEL**
19	0x0	**TCTRL_HS_ADD_SUB**: **0** = ADD **1** = SUBTRACT
18	0x0	**TCTRL_HS_CYA_OFFSET_ADJ_EN**: **0** = DISABLE **1** = ENABLE
17:14	0x0	**TCTRL_FS_STEP_SEL**
13	0x0	**TCTRL_FS_ADD_SUB**: **0** = ADD **1** = SUBTRACT
12	0x1	**TCTRL_FS_CYA_OFFSET_ADJ_EN**: **0** = DISABLE **1** = ENABLE
11:8	0x0	**PCTRL_HS_STEP_SEL**
7	0x0	**PCTRL_HS_ADD_SUB**: **0** = ADD **1** = SUBTRACT
6	0x0	**PCTRL_HS_CYA_OFFSET_ADJ_EN**: **0** = DISABLE **1** = ENABLE
5:2	0x0	**PCTRL_FS_STEP_SEL**
1	0x0	**PCTRL_FS_ADD_SUB**: **0** = ADD **1** = SUBTRACT

Bit	Reset	Description
0	0x1	**PCTRL_FS_CYA_OFFSET_ADJ_EN**: **0** = DISABLE **1** = ENABLE

XUSB_PADCTL_USB2_BATTERY_CHRG_TDCD_DBNC_TIMER_0

Offset: 0x280
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00005000 (0bxxxx,xxxx,xxxx,xxx0,0101,0000,0000,0000)

Bit	Reset	Description
16:11	0xa	**IDDIG_DBNC**
10:0	0x0	**TDCD_DBNC**

XUSB_PADCTL_USB2_BIAS_PAD_CTL_0_0

Offset: 0x284
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x060e0b10 (0bxxx0,0110,0000,1110,0xxx,1011,0001,0000)

Bit	Reset	Description
28:25	0x3	**SPARE**
24:21	0x0	**CHG_DIV**
20:18	0x3	**TEMP_COEF**
17:15	0x4	**VREF_CTRL**
11	0x1	**PD**
10:8	0x3	**TERM_OFFSET**
7:6	0x0	**HS_CHIRP_LEVEL**
5:3	0x2	**HS_DISCON_LEVEL**
2:0	0x0	**HS_SQUELCH_LEVEL**

XUSB_PADCTL_USB2_BIAS_PAD_CTL_1_0

Offset: 0x288
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x04820000 (0b0000,0100,1000,0010,0000,0000,0000,0000)
PROD: 0x0051e000 (0bxxxx,xx00,0101,0001,1110,xxxx,xxxx,xxxx)

Bit	R/W	Reset	PROD	Description
31	RW	0x0	_NONE_	**TRK_COMPLETED**
30	RW	0x0	_NONE_	**FORCE_TRK_CLK_EN**
29	RW	0x0	_NONE_	**TRK_SW_OVRD**
28	RO	0x0	_NONE_	**TRK_DONE**
27	RW	0x0	_NONE_	**TRK_START**
26	RW	0x1	_NONE_	**PD_TRK**
25:19	RW	0x10	0xa	**TRK_DONE_RESET_TIMER**
18:12	RW	0x20	0x1e	**TRK_START_TIMER**
11:6	RO	0x0	_NONE_	**PCTRL**
5:0	RO	0x0	_NONE_	**TCTRL**

XUSB_PADCTL_USB2_BIAS_PAD_CTL_2_0

Offset: 0x28c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xa02fc11c (0b1010,0000,0010,1111,1100,0001,0001,1100)

Bit	Reset	Description
31	0x1	**CYA_TRK_CODE_UPDATE_ON_IDLE**: **0** = DISABLE **1** = ENABLE
30:1	0x1017e08e	**TRK_PERIODIC_TIMER**
0	0x0	**TRK_HW_MODE**

XUSB_PADCTL_USB2_VBUS_ID_0

This is an array of four identical register entries; the register fields below apply to each entry.
Full register list is: XUSB_PADCTL_USB2_VBUS_ID_<i>, among which <i> belongs to <0..3>.
Offset: 0x360,..,0x36c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00211000 (0bxxxx,xxx0,0010,0001,0001,0000,0000,0000)

Bit	R/W	Reset	Description
24	RW	0x0	**VBUS_WAKEUP_CHNG_INTR_EN**: **0** = NO **1** = YES
23	RW	0x0	**VBUS_WAKEUP_ST_CHNG**: **0** = NO **1** = YES
22	RO	0x0	**VBUS_WAKEUP**: **0** = NO **1** = YES
21:18	RW	0x8	**ID_OVERRIDE**: **0** = ID_GND **4** = ID_A **2** = ID_B **1** = ID_C **8** = ID_FLOAT
17:16	RW	0x1	**ID_SOURCE_SELECT**: **0** = VGPIO **1** = ID_OVERRIDE
15	RW	0x0	**VBUS_WAKEUP_OVERRIDE**: **0** = NO **1** = YES
14	RW	0x0	**VBUS_OVERRIDE**: **0** = NO **1** = YES **1** = VBUS_ON **0** = VBUS_OFF
13:12	RW	0x1	**VBUS_SOURCE_SELECT**: **0** = VGPIO **1** = VBUS_OVERRIDE
11	RW	0x0	**IDDIG_CHNG_INTR_EN**: **0** = NO **1** = YES
10	RW	0x0	**IDDIG_ST_CHNG**: **0** = NO **1** = YES
9	RO	0x0	**IDDIG_C**: **0** = NO **1** = YES

Bit	R/W	Reset	Description
8	RO	0x0	**IDDIG_B**: **0** = NO **1** = YES
7	RO	0x0	**IDDIG_A**: **0** = NO **1** = YES
6	RO	0x0	**IDDIG**: **0** = NO **1** = YES
5	RW	0x0	**VBUS_VALID_CHNG_INTR_EN**: **0** = NO **1** = YES
4	RW	0x0	**VBUS_VALID_ST_CHNG**: **0** = NO **1** = YES
3	RO	0x0	**VBUS_VALID**: **0** = NO **1** = YES
2	RW	0x0	**OTG_VBUS_SESS_VLD_CHNG_INTR_EN**: **0** = NO **1** = YES
1	RW	0x0	**OTG_VBUS_SESS_VLD_ST_CHNG**: **0** = NO **1** = YES
0	RO	0x0	**OTG_VBUS_SESS_VLD**: **0** = NO **1** = YES

9.2.4.2.2 XUSB AO Registers

XUSB_AO_GATE_0
Offset: 0x0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**GATE_DBNS**: **0** = OFF **1** = ON
0	0x0	**GATE_WAKE**: **0** = OFF **1** = ON

XUSB_AO_USB_DEBOUNCE_DEL_0

Offset: 0x4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)
PROD: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0001)

Bit	Reset	PROD	Description
3:0	0x0	0x1	**UTMIP_LINE_DEB_CNT**: Number of 32-kHz debounce cycles for UTMIP port 0

XUSB_AO_UTMIP_TRIGGERS_0

This is an array of four identical register entries; the register fields below apply to each entry.
Full register list is: XUSB_AO_UTMIP_TRIGGERS_<i>, among which <i> belongs to <0..3>.
Offset: 0x40,..,0x4c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
3	X	**CLR_WAKE_ALARM**: Clear wake event **0** = NULL **1** = TRIG
2	X	**FORCE_WALK**: Force pointer walk **0** = NULL **1** = TRIG
1	X	**CAP_CFG**: Capture pad configuration **0** = NULL **1** = TRIG
0	X	**CLR_WALK_PTR**: Clear sleep walk pointer **0** = NULL **1** = TRIG

XUSB_AO_UTMIP_SAVED_STATE_0

This is an array of four identical register entries; the register fields below apply to each entry.
Full register list is: XUSB_AO_UTMIP_SAVED_STATE_<i>, among which <i> belongs to <0..3>.
Offset: 0x70,..,0x7c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,1111)

Bit	Reset	Description
7	0x0	**WAKE_EX**: Wakeup on anything Anything except a Particular Line Value **0** = OFF **1** = ON
6	0x0	**IGNORE_MASTER_CFG**
5:2	0x3	**SCRATCH**
1:0	0x3	**SPEED**: UTMIP Speed prior to DPD **0** = HS **1** = FS **2** = LS **3** = RST

XUSB_AO_UTMIP_SLEEPWALK_STATUS_0

This is an array of four identical register entries; the register fields below apply to each entry.
Full register list is: XUSB_AO_UTMIP_SLEEPWALK_STATUS_<i>, among which <i> belongs to <0..3>.
Offset: 0xa0,..,0xac
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
4	X	**WAKE_ALARM**: A wake event occurred on UTMIP port 0
3	X	**USBON_VAL**: Value of D- line detector for UTMIP port 0
2	X	**USBOP_VAL**: Value of D+ line detector for UTMIP port 0
1:0	X	**WALK_PTR**: Walk pointer for UMTIP port 0

XUSB_AO_UTMIP_SLEEPWALK_CFG_0

This is an array of four identical register entries; the register fields below apply to each entry.
Full register list is: XUSB_AO_UTMIP_SLEEPWALK_CFG_<i>, among which <i> belongs to <0..3>.
Offset: 0xd0,..,0xdc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00380001 (0bxxxx,xxxx,x011,1000,0000,0000,0000,0001)

Bit	Reset	Description
22	0x0	**MASTER_CFG_SEL**: Enables low power mode
21	0x1	**LINE_WAKEUP_EN**: Enables latching line wakeup event
20:17	0xc	**WAKE_VAL**: Line Value Wake Up Condition on UTMIP **0** = SE0 **1** = FSK **2** = FSJ **3** = SE1 **4** = DM0 **5** = DM1 **8** = DP0 **10** = DP1 **12** = NONE **13** = DMEDGE **14** = DPEDGE **15** = ANY
16	0x0	**LINEVAL_WALK_EN**: Perform Walk on USB line value wake up for UTMIP
15	0x0	**LP_CFG_ENABLE**: Used to enable some LP features (weak pd/pu ... etc)
14	0x0	**WAKE_WALK_EN**: Perform Walk on any chip wake up event for UTMIP
13	0x0	**GPIO_WALK_EN**: Perform Walk on associated GPIO event for UTMIP
12:8	0x0	**DESIGNATED_GPIO**: GPIO Number associated with UTMIP
7:4	0x0	**RESERVED**
3	0x0	**FAKE_USBON_EN**: Enable the fake line value for D- for the PMC pad macro for UTMIP
2	0x0	**FAKE_USBOP_EN**: Enable the fake line value for D+ for the PMC pad macro for UTMIP
1	0x0	**FAKE_USBON_VAL**: Fake line value for D- for the PMC pad macro for UTMIP

Bit	Reset	Description
0	0x1	**FAKE_USBOP_VAL**: Fake line value for D+ for the PMC pad macro for UTMIP

XUSB_AO_UTMIP_SLEEPWALK_0

This is an array of four identical register entries; the register fields below apply to each entry.
Full register list is: XUSB_AO_UTMIP_SLEEPWALK_<i>, among which <i> belongs to <0..3>.
Offset: 0x100,..,0x10c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xa3a3a363 (0b1010,0011,1010,0011,1010,0011,0110,0011)

Bit	Reset	Description
31	0x1	**MASTER_ENABLE_D**: Phase D Master Enable Signal
30	0x0	**HIGHZ_D**: Phase D Enable Single Ended Drivers
29	0x1	**AN_D**: Phase D Drive Single Ended Value on D- line
28	0x0	**AP_D**: Phase D Drive Single Ended Value on D+ line
27	0x0	**USBON_RPU_D**: Phase D 1.5k Ohm Pull up on D- Line
26	0x0	**USBOP_RPU_D**: Phase D 1.5k Ohm Pull Up on D+ Line
25	0x1	**USBON_RPD_D**: Phase D 15k Ohm Pull Down on D- Line
24	0x1	**USBOP_RPD_D**: Phase D 15k Ohm Pull Down on D+ Line
23	0x1	**MASTER_ENABLE_C**: Phase C Master Enable Signal
22	0x0	**HIGHZ_C**: Phase C Enable Single Ended Drivers
21	0x1	**AN_C**: Phase C Drive Single Ended Value on D- line
20	0x0	**AP_C**: Phase C Drive Single Ended Value on D+ line
19	0x0	**USBON_RPU_C**: Phase C 1.5k Ohm Pull up on D- Line

Bit	Reset	Description
18	0x0	**USBOP_RPU_C**: Phase C 1.5k Ohm Pull Up on D+ Line
17	0x1	**USBON_RPD_C**: Phase C 15k Ohm Pull Down on D- Line
16	0x1	**USBOP_RPD_C**: Phase C 15k Ohm Pull Down on D+ Line
15	0x1	**MASTER_ENABLE_B**: Phase B Master Enable Signal
14	0x0	**HIGHZ_B**: Phase B Enable Single Ended Drivers
13	0x1	**AN_B**: Phase B Drive Single Ended Value on D- line
12	0x0	**AP_B**: Phase B Drive Single Ended Value on D+ line
11	0x0	**USBON_RPU_B**: Phase B 1.5k Ohm Pull up on D- Line
10	0x0	**USBOP_RPU_B**: Phase B 1.5k Ohm Pull Up on D+ Line
9	0x1	**USBON_RPD_B**: Phase B 15k Ohm Pull Down on D- Line
8	0x1	**USBOP_RPD_B**: Phase B 15k Ohm Pull Down on D+ Line
7	0x0	**MASTER_ENABLE_A**: Phase A Master Enable Signal
6	0x1	**HIGHZ_A**: Phase A Enable Single Ended Drivers
5	0x1	**AN_A**: Phase A Drive Single Ended Value on D- line
4	0x0	**AP_A**: Phase A Drive Single Ended Value on D+ line
3	0x0	**USBON_RPU_A**: Phase A 1.5k Ohm Pull up on D- Line
2	0x0	**USBOP_RPU_A**: Phase A 1.5k Ohm Pull Up on D+ Line
1	0x1	**USBON_RPD_A**: Phase A 15k Ohm Pull Down on D- Line
0	0x1	**USBOP_RPD_A**: Phase A 15k Ohm Pull Down on D+ Line

XUSB_AO_UTMIP_PAD_CFG_0

This is an array of four identical register entries; the register fields below apply to each entry.
Full register list is: XUSB_AO_UTMIP_PAD_CFG_<i>, among which <i> belongs to <0..3>.
Offset: 0x130,..,0x13c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000300 (0bxxxx,xxxx,xxxx,xxxx,xxxx,0011,0000,0000)

Bit	Reset	Description
11	0x0	**E_DPD_OVRD_VAL**
10	0x0	**E_DPD_OVRD_EN**
9	0x1	**USBON_VAL_PD**: Power Down D- USBOP_VAL receiver
8	0x1	**USBOP_VAL_PD**: Power Down D+ USBOP_VAL receiver
7	0x0	**VREG_USE_XUSB_AO**: Use XUSB_AO Saved values for VREG pad inputs
6	0x0	**RPU_USE_XUSB_AO**: Use XUSB_AO Saved RPU* pad input controls
5	0x0	**RPD_CTRL_USE_XUSB_AO**: Use XUSB_AO Saved values for RPD_CTRL
4	0x0	**TRK_CTRL_USE_XUSB_AO**: Use XUSB_AO Saved values for TRK control inputs
3	0x0	**FSLS_USE_XUSB_AO**: Use XUSB_AO Saved values for slew controls and spare
2	0x0	**WEAKPD_ANYTIME**
1	0x0	**DP_WEAKPD_CFG**
0	0x0	**DM_WEAKPD_CFG**

XUSB_AO_UTMIP_PAD_STS_SAVED0_0

This is an array of four identical register entries; the register fields below apply to each entry.
Full register list is: XUSB_AO_UTMIP_PAD_STS_SAVED0_<i>, among which <i> belongs to <0..3>.
Offset: 0x160,..,0x16c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	X	**LO_SPD**
30:29	X	**VREG_DIR**
28	X	**PD_VREG**
27	X	**RPU_HIGH_FIXED**
26	X	**RPU_AUTO_SWITCH_EN**
25	X	**RPU_SWITCH_LOW**
24:20	X	**RPD_CTRL**
19:16	X	**SPARE_AO**
15:12	X	**LS_FSLEW**
11:8	X	**LS_RSLEW**
7:4	X	**FS_FSLEW**
3:0	X	**FS_RSLEW**

XUSB_AO_UTMIP_PAD_STS_SAVED1_0

This is an array of four identical register entries; the register fields below apply to each entry.
Full register list is: XUSB_AO_UTMIP_PAD_STS_SAVED1_<i>, among which <i> belongs to <0..3>.
Offset: 0x180,..,0x18c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
26	X	**USBOP_RPU**
25	X	**USBOP_RPD**
24	X	**USBON_RPU**
23	X	**USBON_RPD**
22:17	X	**TCTRL_SW**
16	X	**TERM_SEL**
15:10	X	**TCTRL_TRK**
9:4	X	**PCTRL_TRK**
3:2	X	**VREG_DYN_DLY**
1:0	X	**VREG_LEV**

XUSB_AO_BIAS_PAD_CFG_0

Offset: 0x1b0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**E_DPD_OVRD_VAL**
4	0x0	**E_DPD_OVRD_EN**
3:0	0x0	**SPARE_AO**: Spare Bits for BIAS pad

XUSB_AO_UPHY_CTRL_0_0

Offset: 0x1b8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1111)

Bit	Reset	Description
3	0x1	**SSP3_AUX_RX_TERM_CTRL_EN**
2	0x1	**SSP2_AUX_RX_TERM_CTRL_EN**
1	0x1	**SSP1_AUX_RX_TERM_CTRL_EN**
0	0x1	**SSP0_AUX_RX_TERM_CTRL_EN**

9.2.4.3 XHCI Controller Configuration Registers

T_XUSB_CFG_0_0

Offset: 0x0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Description
31:16	X	**DEVICE_ID_UNIT**: **8856** = XUSB

Bit	R/W	Description
15:0	X	**VENDOR_ID**: **4318** = NVIDIA

T_XUSB_CFG_1_0

The Device Control Command register provides coarse control over a device's ability to generate and respond to PCI cycles.

Offset: 0x4
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXX0XXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,x0xx,xxxx,x000)

Bit	R/W	Reset	Description
31	RO	X	**DETECTED_PERR**: The DETECTED_PERR bit indicates that the device has detected a parity error, even if parity error handing is disabled. (Bit 6 - T_CONFIG_FPCI_PERR_DISABLED) **0** = NOT_ACTIVE
30	RO	X	**SIGNALED_SERR**: The SIGNALED_SERR bit indicates that the device has asserted SERR#. **0** = NOT_ACTIVE
29	RO	X	**RECEIVED_MASTER**: The RECEIVED_MASTER bit indicates that a master device's transaction (except for Special Cycle) was terminated with a master-abort. This means that no device on the PCI bus responded to the address of the mastered transaction. All master devices must implement this bit. When this bit is set, an interrupt is signaled in the PBUS_INTR_0 register. **0** = NO_ABORT
28	RO	X	**RECEIVED_TARGET**: The RECEIVED_TARGET bit indicates that a master device's transaction was terminated with a target-abort. All master devices must implement this bit. When this bit is set, an interrupt is signaled in the PBUS_INTR_0 register. **0** = NO_ABORT
27	RO	X	**SIGNALED_TARGET**: The SIGNALED_TARGET bit indicates that the device has terminated a transaction with target-abort. Devices that will never signal target-abort do not need to implement this bit. When this bit is set, an interrupt is signaled in the PBUS_INTR_0 register. **0** = NO_ABORT

Bit	R/W	Reset	Description
26:25	RO	X	**DEVSEL_TIMING**: The DEVSEL_TIMING bits contain the timing of DEVSEL#. There are three allowable timings for assertion of DEVSEL#. These are encoded as 00b for fast, 01b for medium, and 10b for slow (11b is reserved). These bits are read only and must indicate the slowest time that a device asserts DEVSEL# for any bus command except Configuration Read and Configuration Write. FPCI positive decode device are required to respond with fast DEVSEL# (0-cycle). Only the subtractive FPCI function responds with medium DEVSEL# to accept the cycle for the subtractive bus. **0** = FAST
24	RO	X	**MASTER_DATA_PERR**: **0** = NOT_ACTIVE
23	RO	X	**FAST_BACK2BACK**: The FAST_BACK2BACK bit indicates that the device is capable of handling back-to-back transfers when the transactions are not to the same agent. This bit can be set to 1 if the device can accept these transactions, and it must be set to 0 otherwise. **1** = CAPABLE
21	RO	X	**_66MHZ**: The 66 MHZ bit indicates that the device is capable of 66 MHz PCI Bus operation. This value is initialized by a strapping bit. **1** = CAPABLE
20	RO	X	**CAPLIST**: The CAPLIST bit indicates that the device configuration space includes a capabilities list starting at the offset indicated by T_XUSB_CFG_13. **1** = PRESENT
19	RO	X	**INTR_STATUS**: The INTR_STATUS bit is read-only and reflects the state of the interrupt in the device/function. Only when the INTR_DISABLE bit in the command register is a 0 and this INTR_STATUS bit is a 1, will the device's/function's INTx# signal be asserted. Setting the INTR_DISABLE bit to 1 has no effect on the state of this bit.
10	RW	0x0	**INTR_DISABLE**: The INTR_DISABLE bit indicates that it could disable the device/function from asserting INTx#. A value of 0 enables the assertion of INTx#, and a value of 1 disables the assertion of INTx# signal. The Device Status register is used to record status information for PCI bus related events. **0** = ON **1** = OFF
9	RO	X	**BACK2BACK**: **0** = DISABLED
8	RO	X	**SERR**: **0** = DISABLED
7	RO	X	**STEP**: **0** = DISABLED
6	RO	X	**PERR**: **0** = DISABLED

Bit	R/W	Reset	Description
5	RO	X	**PALETTE_SNOOP**: The PALETTE_SNOOP bit indicates that VGA compatible devices should snoop their palette registers. When this bit is set, special palette snooping behavior is enabled (i.e., device must not respond). When the bit is reset, the device should treat palette accesses like all other accesses. VGA compatible devices should implement this bit. PALETTE_SNOOP is writable. **0** = DISABLED
4	RO	X	**WRITE_AND_INVAL**: The WRITE_AND_INVAL bit indicates that the device can use the Memory Write and Invalidate command when the transfer is aligned, and 16 bytes and the contents of the Cache Line Size Register is 4 DWORDS. When this bit is 1, masters may generate the command. When it is 0, Memory Write must be used instead. State after RST# is 0. This bit must be implemented by master devices that can generate the Memory Write and Invalidate command. **0** = DISABLED
3	RO	X	**SPECIAL_CYCLE**: **0** = DISABLED
2	RW	0x0	**BUS_MASTER**: The BUS_MASTER bit indicates that the device can act as a master on the PCI bus. A value of 0 disables the device from generating PCI accesses. A value of 1 allows the device to behave as a bus master. BUS_MASTER is writable. **0** = DISABLED **1** = ENABLED
1	RW	0x0	**MEMORY_SPACE**: The MEMORY_SPACE bit indicates that the device will respond to memory space accesses. A value of 0 disables the device response. A value of 1 allows the device to respond to Memory space accesses. MEMORY_SPACE is writable. **0** = DISABLED **1** = ENABLED
0	RW	0x0	**IO_SPACE**: The IO_SPACE bit indicates that the device will respond to I/O space accesses. A value of 0 disables the device response. A value of 1 allows the device to respond to I/O space accesses. IO_SPACE is writable. **0** = DISABLED **1** = ENABLED

T_XUSB_CFG_2_0

Offset: 0x8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BASE_CLASS**: The CLASS_CODE bits identify the generic function of the device and (in some cases) a specific register-level programming interface. The register is broken into three byte-size fields. The upper byte (at offset 0BH) is a base class code which broadly classifies the type of function the device performs. The middle-byte (at offset 0BH) is a sub-class code which identifies more specifically the function of the device. The lower byte (at offset 09H) identifies a specific register-level programming interface (if any) so that device independent software can interact with the device. The Class Code and Revision ID are defined by parameters per block. **12** = SBC
23:16	X	**SUB_CLASS**: **3** = XUSB
15:8	X	**PROG_IF**: **48** = XHCI
7:0	X	**REVISION_ID**: The REVISION_ID bits specify a device specific revision identifier. The value is chosen by the vendor. Zero is an acceptable value. This field should be viewed as a vendor defined extension to the DEVICE_ID. **161** = VAL

T_XUSB_CFG_3_0

Offset: 0xc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00XXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
23	X	**HEADER_TYPE_FUNC**: **0** = SINGLE
22:16	X	**HEADER_TYPE_DEVICE**: The HEADER_TYPE bits identify the layout of the bytes 10h through 3FH in configuration space and also whether or not the device contains multiple functions. Bit 7 in this register is used to identify a multi-function device. If the bit is 0, then the device is single function. If the bit is 1, then the device has multiple functions. Bits 6 through 0 specify the layout of bytes 10h through 3Fh. The LATENCY_TIMER and HEADER_TYPE are defined by parameters per block. **0** = NON_BRIDGE

Bit	Reset	Description
15:11	X	**LATENCY_TIMER**: The LATENCY_TIMER bits contain, in units of PCI bus clocks, the value of the Latency Timer for this PCI bus master. This register must be implemented as writable by any master that can burst more than two data phases. This register may be implemented as read-only for devices that burst two or fewer data phases, but the hardwired value must be limited to 16 or less. A typical implementation would be to build the five high-order bits (leaving the bottom three as read-only), resulting in a timer granularity of eight clocks. At reset, the register should be set to 0 (if programmable). LATENCY_TIMER bits are writable. **0** = _0_CLOCKS
7:0	X	**CACHE_LINE_SIZE**: **0** = _0

T_XUSB_CFG_4_0

Offset: 0x10
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000XXXXX (0b0000,0000,0000,00xx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31:18	RW	0x0	**BASE_ADDRESS**: The BASE_ADDRESS bits contain the base address of the device. The number of upper bits that a device actually implements depends on how much of the address space the device will respond to. A device that wants a 1 MB memory address space (using a 32-bit base address register) would build the top 12 bits of the address register, hardwiring the other bits to 0. Power-up software can determine how much address space the device required by writing a value of all 1's to the register and then reading the value back. The device will return 0's in all don't-care address bits, effectively specifying the address space required. **0** = DEFAULT
17:4	RO	X	**BAR_SIZE_256KB**: **0** = RSVD
3	RO	X	**PREFETCHABLE**: **1** = MERGABLE
2:1	RO	X	**ADDRESS_TYPE**: The ADDRESS_TYPE bits contain the type of the Base Address. It can be 32 bits, 20 bits, or 64 bits wide. **2** = _64_BIT
0	RO	X	**SPACE_TYPE**: The SPACE_TYPE bit indicates whether the register maps into Memory or I/O space. **0** = MEMORY

T_XUSB_CFG_5_0

Offset: 0x14
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**BASE_ADDRESSHI:** **0** = DEFAULT

T_XUSB_CFG_6_0

Offset: 0x18
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RSVD**: **0** = _00

T_XUSB_CFG_7_0

Offset: 0x1c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000XXXX (0b0000,0000,0000,0000,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31:16	RW	0x0	**BASE_ADDRESS**: **0** = DEFAULT
15:4	RO	X	**BAR_SIZE_64KB**: **0** = RSVD
3	RO	X	**PREFETCHABLE**: **1** = MERGABLE
2:1	RO	X	**ADDRESS_TYPE**: **2** = _64_BIT

Bit	R/W	Reset	Description
0	RO	X	**SPACE_TYPE**: **0** = MEMORY

T_XUSB_CFG_8_0

Offset: 0x20
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**BASE_ADDRESSHI**: **0** = DEFAULT

T_XUSB_CFG_9_0

This is an array of two identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_CFG_9_<i>, among which <i> belongs to <0..1>.
Offset: 0x24,0x28
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RSVD**: **0** = _00

T_XUSB_CFG_11_0

The SUBSYSTEM_VENDOR_ID bits and SUBSYSTEM_ID bits are used to uniquely identify the add-in board or subsystem where the device resides. When the device is on the motherboard, there is no serial ROM and the registers both initialize to NONE. The motherboard BIOS must set the values of the Subsystem ID and Subsystem Vendor ID by writing the proper values to the SUBSYSTEM_VENDOR_ID and SUBSYSTEM_ID bits in the PCI_T_16 register (NOT PCI_T_11).

Offset: 0x2c
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**SUBSYSTEM_ID**: **0** = NONE
15:0	0x0	**SUBSYSTEM_VENDOR_ID**: **0** = NONE

T_XUSB_CFG_12_0

Offset: 0x30
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVED**: **0** = _0

T_XUSB_CFG_13_0

Offset: 0x34
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**CAP_PTR**: The CAP_PTR bits indicate the offset into configuration space where the capabilities list begins. This always points to 0x44 where at least the PCI-PM registers are expected to reside. **68** = DEFAULT

T_XUSB_CFG_14_0

Offset: 0x38
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVED**: **0** = _0

T_XUSB_CFG_15_0

Offset: 0x3c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXX00 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	R/W	Reset	Description
31:24	RO	X	**MAX_LAT**: The MAX_LAT bits contain the maximum time the device requires to gain access to the CPI bus. This read-only register is used to specify the device's desired settings for Latency Timer values. The value specifies a period of time in units of 1/4 microsecond. Values of 0 indicate that the device has no major requirements for the settings of Latency Timers. MAX_LAT is nonzero. **0** = NO_REQUIREMENTS
23:16	RO	X	**MIN_GNT**: The MIN_GNT bits contain the length of the burst period a device needs assuming a clock rate of 33 MHz. This read-only register is used to specify the device's desired settings for Latency Timer values. The value specifies a period of time in units of 1/4 microsecond. Values of 0 indicate that the device has no major requirements for the settings of Latency Timers. MIN_GNT is nonzero. **0** = NO_REQUIREMENTS
15:8	RO	X	**INTR_PIN**: The INTR_PIN bits contain the interrupt pin the device (or device function) uses. A value of 1 corresponds to INTA#. A value of 2 corresponds to INTB#. A value of 3 corresponds to INTC#. A value of 4 corresponds to INTD#. Devices (or device functions) that don't use an interrupt pin must put a 0 in this register. This register is read-only. **1** = INTA

Bit	R/W	Reset	Description
7:0	RW	0x0	**INTR_LINE**: The INTR_LINE bits contain the interrupt routing information. The register is read/write and must be implemented by any device (or device function) that uses an interrupt pin. POST software will write the routing information into this register as it initializes and configures the system. The value in this register tells which input of the system interrupt controller(s) the device's interrupt pin is connected to. Device drivers and operating systems can use this information to determine priority and vector information. INTR_LINE is initialized to 0xff (no connection) at reset. Some PCI BIOS can't handle aliased INTR_LINEs. Some PCI BIOS' can't handle INTR_LINE initialized to 0xff. **0** = IRQ0 **1** = IRQ1 **15** = IRQ15 **255** = UNKNOWN

T_XUSB_CFG_16_0

Offset: 0x40
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**SUBSYSTEM_ID**: **0** = NONE
15:0	0x0	**SUBSYSTEM_VENDOR_ID**: **0** = NONE

T_XUSB_CFG_17_0

Offset: 0x44
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	X	**D3CPME_SUPPORT**: **1** = YES
30	X	**D3HPME_SUPPORT**: **1** = YES

Bit	Reset	Description
29	X	**D2PME_SUPPORT**: **0** = NO
28	X	**D1PME_SUPPORT**: **0** = NO
27	X	**D0PME_SUPPORT**: **1** = YES
26	X	**D2_SUPPORT**: **0** = NO
25	X	**D1_SUPPORT**: **0** = NO
24:22	X	**AUXCUR**: **0** = SELF
21	X	**DSI**: **0** = NONE
20	X	**RSVD**: **0** = _0
19	X	**PMECLK**: **0** = NOT_REQUIRED
18:16	X	**VER**: **3** = _1P2
15:8	X	**NEXT_PTR**: **100** = DEFAULT
7:0	X	**CAP**: **1** = PCIPM

T_XUSB_CFG_18_PMCSR_0

Offset: 0x48
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00XXXXXX (0bxxxx,xxxx,xxxx,xxxx,0xxx,xxx0,xxxx,xx00)

Bit	R/W	Reset	Description
23:16	RO	X	**BSE_RSVD**: **0** = _00
15	RW	0x0	**PMESTATUS**: **0** = NOT_PENDING **1** = PENDING

Bit	R/W	Reset	Description
14:13	RO	X	**DSCALE**: **0** = INIT
12:9	RO	X	**DSEL**: **0** = INIT
8	RW	0x0	**PME**: **0** = DISABLE **1** = ENABLE
7:4	RO	X	**RSVD1**: **0** = _00
3	RO	X	**NSR**: **1** = NORESET
2	RO	X	**RSVD0**: **0** = _0
1:0	RW	0x0	**PWRSTATE**: **0** = D0 **1** = D1 **2** = D2 **3** = D3H

T_XUSB_CFG_24_0

Offset: 0x60
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000X0XX (0bxxxx,xxxx,xxxx,xxxx,xx10,0000,xxxx,xxxx)

Bit	R/W	Reset	Description
15	RO	X	**RSVDP**: **0** = VALUE
14	RO	X	**NFC**: **0** = VALUE
13:8	RW	0x20	**FLADJ**: Frame Length Adjustment Register. This register is in the auxiliary power well. This feature is used to adjust any offset from the clock source that generates the clock that drives the SOF counter. When a new value is written into these six bits, the length of the frame is adjusted. Its initial programmed value is system dependent based on the accuracy of hardware USB clock and is initialized by system BIOS. This register should only be modified when the HChalted bit in the USBSTS register is a one. Changing value of this register while the host controller is operating yields undefined results. It should not be reprogrammed by USB system software unless the default or BIOS programmed values are incorrect, or the system is restoring the register while returning from a suspended state. **32** = VALUE

Bit	R/W	Reset	Description
7:0	RO	X	**SBRN**: Serial Bus Release Number Register. This register contains the release of the Universal Serial Bus Specification with which this Universal Serial Bus Host Controller module is compliant. **49** = VALUE

T_XUSB_MSI_CTRL_0

Offset: 0xc0
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0080XXXX (0bxxxx,xxx0,1000,0000,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
24	RO	0x0	**VECTOR_MASK_CAP**: The VECTOR_MASK_CAP field indicates whether or not the controller supports MSI-per-vector masking. **0** = DEFAULT
23	RO	0x1	**_64_ADDR_CAP**: The 64_ADDR_CAP field indicates whether or not the controller is capable of generating a 64-bit message address. A value of 1 means the controller is capable of generating a 64-bit message address. **1** = DEFAULT
22:20	RW	0x0	**MULT_MSG_ENABLE**: System software writes to this field to indicate the number of allocated vectors (less than or equal to the number of vectors requested). The number of vectors is aligned as a power of two. When MSI is enabled, the controller will be allocated at least one vector. **0** = DEFAULT
19:17	RO	0x0	**MULT_MSG_CAP**: System software reads this field to determine the number of requested vectors. The number of requested vectors must be aligned to a power of two. Values of 6 and 7 in this field are reserved. **0** = DEFAULT
16	RW	0x0	**MSI_ENABLE**: The MSI_ENABLE field enables the MSI capability. If MSI_ENABLE is written to a 1, the controller is permitted to use MSI to request service and is prohibited from using the legacy interrupt. System configuration software sets this bit to enable MSI. A device driver is prohibited from writing this bit to mask the controller's service request. If this bit is written to a 0, the controller is prohibited from using MSI to request service. **0** = DEFAULT
15:8	RO	X	**NEXT_PTR**: The NEXT_PTR field identifies the next item in the capabilities list. It is a read-only field. **0** = DEFAULT

Bit	R/W	Reset	Description
7:0	RO	X	**CAP_ID**: **5** = MSI

T_XUSB_MSI_ADDR1_0

Offset: 0xc4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**MSG_ADDR**: System-specified message address. When MSI is enabled, this field specifies the d-word-aligned address for the MSI memory write transaction. **0** = DEFAULT

T_XUSB_MSI_ADDR2_0

Offset: 0xc8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSG_ADDR**: System-specified message address. When MSI is enabled, this field specifies the upper 32-bits of the address for the MSI memory write transaction. The contents of this register only apply when T_XUSB_CFG_MSI_CTRL_64_ADDR_CAP bit is set. **0** = DEFAULT

T_XUSB_MSI_DATA_0

Offset: 0xcc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**MSG_DATA**: System-specified message. When MSI is enabled, the message data is driven onto the lower 16-bits of the MSI memory write. The MULT_MSG_ENABLE field in configuration register 80h specifies the number of low order message data bits that the XHCI is permitted to modify to generate its system software allocated vectors. **0** = DEFAULT

T_XUSB_MSI_MASK_0

Offset: 0xd0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**BIT**: For each mask bit that is set, the controller is prohibited from generating the associated message. ▪ bit 0 corresponds to MSI vector 0 ▪ bit 1 corresponds to MSI vector 1 ▪ bit 2 corresponds to MSI vector 2 ▪ bit 3 corresponds to MSI vector 3 ▪ bit 4 corresponds to MSI vector 4 ▪ bit 5 corresponds to MSI vector 5 ▪ bit 6 corresponds to MSI vector 6 ▪ bit 7 corresponds to MSI vector 7 Note: a value of 0 means to allow MSI generation for that vector; value of 1 means to prevent MSI generation for that vector. **0** = DEFAULT

T_XUSB_MSI_PEND_0

Offset: 0xd4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**BIT**: For each Pending bit that is set XUSB has a pending associated message. ▪ bit 0 corresponds to MSI vector 0 ▪ bit 1 corresponds to MSI vector 1 ▪ bit 2 corresponds to MSI vector 2 ▪ bit 3 corresponds to MSI vector 3 ▪ bit 4 corresponds to MSI vector 4 ▪ bit 5 corresponds to MSI vector 5 ▪ bit 6 corresponds to MSI vector 6 ▪ bit 7 corresponds to MSI vector 7 **0** = DEFAULT

T_XUSB_MSI_QUEUE_0

The MSI_QUEUE register is a private register. It specifies to which virtual channel queue (ISO or NON-ISO) that the MSI message is sent to before being sent out on FPCI bus.

Note: When the MSI message is created, the FPCI wrapper does not know which physical queue to put the MSI message into. The MSI_QUEUE setting determines this on a vector-by-vector basis. If the corresponding bit is set to 1, FPCI wrapper will put that MSI message into the ISO command/data queue; otherwise by default the message is placed in the NONISO queue. After the MSI message is popped from the ISO/NONISO queue, the settings in FPCICFG_ISOCMD, FPCICFG_PASSPW, and FPCICFG_RSPPASSPW apply. Since the XUSB FPCI wrapper, physically, only contains the NONISO command/data queues (from MCP8x and beyond), the MSI_QUEUE bits must be all set to 0 for all MSI vectors.

Offset: 0xd8
Read/Write: R/W
Parity Protection: N
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**BITS**: The BITS field specifies to which VC queue a particular MSI message will be sent to. ▪ bit 0 corresponds to MSI vector 0 ▪ bit 1 corresponds to MSI vector 1 ▪ bit 2 corresponds to MSI vector 2 ▪ bit 3 corresponds to MSI vector 3 ▪ bit 4 corresponds to MSI vector 4 ▪ bit 5 corresponds to MSI vector 5 ▪ bit 6 corresponds to MSI vector 6 ▪ bit 7 corresponds to MSI vector 7 **0** = ALL_NONISO **1** = ALL_ISO

T_XUSB_MSI_MAP_0

The MSI_MAP register is used to tell the OS that MSIs are supported in K8 mode. The register is for software information purposes only. This is the per-device HT MSI Capability Block.

Offset: 0xdc
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXX0XXXXX (0bxxxx,xxxx,xxxx,xxx0,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31:27	RO	X	**CAP_TYPE**: The CAP_TYPE field is read only at 0x15 to indicate that this is an MSI Mapping Capability block. **21** = DEFAULT
17	RO	X	**FIXD**: The FIXD bit is a read-only bit indicating if the the next 2 d-words for programmable address are present in the capability. If set, the address for mapping MSIs is fixed at 0x0000_0000_FEEx_xxxx and that this capability block is 1 d-word long. **1** = ON
16	RW	0x0	**EN**: The EN bit indicates if the mapping is active. It is cleared upon warm reset. **0** = DEFAULT
15:8	RO	X	**NEXT_PTR**: The NEXT_PTR field points to the next item in the capabilities list. It is a read-only field. **0** = DEFAULT
7:0	RO	X	**CAP_ID**: The CAP_ID field identifies that this is an HT capability list item. (Read-only as 0x8). **8** = MSI

T_XUSB_CFG_AXI_CFG_0

Offset: 0xf8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,x000,1010)

Bit	Reset	Description
8	0x0	**CLR_MSI_INTR**: **0** = NOT_PENDING **1** = PENDING **1** = TRIGGER

Bit	Reset	Description
6	0x0	**ID_MAP_USE_FULL_SUBID**: **0** = FALSE **1** = TRUE
5	0x0	**ID_MAP_FORCE_ZERO**: **0** = FALSE **1** = TRUE
4	0x0	**ORDER_RSP_VS_WRITES**: **0** = FALSE **1** = TRUE
3	0x1	**AW_FORCE_NONCOH**: **0** = FALSE **1** = TRUE
2	0x0	**AW_FORCE_COH**: **0** = FALSE **1** = TRUE
1	0x1	**AR_FORCE_NONCOH**: **0** = FALSE **1** = TRUE
0	0x0	**AR_FORCE_COH**: **0** = FALSE **1** = TRUE

T_XUSB_CFG_SRIOV_HEADER_0

Offset: 0x100
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:20	X	**NEXTCAP**: Null pointer to indicate end of PCIe Capability chain **0** = VAL
19:16	X	**VERSION**: Indicates Version of SR-IOV Capability **1** = VAL
15:0	X	**CAPID**: Indicates ID of SR-IOV Capability **16** = VAL

T_XUSB_CFG_SRIOV_CAP_0

Offset: 0x104
Read/Write: RO

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXX0000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:21	X	**MIG_INTR**: N/A as Migration is not supported **0** = VAL
0	X	**MIGRATION**: VF Migration not supported **0** = VAL

T_XUSB_CFG_SRIOV_REG0_0

Offset: 0x108
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000000X0 (0bxxxx,xxxx,xxxx,xxx0,xxxx,xxxx,xxxx,0000)

Bit	R/W	Reset	Description
16	RW	0x0	**VFMIG_STS**: N/A as Migration is not supported **0** = FALSE **1** = TRUE
4	RO	X	**ARI_CAP**: ARI Capable Hierarchy **0** = VAL
3	RW	0x0	**VF_MSE**: Memory Space Enable for all Virtual Functions **0** = FALSE **1** = TRUE
2	RW	0x0	**VFMIG_INTR_ENABLE**: Control to Enable VF Migration Interrupt **0** = FALSE **1** = TRUE
1	RW	0x0	**VFMIG_ENABLE**: **0** = FALSE **1** = TRUE
0	RW	0x0	**VF_ENABLE**: **0** = FALSE **1** = TRUE

T_XUSB_CFG_SRIOV_REG1_0

Offset: 0x10c
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**TOTAL_VFS**: Total Number of VFs supported - 4 for XUSB **4** = VAL
15:0	X	**INITIAL_VFS**: Initial number of VFs enabled. Will be same as Total VFs for SR-IOV **4** = VAL

T_XUSB_CFG_SRIOV_REG2_0

Offset: 0x110
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00XX0000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	R/W	Reset	Description
23:16	RO	X	**FUNC_DEP**: Function dependency among Physical Functions. N/A because we have a single Physical Function for XUSB **0** = VAL
15:0	RW	0x0	**NUM_VFS**: Control to specify number of VFs to be enabled **0** = INIT

T_XUSB_CFG_SRIOV_REG3_0

Offset: 0x114
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**VF_STRIDE**: Stride of each consecutive Virtual Function **1** = VAL
15:0	X	**FIRST_VF_OFFSET**: Offset of First VF in the hierarchy **1** = VAL

T_XUSB_CFG_SRIOV_REG4_0

Offset: 0x118
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXX0000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**VF_DEVICEID**: Device ID Value to be exposed for each Virtual function **4003** = VAL

T_XUSB_CFG_SRIOV_REG5_0

Offset: 0x11c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**SUP_PAGESIZE**: Bit array indicating supported Page sizes. We support only 64KB page size. **16** = VAL

T_XUSB_CFG_SRIOV_REG6_0

Offset: 0x120
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000010 (0b0000,0000,0000,0000,0000,0000,0001,0000)

Bit	Reset	Description
31:0	0x10	**SYS_PAGESIZE**: Control register to program Pagesize of the system. Only 64KB is allowed. **16** = VAL

T_XUSB_CFG_SRIOV_VFBAR0_0

Offset: 0x124
Read/Write: See table below

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000XXXXX (0b0000,0000,0000,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31:20	RW	0x0	**BASE_ADDRESS**: Holds lower 32 bits of the Base address **0** = DEFAULT
19:4	RO	X	**BAR_SIZE_256KB**: **0** = RSVD
3	RO	X	**PREFETCHABLE**: Indicates whether BAR access can be prefetched. **1** = MERGABLE
2:1	RO	X	**ADDRESS_TYPE**: Indicates whether BAR is 32-bit addressed or 64-bit addressed. Will be set to 64-bit. **2** = _64_BIT
0	RO	X	**SPACE_TYPE**: Indicates type of BAR0 - Will be set to Memory type. **0** = MEMORY

T_XUSB_CFG_SRIOV_VFBAR1_0

Offset: 0x128
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**BASE_ADDRESSHI**: **0** = DEFAULT

T_XUSB_CFG_SRIOV_VFBAR2_0

Offset: 0x12c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RSVD**: **0** = VAL

T_XUSB_CFG_SRIOV_VFBAR3_0

Offset: 0x130
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RSVD**: **0** = VAL

T_XUSB_CFG_SRIOV_VFBAR4_0

Offset: 0x134
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RSVD**: **0** = VAL

T_XUSB_CFG_SRIOV_VFBAR5_0

Offset: 0x138
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RSVD**: **0** = VAL

T_XUSB_CFG_SRIOV_VFMIG_0

Offset: 0x13c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RSVD**: **0** = VAL

T_XUSB_CFG_ARU_C11PAGESEL0_0

Offset: 0x400
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXX0X0XX (0bxxxx,xxxx,xxxx,0000,xxxx,0000,xxxx,xxx0)

Bit	R/W	Reset	Description
31:20	RO	X	**RSVD2**: **0** = _00
19	RW	0x0	**SSP3**: **0** = DESEL **1** = SEL
18	RW	0x0	**SSP2**: **0** = DESEL **1** = SEL
17	RW	0x0	**SSP1**: **0** = DESEL **1** = SEL
16	RW	0x0	**SSP0**: **0** = DESEL **1** = SEL
15:12	RO	X	**RSVD1**: **0** = _00
11	RW	0x0	**SSB3**: **0** = DESEL **1** = SEL
10	RW	0x0	**SSB2**: **0** = DESEL **1** = SEL

Bit	R/W	Reset	Description
9	RW	0x0	**SSB1**: **0** = DESEL **1** = SEL
8	RW	0x0	**SSB0**: **0** = DESEL **1** = SEL
7:1	RO	X	**RSVD0**: **0** = _00
0	RW	0x0	**RB**: **0** = DESEL **1** = SEL

T_XUSB_CFG_ARU_C11PAGESEL1_0

Offset: 0x404
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000X0XXX (0bxxxx,xxxx,0000,xx00,0000,xx00,xxxx,xxxx)

Bit	R/W	Reset	Description
23	RW	0x0	**FSP3**: **0** = DESEL **1** = SEL
22	RW	0x0	**FSP2**: **0** = DESEL **1** = SEL
21	RW	0x0	**FSP1**: **0** = DESEL **1** = SEL
20	RW	0x0	**FSP0**: **0** = DESEL **1** = SEL
19:18	RO	X	**RSVD2**: **0** = _00
17	RW	0x0	**FSB1**: **0** = DESEL **1** = SEL
16	RW	0x0	**FSB0**: **0** = DESEL **1** = SEL
15	RW	0x0	**HSP3**: **0** = DESEL **1** = SEL

Bit	R/W	Reset	Description
14	RW	0x0	**HSP2**: **0** = DESEL **1** = SEL
13	RW	0x0	**HSP1**: **0** = DESEL **1** = SEL
12	RW	0x0	**HSP0**: **0** = DESEL **1** = SEL
11:10	RO	X	**RSVD1**: **0** = _00
9	RW	0x0	**HSB1**: **0** = DESEL **1** = SEL
8	RW	0x0	**HSB0**: **0** = DESEL **1** = SEL
7:0	RO	X	**RSVD0**: **0** = _00

T_XUSB_CFG_ARU_PRIPAGESEL0_0

Offset: 0x408
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXX0X0XX (0bxxxx,xxxx,xxxx,0000,xxxx,0000,xxxx,xxx0)

Bit	R/W	Reset	Description
31:20	RO	X	**RSVD2**: **0** = _00
19	RW	0x0	**SSP3**: **0** = DESEL **1** = SEL
18	RW	0x0	**SSP2**: **0** = DESEL **1** = SEL
17	RW	0x0	**SSP1**: **0** = DESEL **1** = SEL
16	RW	0x0	**SSP0**: **0** = DESEL **1** = SEL

Bit	R/W	Reset	Description
15:12	RO	X	**RSVD1**: **0** = _00
11	RW	0x0	**SSB3**: **0** = DESEL **1** = SEL
10	RW	0x0	**SSB2**: **0** = DESEL **1** = SEL
9	RW	0x0	**SSB1**: **0** = DESEL **1** = SEL
8	RW	0x0	**SSB0**: **0** = DESEL **1** = SEL
7:1	RO	X	**RSVD0**: **0** = _00
0	RW	0x0	**RB**: **0** = DESEL **1** = SEL

T_XUSB_CFG_ARU_PRIPAGESEL1_0

Offset: 0x40c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000X0XXX (0bxxxx,xxxx,0000,xx00,0000,xx00,xxxx,xxxx)

Bit	R/W	Reset	Description
23	RW	0x0	**FSP3**: **0** = DESEL **1** = SEL
22	RW	0x0	**FSP2**: **0** = DESEL **1** = SEL
21	RW	0x0	**FSP1**: **0** = DESEL **1** = SEL
20	RW	0x0	**FSP0**: **0** = DESEL **1** = SEL
19:18	RO	X	**RSVD2**: **0** = _00

Bit	R/W	Reset	Description
17	RW	0x0	**FSB1**: **0** = DESEL **1** = SEL
16	RW	0x0	**FSB0**: **0** = DESEL **1** = SEL
15	RW	0x0	**HSP3**: **0** = DESEL **1** = SEL
14	RW	0x0	**HSP2**: **0** = DESEL **1** = SEL
13	RW	0x0	**HSP1**: **0** = DESEL **1** = SEL
12	RW	0x0	**HSP0**: **0** = DESEL **1** = SEL
11:10	RO	X	**RSVD1**: **0** = _00
9	RW	0x0	**HSB1**: **0** = DESEL **1** = SEL
8	RW	0x0	**HSB0**: **0** = DESEL **1** = SEL
7:0	RO	X	**RSVD0**: **0** = _00

T_XUSB_CFG_ARU_CSBPAGESEL0_0

Offset: 0x410
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXX0X0XX (0bxxxx,xxxx,xxxx,0000,xxxx,0000,xxxx,xxx0)

Bit	R/W	Reset	Description
31:20	RO	X	**RSVD2**: **0** = _00
19	RW	0x0	**SSP3**: **0** = DESEL **1** = SEL

Bit	R/W	Reset	Description
18	RW	0x0	**SSP2**: **0** = DESEL **1** = SEL
17	RW	0x0	**SSP1**: **0** = DESEL **1** = SEL
16	RW	0x0	**SSP0**: **0** = DESEL **1** = SEL
15:12	RO	X	**RSVD1**: **0** = _00
11	RW	0x0	**SSB3**: **0** = DESEL **1** = SEL
10	RW	0x0	**SSB2**: **0** = DESEL **1** = SEL
9	RW	0x0	**SSB1**: **0** = DESEL **1** = SEL
8	RW	0x0	**SSB0**: **0** = DESEL **1** = SEL
7:1	RO	X	**RSVD0**: **0** = _00
0	RW	0x0	**RB**: **0** = DESEL **1** = SEL

T_XUSB_CFG_ARU_CSBPAGESEL1_0

Offset: 0x414
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000X0XXX (0bxxxx,xxxx,0000,xx00,0000,xx00,xxxx,xxxx)

Bit	R/W	Reset	Description
23	RW	0x0	**FSP3**: **0** = DESEL **1** = SEL
22	RW	0x0	**FSP2**: **0** = DESEL **1** = SEL

Bit	R/W	Reset	Description
21	RW	0x0	**FSP1**: **0** = DESEL **1** = SEL
20	RW	0x0	**FSP0**: **0** = DESEL **1** = SEL
19:18	RO	X	**RSVD2**: **0** = _00
17	RW	0x0	**FSB1**: **0** = DESEL **1** = SEL
16	RW	0x0	**FSB0**: **0** = DESEL **1** = SEL
15	RW	0x0	**HSP3**: **0** = DESEL **1** = SEL
14	RW	0x0	**HSP2**: **0** = DESEL **1** = SEL
13	RW	0x0	**HSP1**: **0** = DESEL **1** = SEL
12	RW	0x0	**HSP0**: **0** = DESEL **1** = SEL
11:10	RO	X	**RSVD1**: **0** = _00
9	RW	0x0	**HSB1**: **0** = DESEL **1** = SEL
8	RW	0x0	**HSB0**: **0** = DESEL **1** = SEL
7:0	RO	X	**RSVD0**: **0** = _00

T_XUSB_CFG_ARU_CYA_0

Offset: 0x424
Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00002X1a (0bxxxx,xxxx,xxxx,xx00,0x10,xxxx,xxx1,1010)

Bit	R/W	Reset	Description
17	RW	0x0	**EMU_VMM_WRITES**: **0** = TRUE
16	RW	0x0	**EMU_VMM_READS**: **0** = TRUE
15	RO	0x0	**USE_UNSECURE_FW**: **0** = DEFAULT
13	RW	0x1	**DROP_INVLD_INTR_TGT_EVENT**: **0** = FALSE **1** = TRUE
12	RW	0x0	**EMU_VF_HOLES**: **0** = FALSE **1** = TRUE
11:8	RO	X	**EVENTQ_WAIT_CNT**: **4** = _00
4	RW	0x1	**FW_HANG_HCE**: **0** = FALSE **1** = TRUE
3	RW	0x1	**HCRST_SMI**: **0** = FALSE **1** = TRUE
2	RW	0x0	**EMU_BAR0**: **0** = TRUE **1** = FALSE
1	RW	0x1	**EMU_EREGS**: **0** = FALSE **1** = TRUE
0	RW	0x0	**EMU_HOLES**: **0** = FALSE **1** = TRUE

T_XUSB_CFG_ARU_RST_0

Offset: 0x42c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**CYA_RESET_AUX**: Used to generate SSPI AUX HCRST **0** = INIT
1	0x0	**CYA_DMAIDLE**: **0** = WAIT **1** = FORCE
0	0x0	**CFGRST**: **0** = NOT_PENDING **1** = SET

T_XUSB_CFG_ARU_RSTCNT_0

Offset: 0x430
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000010 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0001,0000)

Bit	Reset	Description
7:0	0x10	**CFGRST**: **16** = INIT

T_XUSB_CFG_ARU_CONTEXT_0

Offset: 0x43c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**FW_RELOAD**: **0** = FALSE **1** = TRUE
0	0x0	**SAVED**: **0** = FALSE **1** = TRUE

T_XUSB_CFG_ARU_CONTEXT_HS_PLS_0

Offset: 0x478
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**PORT7**: **0** = INIT
27:24	0x0	**PORT6**: **0** = INIT
23:20	0x0	**PORT5**: **0** = INIT
19:16	0x0	**PORT4**: **0** = INIT
15:12	0x0	**PORT3**: **0** = INIT
11:8	0x0	**PORT2**: **0** = INIT
7:4	0x0	**PORT1**: **0** = INIT
3:0	0x0	**PORT0**: **0** = INIT

T_XUSB_CFG_ARU_CONTEXT_FS_PLS_0

Offset: 0x47c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**PORT7**: **0** = INIT
27:24	0x0	**PORT6**: **0** = INIT
23:20	0x0	**PORT5**: **0** = INIT
19:16	0x0	**PORT4**: **0** = INIT

Bit	Reset	Description
15:12	0x0	**PORT3**: **0** = INIT
11:8	0x0	**PORT2**: **0** = INIT
7:4	0x0	**PORT1**: **0** = INIT
3:0	0x0	**PORT0**: **0** = INIT

T_XUSB_CFG_ARU_CONTEXT_HSFS_SPEED_0

Offset: 0x480
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**PORT7**: **0** = INIT
27:24	0x0	**PORT6**: **0** = INIT
23:20	0x0	**PORT5**: **0** = INIT
19:16	0x0	**PORT4**: **0** = INIT
15:12	0x0	**PORT3**: **0** = INIT
11:8	0x0	**PORT2**: **0** = INIT
7:4	0x0	**PORT1**: **0** = INIT
3:0	0x0	**PORT0**: **0** = INIT

T_XUSB_CFG_ARU_CONTEXT_HSFS_PP_0

Offset: 0x484
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxx0,xxx0,xxx0,xxx0,xxx0,xxx0,xxx0,xxx0)

Bit	Reset	Description
28	0x0	**PORT7**: **0** = INIT
24	0x0	**PORT6**: **0** = INIT
20	0x0	**PORT5**: **0** = INIT
16	0x0	**PORT4**: **0** = INIT
12	0x0	**PORT3**: **0** = INIT
8	0x0	**PORT2**: **0** = INIT
4	0x0	**PORT1**: **0** = INIT
0	0x0	**PORT0**: **0** = INIT

T_XUSB_CFG_ARU_FAULT_CTRL_0

Offset: 0x5b8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,xx01)

Bit	Reset	Description
5	0x0	**MASK_FW_HANG**: **0** = CLR **1** = SET
4	0x0	**MASK_HSE**: **0** = CLR **1** = SET
1	0x0	**FORCE_ASSERT**: **0** = DIS
0	0x1	**ENABLE**: **0** = CLR **1** = SET

T_XUSB_CFG_ARU_FAULT_STATUS_0

Offset: 0x5bc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**FW_HANG**: **0** = FALSE **1** = CLR **1** = TRUE
0	0x0	**HSE**: **0** = FALSE **1** = CLR **1** = TRUE

T_XUSB_CFG_ARU_DBG_CTRL_0

Offset: 0x5c0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**INTERNAL_DBGBUS**: **0** = FALSE **1** = TRUE
15:12	0x0	**CLKSEL_BYTE3**: **0** = INIT
11:8	0x0	**CLKSEL_BYTE2**: **0** = INIT
7:4	0x0	**CLKSEL_BYTE1**: **0** = INIT
3:0	0x0	**CLKSEL_BYTE0**: **0** = CORE **1** = FALCON **2** = SS **3** = HS **4** = FS **5** = SS_TX **6** = ULPI

T_XUSB_CFG_ARU_DBG_OFFSET0_0

Offset: 0x5c4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**REG1**: **0** = INIT
15:0	0x0	**REG0**: **0** = INIT

T_XUSB_CFG_ARU_DBG_OFFSET1_0

Offset: 0x5c8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**REG3**: **0** = INIT
15:0	0x0	**REG2**: **0** = INIT

T_XUSB_CFG_ARU_DBG_BYTESEL_0

Offset: 0x5cc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x03020100 (0bxxxx,xx11,xxxx,xx10,xxxx,xx01,xxxx,xx00)

Bit	Reset	Description
25:24	0x3	**BYTE3**: **3** = INIT

Bit	Reset	Description
17:16	0x2	**BYTE2**: **2** = INIT
9:8	0x1	**BYTE1**: **1** = INIT
1:0	0x0	**BYTE0**: **0** = INIT

T_XUSB_CFG_ARU_DBG_OUTPUT_0

Offset: 0x5d0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**FIELD**

T_XUSB_CFG_RB_BKDRREG0_0

Offset: 0x600
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x01200020 (0b0000,0001,0010,0000,0000,0000,0010,0000)

Bit	Reset	Description
31:16	0x120	**HCIVERSION**: **288** = INIT
15:8	0x0	**RSVD**: **0** = _00
7:0	0x20	**CAPLENGTH**: **32** = INIT

T_XUSB_CFG_SSBX_CTRL_0

Offset: 0x600
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00661360 (0b0000,0000,0110,0110,0001,0011,0110,0000)

Bit	Reset	Description
31:23	0x0	**MASK_SPARE**: **0** = INIT
22	0x1	**EPFETCH_WAIT_CTX_FREEZE_CLEAR**: **1** = INIT
21	0x1	**BULK_BW_RETRY**: **1** = INIT
20	0x0	**MASK_EPT_NRDY**: **0** = DIS **1** = EN
19	0x0	**MASK_NOACTIV**: **0** = DIS **1** = EN
18	0x1	**SCRATCH_NUMP**: **1** = INIT
17	0x1	**EXTRA_NUMP**: **1** = INIT
16	0x0	**XFER_CLAMP_NUMP**: **0** = DIS **1** = EN
15	0x0	**MODR8_CTRL_TFER**: **0** = DIS **1** = EN
14	0x0	**MULTI_IN_DISABLE**: **0** = FALSE **1** = TRUE
13	0x0	**ENABLE_G1_G2_CRD_SCALING**: **0** = DIS **1** = EN
12	0x1	**OUT_DATA_BUF_INTERLEAVE**: **0** = DIS **1** = EN
11	0x0	**PRDC2ASYNC_EARLY_START**: **0** = DIS **1** = EN
10	0x0	**EPFETCH_USE_PFID**: **0** = DIS **1** = EN
9	0x1	**STREAM_STOPEP_PTR0**: **0** = DIS **1** = EN

Bit	Reset	Description
8	0x1	**MASK_ISOMISS**: **0** = DIS **1** = EN
7	0x0	**MASK_OPDONE**: **0** = DIS **1** = EN
6	0x1	**CYA_RTY**: **0** = DIS **1** = EN
5	0x1	**LOCAL_ROTATE**: **0** = DIS **1** = EN
4	0x0	**CTX_FORCE_FETCH**: **0** = FALSE **1** = TRUE
3	0x0	**DIS_IN**: **0** = FALSE **1** = TRUE
2	0x0	**DIS_OUT**: **0** = FALSE **1** = TRUE
1	0x0	**DIS_PRDC**: **0** = FALSE **1** = TRUE
0	0x0	**DIS_ASYNC**: **0** = FALSE **1** = TRUE

T_XUSB_CFG_HSBX_CTRL_0

Offset: 0x600
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXX0 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**MASK_SPARE**: **0** = INIT
15	X	**EPFETCH_WAIT_CTX_FREEZE_CLEAR**: **1** = INIT
14	X	**MASK_NOACTIV**: **0** = DIS **1** = EN

Bit	Reset	Description
13	X	**SPLITS_STOP_EP_CTRL**: **0** = INIT
12	X	**MODR8_CTRL_TFER**: **0** = DIS **1** = EN
11	X	**PRDC2ASYNC_EARLY_START**: **0** = INIT
10	X	**EPFETCH_USE_PFID**: **0** = DIS **1** = EN
8	X	**MASK_ISOMISS**: **0** = DIS **1** = EN
7	X	**MASK_OPDONE**: **0** = DIS **1** = EN
5	X	**LOCAL_ROTATE**: **0** = DIS **1** = EN **1** = INIT
4	X	**CTX_FORCE_FETCH**: **0** = FALSE **1** = TRUE

T_XUSB_CFG_FSBX_CTRL_0

Offset: 0x600
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00004120 (0b0000,0000,0000,0000,0100,00x1,0x10,xxxx)

Bit	Reset	Description
31:15	0x0	**MASK_SPARE**: **0** = INIT
14	0x1	**EPFETCH_WAIT_CTX_FREEZE_CLEAR**: **1** = INIT
13	0x0	**MASK_NOACTIV**: **0** = DIS **1** = EN
12	0x0	**MODR8_CTRL_TFER**: **0** = DIS **1** = EN

Bit	Reset	Description
11	0x0	**PRDC2ASYNC_EARLY_START**: **0** = INIT
10	0x0	**EPFETCH_USE_PFID**: **0** = DIS **1** = EN
8	0x1	**MASK_ISOMISS**: **0** = DIS **1** = EN
7	0x0	**MASK_OPDONE**: **0** = DIS **1** = EN
5	0x1	**LOCAL_ROTATE**: **0** = DIS **1** = EN
4	0x0	**CTX_FORCE_FETCH**: **0** = FALSE **1** = TRUE

T_XUSB_CFG_SSPX_CORE_MIN_TX1_0

Offset: 0x600
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00010000 (0bxxxx,xxxx,0000,0001,0000,0000,0000,0000)

Bit	Reset	Description
23:0	0x10000	**TSEQ_CNT**: **65536** = INIT

T_XUSB_CFG_HSPX_CORE_CTRL_0

Offset: 0x600
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x2ac00090 (0bxx10,101x,1100,0000,0000,0000,1001,0000)

Bit	Reset	Description
29	0x1	**TOGGLE_ERR_CMPL_CYA**: **1** = INIT

Bit	Reset	Description
28	0x0	**FORCE_LINK_FS_CYA**: **0** = INIT
27	0x1	**BULK_EVICTION**: **0** = DISABLED **1** = ENABLED
26	0x0	**BUS_TIMEOUT_CYA**: **0** = DISABLED **1** = ENABLED
25	0x1	**PORT_U0_EN**: **0** = DISABLED **1** = ENABLED
23	0x1	**BITSTUFF**: **0** = DISABLED **1** = ENABLED
22	0x1	**NRZI**: **0** = DISABLED **1** = ENABLED
21:0	0x90	**EOF1**: **144** = DEFAULT

T_XUSB_CFG_FSPX_CORE_CTRL_0

Offset: 0x600
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x2ec00080 (0bxx10,111x,1100,0000,0000,0000,1000,0000)

Bit	Reset	Description
29	0x1	**_16BIT_IDLE_DET_CYA**: **0** = DISABLED **1** = ENABLED
28	0x0	**BABBLE_TIMEOUT_CYA**: **0** = DISABLED **1** = ENABLED
27	0x1	**BULK_EVICTION**: **0** = DISABLED **1** = ENABLED
26	0x1	**BUS_TIMEOUT_CYA**: **0** = DISABLED **1** = ENABLED
25	0x1	**PORT_U0_EN**: **0** = DISABLED **1** = ENABLED

Bit	Reset	Description
23	0x1	**BITSTUFF**: **0** = DISABLED **1** = ENABLED
22	0x1	**NRZI**: **0** = DISABLED **1** = ENABLED
21:0	0x80	**EOF1**: **128** = DEFAULT

T_XUSB_CFG_RB_BKDRHCSP1_0

Offset: 0x604
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000005ff (0bxxxx,xxxx,x000,0000,0000,0101,1111,1111)

Bit	Reset	Description
22:19	0x0	**RSVD0**: **0** = INIT
18:8	0x5	**MAXINTRS**: **5** = INIT
7:0	0xff	**MAXSLOTS**: **255** = INIT

T_XUSB_CFG_SSBX_PING_0

Offset: 0x604
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000801 (0bxxxx,xxxx,xxxx,xxxx,0000,1000,xxxx,x001)

Bit	Reset	Description
15:8	0x8	**INV_DURATION**: **8** = INIT
2	0x0	**CMPL**: **0** = OFF **1** = ON

Bit	Reset	Description
1	0x0	**GUESS**: **0** = OFF **1** = ON
0	0x1	**ACCEL**: **0** = OFF **1** = ON

T_XUSB_CFG_SSPX_CORE_MIN_TX2_0

Offset: 0x604
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00100010 (0b0000,0000,0001,0000,0000,0000,0001,0000)

Bit	Reset	Description
31:16	0x10	**IDLES_CNT**: **16** = INIT
15:0	0x10	**TS2_CNT**: **16** = INIT

T_XUSB_CFG_HSPX_CORE_CTRL1_0

Offset: 0x604
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x02800018 (0bxxxx,xx10,1000,0000,0000,0000,0001,1000)

Bit	Reset	Description
25	0x1	**_120MCLK**: **1** = DEFAULT
24:22	0x2	**BW_MODE**: **2** = DEFAULT
21:0	0x18	**EOF2**: **24** = DEFAULT

T_XUSB_CFG_FSPX_CORE_CTRL1_0

Offset: 0x604
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x10000028 (0b0001,0000,xx00,0000,0000,0000,0010,1000)

Bit	Reset	Description
31:24	0x10	**BABBLE_TIMEOUT**: **16** = INIT
21:0	0x28	**EOF2**: **40** = DEFAULT

T_XUSB_CFG_RB_BKDRHCSP2_0

Offset: 0x608
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000000f8 (0b0000,0000,0000,0000,0000,0000,1111,1000)

Bit	Reset	Description
31:27	0x0	**MAXSPBLO**: **0** = INIT
26	0x0	**SPR**: **0** = FALSE **1** = TRUE
25:21	0x0	**MAXSPBHI**: **0** = INIT
20:8	0x0	**RSVD**: **0** = _00
7:4	0xf	**ERST_MAX**: **15** = INIT
3:0	0x8	**IST**: **8** = INIT

T_XUSB_CFG_SSBX_THREADS_0

Offset: 0x608
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00020200 (0bxxxx,xxxx,xx00,0010,xx00,0010,xx00,0000)

Bit	Reset	Description
21:16	0x2	**MAX_OUT**: **2** = INIT
13:8	0x2	**MAX_IN**: **2** = INIT
5:0	0x0	**MAX**: **0** = INIT

T_XUSB_CFG_HSBX_THREADS_0

Offset: 0x608
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**MAX**: **0** = INIT

T_XUSB_CFG_FSBX_THREADS_0

Offset: 0x608
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**MAX**: **0** = INIT

T_XUSB_CFG_SSPX_CORE_MIN_RX_0

Offset: 0x608
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x08020808 (0b0000,1000,0000,0010,0000,1000,0000,1000)

Bit	Reset	Description
31:24	0x8	**IDLES_CNT**: **8** = INIT
23:16	0x2	**TS2HOTRST_CNT**: **2** = INIT
15:8	0x8	**TS2_CNT**: **8** = INIT
7:0	0x8	**TS1_CNT**: **8** = INIT

T_XUSB_CFG_HSPX_CORE_CNT0_0

Offset: 0x608
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00124f80 (0bxx00,0000,0001,0010,0100,1111,1000,0000)

Bit	Reset	Description
29:0	0x124f80	**U3_ENTRY_DELAY**: **1200000** = INIT

T_XUSB_CFG_FSPX_CORE_CNT0_0

Offset: 0x608
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000078 (0bxxxx,xxxx,xx00,0000,0000,0000,0111,1000)

Bit	Reset	Description
21:0	0x78	**DISCON_SE0_MIN**: **120** = INIT

T_XUSB_CFG_RB_BKDRHCSP3_0

Offset: 0x60c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0256000a (0b0000,0010,0101,0110,0000,0000,0000,1010)

Bit	Reset	Description
31:16	0x256	**U2LAT**: **598** = INIT
15:8	0x0	**RSVD**: **0** = _00
7:0	0xa	**U1LAT**: **10** = INIT

T_XUSB_CFG_SSBX_TRBFETCH_0

Offset: 0x60c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00400260 (0bxxxx,xxxx,0100,0000,xxx0,0010,x110,00xx)

Bit	Reset	Description
23:16	0x40	**NUMSKIP**: **64** = INIT
12:8	0x2	**EXTRA_NUMP**: **2** = INIT
6	0x1	**DOUBLE_SSP_WEIGHT**: **1** = INIT
5	0x1	**SINGLE_NOOP_SKIP**: **1** = INIT
4	0x0	**RDPASSPW**: **0** = INIT
3	0x0	**ISOBURST2**: **0** = INIT
2	0x0	**IDT_IN**: **0** = INIT

T_XUSB_CFG_HSBX_TRBFETCH_0

Offset: 0x60c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00400220 (0bxxxx,xxxx,0100,0000,xxx0,0010,xx10,x0xx)

Bit	Reset	Description
23:16	0x40	**NUMSKIP**: **64** = INIT
12:8	0x2	**EXTRA_NUMP**: **2** = INIT
5	0x1	**SINGLE_NOOP_SKIP**: **1** = INIT
4	0x0	**RDPASSPW**: **0** = INIT
2	0x0	**IDT_IN**: **0** = INIT

T_XUSB_CFG_FSBX_TRBFETCH_0

Offset: 0x60c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00400220 (0bxxxx,xxxx,0100,0000,xxx0,0010,xx10,x0xx)

Bit	Reset	Description
23:16	0x40	**NUMSKIP**: **64** = INIT
12:8	0x2	**EXTRA_NUMP**: **2** = INIT
5	0x1	**SINGLE_NOOP_SKIP**: **1** = INIT
4	0x0	**RDPASSPW**: **0** = INIT
2	0x0	**IDT_IN**: **0** = INIT

T_XUSB_CFG_SSPX_CORE_CTRL_0

Offset: 0x60c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xc562486b (0b1100,0101,0110,0010,0100,1000,0110,1011)

Bit	R/W	Reset	Description
31	RW	0x1	**CYA_PP_DIRECTED_DETECT**: **1** = INIT
30	RW	0x1	**CYA_PP_RDET_CTRL**: **1** = INIT
29	RW	0x0	**_10B_LOOPBACK**: **0** = DIS **1** = EN
28	RW	0x0	**SS_ONLY_DEVICE**: **0** = DIS **1** = EN
27	RO	0x0	**REMOTE_DEV_CAP_IS_OTG**: **0** = INIT
26	RW	0x1	**DISABLE_OTG_ARC_CYA**: **1** = INIT
25:16	RW	0x162	**SKIP_CNT**: **354** = INIT
15:13	RW	0x2	**CPOLL_TIMEOUT_CNT**: **2** = INIT
12	RW	0x0	**SCD_END_AFTER_SCD**: **0** = INIT
11	RW	0x1	**LDM_ENABLE**: **1** = INIT
10	RW	0x0	**DISABLE_IOPLL_PD_IN_U2**: **0** = INIT
9	RW	0x0	**DISABLE_COREPLL_PD_IN_U2**: **0** = INIT
8	RW	0x0	**U3_ENTRY_ALWAYS_ACCEPT**: **0** = INIT
7	RW	0x0	**U3_LFPS_INITIATE_ONLY**: **0** = INIT
6	RW	0x1	**PORTCAP_NUMBUF_CHK_EN**: **1** = INIT
5	RW	0x1	**PORTCAP_CHK_EN**: **1** = INIT

Bit	R/W	Reset	Description
4	RW	0x0	**TXLFPS_DIV4**: **0** = INIT
3	RW	0x1	**DYN_SYMLOCK_EN**: **1** = INIT
2	RW	0x0	**INIT_RDVAL**: **0** = INIT
1	RW	0x1	**SCD_END_AFTER_SCD_FAIL**: **1** = INIT
0	RW	0x1	**SCRAMBLER**: **0** = OFF **1** = ON

T_XUSB_CFG_HSPX_CORE_CNT1_0

Offset: 0x60c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00001770 (0bxxxx,xxxx,xx00,0000,0001,0111,0111,0000)

Bit	Reset	Description
21:0	0x1770	**CHIRP_MID**: **6000** = INIT

T_XUSB_CFG_FSPX_CORE_CNT1_0

Offset: 0x60c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00017700 (0bxx00,0000,0000,0001,0111,0111,0000,0000)

Bit	Reset	Description
29:0	0x17700	**CON_NONSE0_MIN**: **96000** = INIT

T_XUSB_CFG_RB_BKDRHCCP1_0

Offset: 0x610
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0180ff07 (0b0000,0001,1000,0000,1111,1111,0000,0111)

Bit	Reset	Description
31:16	0x180	**XECP**: **384** = INIT
15:12	0xf	**MAXPSASIZE**: **15** = INIT
11	0x1	**CFC**: **0** = FALSE **1** = TRUE
10	0x1	**SEC**: **0** = FALSE **1** = TRUE
9	0x1	**SPC**: **0** = FALSE **1** = TRUE
8	0x1	**PAE**: **0** = FALSE **1** = TRUE
7	0x0	**NSS**: **0** = FALSE **1** = TRUE
6	0x0	**LTC**: **0** = FALSE **1** = TRUE
5	0x0	**LHRC**: **0** = FALSE **1** = TRUE
4	0x0	**PIND**: **0** = FALSE **1** = TRUE
3	0x0	**PPC**: **0** = FALSE **1** = TRUE
2	0x1	**CSZ**: **0** = _32B **1** = _64B
1	0x1	**BNC**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
0	0x1	**AC64**: **0** = FALSE **1** = TRUE

T_XUSB_CFG_SSBX_DATABUF_0

Offset: 0x610
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0d110000 (0bxxx0,1101,0001,0001,xx00,0000,xx00,0000)

Bit	Reset	Description
28:26	0x3	**BUF3_MAP**: **3** = INIT
25:23	0x2	**BUF2_MAP**: **2** = INIT
22:20	0x1	**BUF1_MAP**: **1** = INIT
19:17	0x0	**BUF0_MAP**: **0** = INIT
16	0x1	**FIXED_MAP**: **1** = INIT
13:8	0x0	**MAXOUT**: **0** = INIT
5:0	0x0	**MAXIN**: **0** = INIT

T_XUSB_CFG_SSPX_CORE_CNT0_0

Offset: 0x610
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00b4780a (0bxxxx,xxxx,1011,0100,0111,1000,0000,1010)

Bit	Reset	Description
23:16	0xb4	**U1EXIT_TBRSTMIN**: **180** = INIT

Bit	Reset	Description
15:8	0x78	**U1HSHK_TBRST**: **120** = INIT
7:0	0xa	**PING_TBRST**: **10** = INIT

T_XUSB_CFG_HSPX_CORE_CNT2_0

Offset: 0x610
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0001d4c0 (0bxx00,0000,0000,0001,1101,0100,1100,0000)

Bit	Reset	Description
29:0	0x1d4c0	**RXCHIRP_MIN**: **120000** = INIT

T_XUSB_CFG_FSPX_CORE_CNT2_0

Offset: 0x610
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00023280 (0bxx00,0000,0000,0010,0011,0010,1000,0000)

Bit	Reset	Description
29:0	0x23280	**U3_ENTRY**: **144000** = INIT

T_XUSB_CFG_RB_BKDRDBOFF_0

Offset: 0x614
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00030000 (0b0000,0000,0000,0011,0000,0000,0000,0000)

Bit	Reset	Description
31:2	0xc000	**OFFSET**: **49152** = INIT
1:0	0x0	**RSVD**: **0** = _00

T_XUSB_CFG_SSBX_UPSTREAM0_0

Offset: 0x614
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x04040404 (0bxxx0,010x,xxx0,010x,xxx0,010x,xxx0,010x)

Bit	Reset	Description
28	0x0	**ASYNCOUT_FETCH_RDPASSPW**: **0** = INIT
27	0x0	**ASYNCOUT_FETCH_TC**: **0** = INIT
26	0x1	**ASYNCOUT_FETCH_RO**: **1** = INIT
25	0x0	**ASYNCOUT_FETCH_NS**: **0** = INIT
20	0x0	**ASYNCIN_FETCH_RDPASSPW**: **0** = INIT
19	0x0	**ASYNCIN_FETCH_TC**: **0** = INIT
18	0x1	**ASYNCIN_FETCH_RO**: **1** = INIT
17	0x0	**ASYNCIN_FETCH_NS**: **0** = INIT
12	0x0	**PRDCOUT_FETCH_RDPASSPW**: **0** = INIT
11	0x0	**PRDCOUT_FETCH_TC**: **0** = INIT
10	0x1	**PRDCOUT_FETCH_RO**: **1** = INIT
9	0x0	**PRDCOUT_FETCH_NS**: **0** = INIT
4	0x0	**PRDCIN_FETCH_RDPASSPW**: **0** = INIT

Bit	Reset	Description
3	0x0	**PRDCIN_FETCH_TC**: **0** = INIT
2	0x1	**PRDCIN_FETCH_RO**: **1** = INIT
1	0x0	**PRDCIN_FETCH_NS**: **0** = INIT

T_XUSB_CFG_HSBX_UPSTREAM0_0

Offset: 0x614
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00040404 (0bxxxx,xxxx,xxx0,010x,xxx0,010x,xxx0,010x)

Bit	Reset	Description
20	0x0	**ASYNCINOUT_FETCH_RDPASSPW**: **0** = INIT
19	0x0	**ASYNCINOUT_FETCH_TC**: **0** = INIT
18	0x1	**ASYNCINOUT_FETCH_RO**: **1** = INIT
17	0x0	**ASYNCINOUT_FETCH_NS**: **0** = INIT
12	0x0	**SPLITINOUT_FETCH_RDPASSPW**: **0** = INIT
11	0x0	**SPLITINOUT_FETCH_TC**: **0** = INIT
10	0x1	**SPLITINOUT_FETCH_RO**: **1** = INIT
9	0x0	**SPLITINOUT_FETCH_NS**: **0** = INIT
4	0x0	**PRDCINOUT_FETCH_RDPASSPW**: **0** = INIT
3	0x0	**PRDCINOUT_FETCH_TC**: **0** = INIT
2	0x1	**PRDCINOUT_FETCH_RO**: **1** = INIT
1	0x0	**PRDCINOUT_FETCH_NS**: **0** = INIT

T_XUSB_CFG_FSBX_UPSTREAM0_0

Offset: 0x614
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00040004 (0bxxxx,xxxx,xxx0,010x,xxxx,xxxx,xxx0,010x)

Bit	Reset	Description
20	0x0	**ASYNCINOUT_FETCH_RDPASSPW**: **0** = INIT
19	0x0	**ASYNCINOUT_FETCH_TC**: **0** = INIT
18	0x1	**ASYNCINOUT_FETCH_RO**: **1** = INIT
17	0x0	**ASYNCINOUT_FETCH_NS**: **0** = INIT
4	0x0	**PRDCINOUT_FETCH_RDPASSPW**: **0** = INIT
3	0x0	**PRDCINOUT_FETCH_TC**: **0** = INIT
2	0x1	**PRDCINOUT_FETCH_RO**: **1** = INIT
1	0x0	**PRDCINOUT_FETCH_NS**: **0** = INIT

T_XUSB_CFG_SSPX_CORE_CNT1_0

Offset: 0x614
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000078 (0bxxxx,xxxx,xxxx,0000,0000,0000,0111,1000)

Bit	Reset	Description
19:0	0x78	**POLLING_TBRST**: **120** = INIT

T_XUSB_CFG_HSPX_CORE_CNT3_0

Offset: 0x614
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000543a8 (0bxx00,0000,0000,0101,0100,0011,1010,1000)

Bit	Reset	Description
29:0	0x543a8	**U3_ENTRY**: **345000** = INIT

T_XUSB_CFG_FSPX_CORE_CNT3_0

Offset: 0x614
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00249f00 (0bxx00,0000,0010,0100,1001,1111,0000,0000)

Bit	Reset	Description
29:0	0x249f00	**RESET_RELEASE**: **2400000** = INIT

T_XUSB_CFG_RB_BKDRRTSOFF_0

Offset: 0x618
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00020000 (0b0000,0000,0000,0010,0000,0000,0000,0000)

Bit	Reset	Description
31:5	0x1000	**OFFSET**: **4096** = INIT
4:0	0x0	**RSVD**: **0** = _00

T_XUSB_CFG_SSBX_UPSTREAM1_0

Offset: 0x618
Read/Write: R/W

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000400 (0bxxxx,000x,xxxx,000x,xxx0,010x,xxxx,000x)

Bit	Reset	Description
27	0x0	**PINGRSP_TC**: **0** = INIT
26	0x0	**PINGRSP_RO**: **0** = INIT
25	0x0	**PINGRSP_NS**: **0** = INIT
19	0x0	**PINGSCHED_TC**: **0** = INIT
18	0x0	**PINGSCHED_RO**: **0** = INIT
17	0x0	**PINGSCHED_NS**: **0** = INIT
12	0x0	**PINGFETCH_RDPASSPW**: **0** = INIT
11	0x0	**PINGFETCH_TC**: **0** = INIT
10	0x1	**PINGFETCH_RO**: **1** = INIT
9	0x0	**PINGFETCH_NS**: **0** = INIT
3	0x0	**COMPL_TC**: **0** = INIT
2	0x0	**COMPL_RO**: **0** = INIT
1	0x0	**COMPL_NS**: **0** = INIT

T_XUSB_CFG_HSBX_UPSTREAM1_0

Offset: 0x618
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,000x)

Bit	Reset	Description
3	0x0	**COMPL_TC**: **0** = INIT
2	0x0	**COMPL_RO**: **0** = INIT
1	0x0	**COMPL_NS**: **0** = INIT

T_XUSB_CFG_FSBX_UPSTREAM1_0

Offset: 0x618
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,000x)

Bit	Reset	Description
3	0x0	**COMPL_TC**: **0** = INIT
2	0x0	**COMPL_RO**: **0** = INIT
1	0x0	**COMPL_NS**: **0** = INIT

T_XUSB_CFG_SSPX_CORE_CNT2_0

Offset: 0x618
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000001e0 (0bxxxx,xxxx,xxxx,0000,0000,0001,1110,0000)

Bit	Reset	Description
19:0	0x1e0	**PM_LC_TIMER**: **480** = INIT

T_XUSB_CFG_HSPX_CORE_CNT4_0

Offset: 0x618
Read/Write: R/W

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000cd140 (0bxx00,0000,0000,1100,1101,0001,0100,0000)

Bit	Reset	Description
29:0	0xcd140	**RXCHIRP_MAX**: **840000** = INIT

T_XUSB_CFG_FSPX_CORE_CNT4_0

Offset: 0x618
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000ea600 (0bxx00,0000,0000,1110,1010,0110,0000,0000)

Bit	Reset	Description
29:0	0xea600	**U3_RESUMEK_MIN**: **960000** = INIT

T_XUSB_CFG_RB_BKDRHCCP2_0

Offset: 0x61c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000001ff (0b0000,0000,0000,0000,0000,0001,1111,1111)

Bit	Reset	Description
31:10	0x0	**RSVD**: **0** = INIT
9	0x0	**VTC**: **0** = FALSE **1** = TRUE
8	0x1	**GSC**: **0** = FALSE **1** = TRUE
7	0x1	**ETC_TSC**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
6	0x1	**ETC**: **0** = FALSE **1** = TRUE
5	0x1	**CIC**: **0** = FALSE **1** = TRUE
4	0x1	**LEC**: **0** = FALSE **1** = TRUE
3	0x1	**CTC**: **0** = FALSE **1** = TRUE
2	0x1	**FSC**: **0** = FALSE **1** = TRUE
1	0x1	**CMC**: **0** = FALSE **1** = TRUE
0	0x1	**U3C**: **0** = FALSE **1** = TRUE

T_XUSB_CFG_SSBX_DMA_0

Offset: 0x61c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00001050 (0b0000,0000,0000,0000,x001,0000,x101,0000)

Bit	Reset	Description
31:24	0x0	**WR_LIMIT**: **0** = INIT
23:16	0x0	**RD_LIMIT**: **0** = INIT
14	0x0	**WR_USE_WR_ACKS**: **0** = INIT
13	0x0	**WR_UPSTREAM_RO**: **0** = INIT
12:8	0x10	**WR_MAX_ALOM**: **16** = INIT
6	0x1	**RD_UPSTREAM_RO**: **1** = INIT

Bit	Reset	Description
5	0x0	**RD_UPSTREAM_RDPASSPW**: **0** = INIT
4:0	0x10	**RD_MAX_ALOM**: **16** = INIT

T_XUSB_CFG_HSBX_DMA_0

Offset: 0x61c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00001030 (0b0000,0000,0000,0000,x001,0000,x011,0000)

Bit	Reset	Description
31:24	0x0	**WR_LIMIT**: **0** = INIT
23:16	0x0	**RD_LIMIT**: **0** = INIT
14	0x0	**WR_USE_WR_ACKS**: **0** = INIT
13	0x0	**WR_UPSTREAM_RO**: **0** = INIT
12:8	0x10	**WR_MAX_ALOM**: **16** = INIT
6	0x0	**RD_UPSTREAM_RDPASSPW**: **0** = INIT
5	0x1	**RD_UPSTREAM_RO**: **1** = INIT
4:0	0x10	**RD_MAX_ALOM**: **16** = INIT

T_XUSB_CFG_FSBX_DMA_0

Offset: 0x61c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00001030 (0b0000,0000,0000,0000,x001,0000,x011,0000)

Bit	Reset	Description
31:24	0x0	**WR_LIMIT**: **0** = INIT
23:16	0x0	**RD_LIMIT**: **0** = INIT
14	0x0	**WR_USE_WR_ACKS**: **0** = INIT
13	0x0	**WR_UPSTREAM_RO**: **0** = INIT
12:8	0x10	**WR_MAX_ALOM**: **16** = INIT
6	0x0	**RD_UPSTREAM_RDPASSPW**: **0** = INIT
5	0x1	**RD_UPSTREAM_RO**: **1** = INIT
4:0	0x10	**RD_MAX_ALOM**: **16** = INIT

T_XUSB_CFG_SSPX_CORE_CNT3_0

Offset: 0x61c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000004cc (0bxxxx,xxxx,xxxx,0000,0000,0100,1100,1100)

Bit	Reset	Description
19:0	0x4cc	**PEND_HP_TIMER**: **1228** = INIT

T_XUSB_CFG_HSPX_CORE_CNT5_0

Offset: 0x61c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x005b8d80 (0bxx00,0000,0101,1011,1000,1101,1000,0000)

Bit	Reset	Description
29:0	0x5b8d80	**RESET_RELEASE**: **6000000** = INIT

T_XUSB_CFG_FSPX_CORE_CNT5_0

Offset: 0x61c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000514 (0bx000,0000,0000,0000,0000,0101,0001,0100)

Bit	Reset	Description
30	0x0	**ADJUST_ADD**: **0** = INIT
29:22	0x0	**MAX_TIME_ADJUST**: **0** = INIT
21:0	0x514	**FS_FSM_TIMEOUT_VAL**: **1300** = INIT

T_XUSB_CFG_RB_COUNT0_0

Offset: 0x620
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000026 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0010,0110)

Bit	Reset	Description
9	0x0	**EHB_DLY_IMOD**: **0** = INIT
8	0x0	**OVERRUN_DEQPTR0**: **0** = INIT
7:0	0x26	**IMOD**: **38** = _250NS

T_XUSB_CFG_SSBX_EVENTS_0

Offset: 0x620
Read/Write: R/W

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1010)

Bit	Reset	Description
3	0x1	**STOP_ON_NOOP**: **1** = INIT
2	0x0	**STOP_ON_EVT_DATA**: **0** = INIT
1	0x1	**RING_END_CHECK**: **1** = INIT
0	0x0	**RETIREFLOW**: **0** = INIT

T_XUSB_CFG_HSBX_EVENTS_0

Offset: 0x620
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000008 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1000)

Bit	Reset	Description
3	0x1	**STOP_ON_NOOP**: **1** = INIT
2	0x0	**STOP_ON_EVT_DATA**: **0** = INIT
1	0x0	**RING_END_CHECK**: **0** = INIT
0	0x0	**RETIREFLOW**: **0** = INIT

T_XUSB_CFG_FSBX_EVENTS_0

Offset: 0x620
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000008 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1000)

Bit	Reset	Description
3	0x1	**STOP_ON_NOOP**: **1** = INIT
2	0x0	**STOP_ON_EVT_DATA**: **0** = INIT
1	0x0	**RING_END_CHECK**: **0** = INIT
0	0x0	**RETIREFLOW**: **0** = INIT

T_XUSB_CFG_SSPX_CORE_CNT4_0

Offset: 0x620
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000003c0 (0bxxxx,xxxx,xxxx,0000,0000,0011,1100,0000)

Bit	Reset	Description
19:0	0x3c0	**PM_ENTRY**: **960** = INIT

T_XUSB_CFG_HSPX_CORE_CNT6_0

Offset: 0x620
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00249f00 (0bxx00,0000,0010,0100,1001,1111,0000,0000)

Bit	Reset	Description
29:0	0x249f00	**U3_RESUMEK_MIN**: **2400000** = INIT

T_XUSB_CFG_FSPX_CORE_CNT6_0

Offset: 0x620
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00002134 (0bxxxx,xxxx,xx00,0000,0010,0001,0011,0100)

Bit	Reset	Description
21:0	0x2134	**LS_FSM_TIMEOUT_VAL**: **8500** = INIT

T_XUSB_CFG_RB_SPARE0_0

Offset: 0x624
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**F**: **0** = _00

T_XUSB_CFG_SSBX_EPFETCH_EARLYSTART_0

Offset: 0x624
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x034801a4 (0b0000,0011,0100,1000,0000,0001,1010,0100)

Bit	Reset	Description
31:16	0x348	**PRDCOUT**: **840** = INIT
15:0	0x1a4	**PRDCIN**: **420** = INIT

T_XUSB_CFG_HSBX_EPFETCH_EARLYSTART_0

Offset: 0x624
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x03480348 (0b0000,0011,0100,1000,0000,0011,0100,1000)

Bit	Reset	Description
31:16	0x348	**SPLIT**: **840** = INIT
15:0	0x348	**PRDC**: **840** = INIT

T_XUSB_CFG_FSBX_EPFETCH_EARLYSTART_0

Offset: 0x624
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000348 (0bxxxx,xxxx,xxxx,xxxx,0000,0011,0100,1000)

Bit	Reset	Description
15:0	0x348	**PRDC**: **840** = INIT

T_XUSB_CFG_SSPX_CORE_CNT5_0

Offset: 0x624
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000004b0 (0bxxxx,xxxx,xxxx,0000,0000,0100,1011,0000)

Bit	Reset	Description
19:0	0x4b0	**POLLING_TRPT**: **1200** = INIT

T_XUSB_CFG_HSPX_CORE_CNT7_0

Offset: 0x624
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00005dc0 (0bxxxx,xxxx,xx00,0000,0101,1101,1100,0000)

Bit	Reset	Description
21:0	0x5dc0	**RST_PROLOG**: **24000** = INIT

T_XUSB_CFG_FSPX_CORE_CNT7_0

Offset: 0x624
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000001f4 (0bxxxx,xxxx,xxx0,0000,0000,0001,1111,0100)

Bit	Reset	Description
20:0	0x1f4	**FS_PIPE_MIN_DLY**: **500** = INIT

T_XUSB_CFG_RB_IFRDMA_ATTR_0

Offset: 0x628
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000002 (0bxxxx,xxxx,xxxx,xxxx,xxxx,x000,0000,0010)

Bit	Reset	Description
10:7	0x0	**VF_ID**: **0** = INIT
6	0x0	**NARROW**: **0** = INIT
5	0x0	**ACK**: **0** = INIT
4	0x0	**ALOM**: **0** = INIT
3	0x0	**RDPASSPW**: **0** = INIT
2	0x0	**RO**: **0** = INIT
1	0x1	**NS**: **1** = INIT

Bit	Reset	Description
0	0x0	**TC**: **0** = INIT

T_XUSB_CFG_HSBX_NAK_MOD_0

Offset: 0x628
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xb0000f00 (0b1011,0000,0000,0000,0000,1111,0000,0000)

Bit	Reset	Description
31	0x1	**EN**: **0** = NO **1** = YES
30	0x0	**EN_LOCALROTATE**: **0** = NO **1** = YES
29	0x1	**PING_CNTRL**: **0** = NO **1** = YES
28	0x1	**PING_BULK**: **0** = NO **1** = YES
27:0	0xf00	**COUNTER**: **3840** = INIT

T_XUSB_CFG_FSBX_NAK_MOD_0

Offset: 0x628
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x80000f00 (0b10xx,0000,0000,0000,0000,1111,0000,0000)

Bit	Reset	Description
31	0x1	**EN**: **0** = NO **1** = YES
30	0x0	**EN_LOCALROTATE**: **0** = INIT **1** = YES

Bit	Reset	Description
27:0	0xf00	**COUNTER**: **3840** = INIT

T_XUSB_CFG_SSPX_CORE_CNT6_0

Offset: 0x628
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000004b0 (0bxxxx,xxxx,xxxx,0000,0000,0100,1011,0000)

Bit	Reset	Description
19:0	0x4b0	**LDN_LUP_TIMER**: **1200** = INIT

T_XUSB_CFG_HSPX_CORE_CNT8_0

Offset: 0x628
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000012c (0bxxxx,xxxx,xx00,0000,0000,0001,0010,1100)

Bit	Reset	Description
21:0	0x12c	**DISCON_SE0_MIN**: **300** = INIT

T_XUSB_CFG_FSPX_CORE_CNT8_0

Offset: 0x628
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000001f4 (0bxxxx,xxxx,xxx0,0000,0000,0001,1111,0100)

Bit	Reset	Description
20:0	0x1f4	**FS_PIPE_MAX_DLY**: **500** = INIT

T_XUSB_CFG_SSBX_DMA_CTRL_0

Offset: 0x62c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
18:16	0x0	**WR_UPSTREAM_NUM_REQ**: The default value of DMA write request size is set by XUSB_DMA_REQ_SIZE parameter **0** = INIT
2:0	0x0	**RD_UPSTREAM_NUM_REQ**: The default value of DMA read request size is set by XUSB_DMA_REQ_SIZE parameter **0** = INIT

T_XUSB_CFG_HSBX_DMA_CTRL_0

Offset: 0x62c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
18:16	0x0	**WR_UPSTREAM_NUM_REQ**: The default value of DMA write request size is set by XUSB_DMA_REQ_SIZE parameter **0** = INIT
2:0	0x0	**RD_UPSTREAM_NUM_REQ**: The default value of DMA read request size is set by XUSB_DMA_REQ_SIZE parameter **0** = INIT

T_XUSB_CFG_FSBX_DMA_CTRL_0

Offset: 0x62c
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,x000,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
18:16	0x0	**WR_UPSTREAM_NUM_REQ**: The default value of DMA write request size is set by XUSB_DMA_REQ_SIZE parameter **0** = INIT
2:0	0x0	**RD_UPSTREAM_NUM_REQ**: The default value of DMA read request size is set by XUSB_DMA_REQ_SIZE parameter **0** = INIT

T_XUSB_CFG_SSPX_CORE_CNT7_0

Offset: 0x62c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00002ee0 (0bxxxx,xxxx,xxxx,0000,0010,1110,1110,0000)

Bit	Reset	Description
19:0	0x2ee0	**U2U3LP_HSHK_TBRST**: **12000** = INIT

T_XUSB_CFG_HSPX_CORE_CNT9_0

Offset: 0x62c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0003a980 (0bxx00,0000,0000,0011,1010,1001,1000,0000)

Bit	Reset	Description
29:0	0x3a980	**CON_NONSE0_MIN**: **240000** = INIT

T_XUSB_CFG_FSPX_CORE_CNT9_0

Offset: 0x62c
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00001f40 (0bxxxx,xxxx,xxx0,0000,0001,1111,0100,0000)

Bit	Reset	Description
20:0	0x1f40	**LS_PIPE_MIN_DLY**: **8000** = INIT

T_XUSB_CFG_SSBX_ASYNC_ERDY_ACCEL_0

Offset: 0x630
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000020 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,0000)

Bit	Reset	Description
31	0x0	**ENABLE**: **0** = INIT
5:0	0x20	**ERDY_QUEUE_FULL_COUNT**: **32** = INIT

T_XUSB_CFG_SSPX_CORE_CNT8_0

Offset: 0x630
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0001d4c0 (0bxx00,0000,0000,0001,1101,0100,1100,0000)

Bit	Reset	Description
29:0	0x1d4c0	**U0LNKCMD_TIMER**: **120000** = INIT

T_XUSB_CFG_HSPX_CORE_CNT10_0

Offset: 0x630
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000112 (0bx000,0000,0000,0000,0000,0001,0001,0010)

Bit	Reset	Description
30	0x0	**ADJUST_ADD**: **0** = INIT
29:22	0x0	**MAX_TIME_ADJUST**: **0** = INIT
21:0	0x112	**FSM_TIMEOUT_VAL**: **274** = INIT

T_XUSB_CFG_FSPX_CORE_CNT10_0

Offset: 0x630
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00001f40 (0bxxxx,xxxx,xxx0,0000,0001,1111,0100,0000)

Bit	Reset	Description
20:0	0x1f40	**LS_PIPE_MAX_DLY**: **8000** = INIT

T_XUSB_CFG_SSPX_CORE_CNT9_0

Offset: 0x634
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0003a980 (0bxx00,0000,0000,0011,1010,1001,1000,0000)

Bit	Reset	Description
29:0	0x3a980	**POLL_IDLE_TIMER**: **240000** = INIT

T_XUSB_CFG_HSPX_CORE_CNT11_0

Offset: 0x634
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x0000010c (0bxxxx,xxxx,xxx0,0000,0000,0001,0000,1100)

Bit	Reset	Description
20:0	0x10c	**HS_PIPE_MIN_DLY**: **268** = INIT

T_XUSB_CFG_FSPX_CORE_CNT11_0

Offset: 0x634
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000d20 (0bxx00,0000,0000,0000,0000,1101,0010,0000)

Bit	Reset	Description
29:0	0xd20	**U2_DEV_RESUMEK_MIN**: **3360** = INIT

T_XUSB_CFG_SSPX_CORE_CNT10_0

Offset: 0x638
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0003a980 (0bxx00,0000,0000,0011,1010,1001,1000,0000)

Bit	Reset	Description
29:0	0x3a980	**HRST_EXIT_TIMER**: **240000** = INIT

T_XUSB_CFG_HSPX_CORE_CNT12_0

Offset: 0x638
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000010c (0bxxxx,xxxx,xxx0,0000,0000,0001,0000,1100)

Bit	Reset	Description
20:0	0x10c	**HS_PIPE_MAX_DLY**: **268** = INIT

T_XUSB_CFG_FSPX_CORE_CNT12_0

Offset: 0x638
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000001e0 (0bxxxx,xxxx,xx00,0000,0000,0001,1110,0000)

Bit	Reset	Description
21:0	0x1e0	**U2_RESUME_RECOVERY**: **480** = INIT

T_XUSB_CFG_SSPX_CORE_CNT11_0

Offset: 0x63c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0003a980 (0bxx00,0000,0000,0011,1010,1001,1000,0000)

Bit	Reset	Description
29:0	0x3a980	**LPBK_EXIT_TIMER**: **240000** = INIT

T_XUSB_CFG_HSPX_CORE_CNT13_0

Offset: 0x63c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000020d0 (0bxx00,0000,0000,0000,0010,0000,1101,0000)

Bit	Reset	Description
29:0	0x20d0	**U2_DEV_RESUMEK_MIN**: **8400** = INIT

T_XUSB_CFG_FSPX_CORE_CNT13_0

Offset: 0x63c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000005a (0bxxxx,xxxx,xx00,0000,0000,0000,0101,1010)

Bit	Reset	Description
21:0	0x5a	**FS_LPM_TIMEOUT_VAL**: **90** = INIT

T_XUSB_CFG_SSBX_IDLE_0

Offset: 0x640
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000XXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
19	X	**THREAD3**
18	X	**THREAD2**
17	X	**THREAD1**
16	X	**THREAD0**
15	X	**ASYNC**
14	X	**PING_PROC**
13	X	**COMPL**
12	X	**UPSTREAM_WRITE_ARB**
11	X	**UPSTREAM_READ_ARB**
10	X	**DMA_WR**
9	X	**DMA_RD**
8	X	**DATABUF**
4	X	**EPFETCH_PING**
3	X	**EPFETCH_PRDCOUT**

Bit	Reset	Description
2	X	**EPFETCH_PRDCIN**
1	X	**EPFETCH_BULKOUT**
0	X	**EPFETCH_BULKIN**

T_XUSB_CFG_HSBX_IDLE_0

Offset: 0x640
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000XXX0X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
17	X	**THREAD1**
16	X	**THREAD0**
13	X	**COMPL**
12	X	**UPSTREAM_WRITE_ARB**
11	X	**UPSTREAM_READ_ARB**
10	X	**DMA_WR**
9	X	**DMA_RD**
8	X	**DATABUF**
2	X	**EPFETCH_SPLITINOUT**
1	X	**EPFETCH_PRDCINOUT**
0	X	**EPFETCH_BULKINOUT**

T_XUSB_CFG_FSBX_IDLE_0

Offset: 0x640
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000XXX0X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
17	X	**THREAD1**
16	X	**THREAD0**
13	X	**COMPL**
12	X	**UPSTREAM_WRITE_ARB**
11	X	**UPSTREAM_READ_ARB**
10	X	**DMA_WR**
9	X	**DMA_RD**
8	X	**DATABUF**
1	X	**EPFETCH_PRDCINOUT**
0	X	**EPFETCH_BULKINOUT**

T_XUSB_CFG_SSPX_CORE_CNT12_0

Offset: 0x640
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0003a980 (0bxx00,0000,0000,0011,1010,1001,1000,0000)

Bit	Reset	Description
29:0	0x3a980	**U1U2_NOLFPS_TIMER**: **240000** = INIT

T_XUSB_CFG_HSPX_CORE_CNT14_0

Offset: 0x640
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000004b0 (0bxxxx,xxxx,xx00,0000,0000,0100,1011,0000)

Bit	Reset	Description
21:0	0x4b0	**U2_RESUME_RECOVERY**: **1200** = INIT

T_XUSB_CFG_FSPX_CORE_CNT14_0

Offset: 0x640
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000253 (0bxxxx,xxxx,xx00,0000,0000,0010,0101,0011)

Bit	Reset	Description
21:0	0x253	**LS_LPM_TIMEOUT_VAL**: **595** = INIT

T_XUSB_CFG_SSBX_ASYNC_FIXED_EPLIST_0

Offset: 0x644
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**EN**: **0** = INIT

T_XUSB_CFG_HSBX_ASYNC_FIXED_EPLIST_0

Offset: 0x644
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**EN**: **0** = INIT

T_XUSB_CFG_FSBX_ASYNC_FIXED_EPLIST_0

Offset: 0x644
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**EN**: **0** = INIT

T_XUSB_CFG_SSPX_CORE_CNT13_0

Offset: 0x644
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000927c0 (0bxx00,0000,0000,1001,0010,0111,1100,0000)

Bit	Reset	Description
29:0	0x927c0	**CRDTHP_TIMER**: **600000** = INIT

T_XUSB_CFG_HSPX_CORE_CNT15_0

Offset: 0x644
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffc000f0 (0b1111,1111,1100,0000,0000,0000,1111,0000)

Bit	Reset	Description
31:22	0x3ff	**EOP_TIMEOUT**: **1023** = INIT
21:0	0xf0	**U3_RESUME_RECOVERY**: **240** = INIT

T_XUSB_CFG_FSPX_CORE_CNT15_0

Offset: 0x644
Read/Write: R/W

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00090048 (0bxxxx,xxxx,xxxx,1001,0000,0000,0100,1000)

Bit	Reset	Description
19:10	0x240	**LS_BUS_TIMEOUT_VAL**: **576** = INIT
9:0	0x48	**FS_BUS_TIMEOUT_VAL**: **72** = INIT

T_XUSB_CFG_SSPX_CORE_CNT14_0

Offset: 0x648
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000afc80 (0bxx00,0000,0000,1010,1111,1100,1000,0000)

Bit	Reset	Description
29:0	0xafc80	**UXEXIT_TIMER**: **720000** = INIT

T_XUSB_CFG_HSPX_CORE_CNT16_0

Offset: 0x648
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0007ef54 (0bxx00,0000,0000,0111,1110,1111,0101,0100)

Bit	Reset	Description
29:0	0x7ef54	**DISABLED_ENTRY_DELAY**: **520020** = INIT

T_XUSB_CFG_FSPX_CORE_NVWRAP_0

Offset: 0x648
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00530003 (0bxxxx,xxxx,0101,0011,xxxx,x000,0000,0011)

Bit	Reset	Description
23:20	0x5	**SE0_LS_WIDTH**: **5** = INIT
19:16	0x3	**SE0_FS_WIDTH**: **3** = INIT
10	0x0	**FS_SEND_EXTRA_J**: **0** = INIT
9:0	0x3	**FS_INTERPKT_DLY**: **3** = INIT

T_XUSB_CFG_SSPX_CORE_CNT15_0

Offset: 0x64c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000afc80 (0bxx00,0000,0000,1010,1111,1100,1000,0000)

Bit	Reset	Description
29:0	0xafc80	**RECOVCFG_TIMER**: **720000** = INIT

T_XUSB_CFG_HSPX_CORE_CNT17_0

Offset: 0x64c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x2f8000de (0b0010,1111,1000,0000,0000,0000,1101,1110)

Bit	Reset	Description
31:22	0xbe	**BUS_TIMEOUT_VAL**: **190** = INIT
21:0	0xde	**LPM_TIMEOUT_VAL**: **222** = INIT

T_XUSB_CFG_FSPX_CORE_ULPIWRAP_0

Offset: 0x64c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0001900f (0bxxxx,xx0x,xxxx,0001,1001,0000,0000,1111)

Bit	Reset	Description
25	0x0	**FS_U3_TO_U0_LNST_CHK_MODE**: **0** = INIT
19:10	0x64	**LS_INTERPKT_DLY**: **100** = INIT
9:0	0xf	**FS_INTERPKT_DLY**: **15** = INIT

T_XUSB_CFG_SSPX_CORE_CNT16_0

Offset: 0x650
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00124f80 (0bxx00,0000,0001,0010,0100,1111,1000,0000)

Bit	Reset	Description
29:0	0x124f80	**U3NOLFPS_TIMER**: **1200000** = INIT

T_XUSB_CFG_HSPX_CORE_NVWRAP_0

Offset: 0x650
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00004c60 (0bxxxx,xxxx,xxxx,xxxx,0100,1100,0110,0000)

Bit	Reset	Description
15:11	0x9	**HS_DISCON_DLY**: **9** = INIT
10	0x1	**NOBROADCAST**: **1** = INIT

Bit	Reset	Description
9:0	0x60	**HS_INTERPKT_DLY**: **96** = INIT

T_XUSB_CFG_SSBX_EPFETCH_STATUS_1_EPTYPE_0

This is an array of five identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_CFG_SSBX_EPFETCH_STATUS_1_EPTYPE_<i>, among which <i> belongs to <0..4>.
Offset: 0x650,..,0x690
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000XXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
10	X	**DB_PENDING**
9:6	X	**VF_ID**
5	X	**ELM**
4:0	X	**FSM_STATE**

T_XUSB_CFG_HSBX_EPFETCH_STATUS_1_EPTYPE_0

This is an array of three identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_CFG_HSBX_EPFETCH_STATUS_1_EPTYPE_<i>, among which <i> belongs to <0..2>.
Offset: 0x650,..,0x670
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000XXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
10	X	**DB_PENDING**
9:6	X	**VF_ID**
5	X	**ELM**
4:0	X	**FSM_STATE**

T_XUSB_CFG_FSBX_EPFETCH_STATUS_1_EPTYPE_0

This is an array of two identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_CFG_FSBX_EPFETCH_STATUS_1_EPTYPE_<i>, among which <i> belongs to <0..1>.
Offset: 0x650,0x660
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000XXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
10	X	**DB_PENDING**
9:6	X	**VF_ID**
5	X	**ELM**
4:0	X	**FSM_STATE**

T_XUSB_CFG_SSPX_CORE_MIN_TX3_0

Offset: 0x654
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00080000 (0bxxxx,xxxx,0000,1000,0000,0000,0000,0000)

Bit	Reset	Description
23:0	0x80000	**GEN2_TSEQ_CNT**: **524288** = INIT

T_XUSB_CFG_HSPX_CORE_ULPIWRAP_0

Offset: 0x654
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00606414 (0bxxxx,xx00,0110,0000,0110,0100,0001,0100)

Bit	Reset	Description
25	0x0	**HS_U3_TO_U0_LNST_CHK_MODE**: **0** = INIT
24:21	0x3	**HS_SOF_DLY_VAL**: **3** = INIT
20	0x0	**HS_SOF_DLY_EN**: **0** = INIT
19:10	0x19	**HS_SOF_INTERPKT_DLY**: **25** = INIT
9:0	0x14	**HS_INTERPKT_DLY**: **20** = INIT

T_XUSB_CFG_SSBX_EPFETCH_STATUS_2_EPTYPE_0

This is an array of five identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_CFG_SSBX_EPFETCH_STATUS_2_EPTYPE_<i>, among which <i> belongs to <0..4>.
Offset: 0x654,..,0x694
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**EP_PTRLO**

T_XUSB_CFG_HSBX_EPFETCH_STATUS_2_EPTYPE_0

This is an array of three identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_CFG_HSBX_EPFETCH_STATUS_2_EPTYPE_<i>, among which <i> belongs to <0..2>.
Offset: 0x654,..,0x674
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**EP_PTRLO**

T_XUSB_CFG_FSBX_EPFETCH_STATUS_2_EPTYPE_0

This is an array of two identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_CFG_FSBX_EPFETCH_STATUS_2_EPTYPE_<i>, among which <i> belongs to <0..1>.
Offset: 0x654,0x664
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**EP_PTRLO**

T_XUSB_CFG_SSPX_CORE_CNT18_0

Offset: 0x658
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0015f900 (0bxx00,0000,0001,0101,1111,1001,0000,0000)

Bit	Reset	Description
29:0	0x15f900	**POLL_ACT_TIMER**: **1440000** = INIT

T_XUSB_CFG_SSBX_EPFETCH_STATUS_3_EPTYPE_0

This is an array of five identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_CFG_SSBX_EPFETCH_STATUS_3_EPTYPE_<i>, among which <i> belongs to <0..4>.
Offset: 0x658,..,0x698
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00XXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
21:8	X	**CHECK_UFRAME**

Bit	Reset	Description
7:0	X	**EP_PTRHI**

T_XUSB_CFG_HSBX_EPFETCH_STATUS_3_EPTYPE_0

This is an array of three identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_CFG_HSBX_EPFETCH_STATUS_3_EPTYPE_<i>, among which <i> belongs to <0..2>.
Offset: 0x658,..,0x678
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00XXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
21:8	X	**CHECK_UFRAME**
7:0	X	**EP_PTRHI**

T_XUSB_CFG_FSBX_EPFETCH_STATUS_3_EPTYPE_0

This is an array of two identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_CFG_FSBX_EPFETCH_STATUS_3_EPTYPE_<i>, among which <i> belongs to <0..1>.
Offset: 0x658,0x668
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00XXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
21:8	X	**CHECK_UFRAME**
7:0	X	**EP_PTRHI**

T_XUSB_CFG_SSPX_CORE_CNT19_0

Offset: 0x65c
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x0015f900 (0bxx00,0000,0001,0101,1111,1001,0000,0000)

Bit	Reset	Description
29:0	0x15f900	**POLL_CFG_TIMER**: **1440000** = INIT

T_XUSB_CFG_HSPX_CORE_NVWRAP_DESER_0

Offset: 0x65c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x01568c03 (0bxxxx,xxx1,0101,0110,1000,1100,0000,0011)

Bit	Reset	Description
24	0x1	**USECOMMONUPDNCNT**: **1** = INIT
23	0x0	**THREECYCLEEDGEDETCFG**: **0** = INIT
22	0x1	**USEWEIGHTEDEDGESCFG**: **1** = INIT
21	0x0	**HSRXLATESQUELCHCFG**: **0** = INIT
20:19	0x2	**KEEPPATTERNONACTIVECFG**: **2** = INIT
18	0x1	**ALLOWCONSECUTIVEUPDOWNCFG**: **1** = INIT
17	0x1	**REALIGNONNEWPACKETCFG**: **1** = INIT
16:13	0x4	**PCOUNTUPDOWNDIVCFG**: **4** = INIT
12:10	0x3	**SQUELCHEOPDELAYCFG**: **3** = INIT
9	0x0	**PASSCHIRPCFG**: **0** = INIT
8	0x0	**PASSFEEDBACKCFG**: **0** = INIT
7:6	0x0	**PCOUNTINERTIACFG**: **0** = INIT
5:4	0x0	**PHASEADJUSTCFG**: **0** = INIT

Bit	Reset	Description
3	0x0	**THREESYNCBITSCFG**: **0** = INIT
2	0x0	**USE4SYNCTRANCFG**: **0** = INIT
1	0x1	**T210_A02_MODE**: **1** = INIT
0	0x1	**MCP_MODE**: **1** = INIT

T_XUSB_CFG_SSPX_CORE_CNT20_0

Offset: 0x660
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0015f900 (0bxx00,0000,0001,0101,1111,1001,0000,0000)

Bit	Reset	Description
29:0	0x15f900	**RCOV_ACT_TIMER**: **1440000** = INIT

T_XUSB_CFG_SSPX_CORE_CNT21_0

Offset: 0x664
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0003a980 (0bxx00,0000,0000,0011,1010,1001,1000,0000)

Bit	Reset	Description
29:0	0x3a980	**RCOV_IDL_TIMER**: **240000** = INIT

T_XUSB_CFG_SSPX_CORE_CNT22_0

Offset: 0x668
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x0000fff0 (0bxxxx,xxxx,xxxx,xxxx,1111,1111,1111,0000)

Bit	Reset	Description
15:0	0xfff0	**CP9_RPT_CNT**: **65520** = INIT

T_XUSB_CFG_SSPX_CORE_CNT23_0

Offset: 0x66c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0015f900 (0bxx00,0000,0001,0101,1111,1001,0000,0000)

Bit	Reset	Description
29:0	0x15f900	**HRST_ACT_TIMER**: **1440000** = INIT

T_XUSB_CFG_SSPX_CORE_CNT24_0

Offset: 0x670
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00b71b00 (0bxx00,0000,1011,0111,0001,1011,0000,0000)

Bit	Reset	Description
29:0	0xb71b00	**WARMRST_TBRST**: **12000000** = INIT

T_XUSB_CFG_SSPX_CORE_CNT25_0

Offset: 0x674
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00b71b00 (0bxx00,0000,1011,0111,0001,1011,0000,0000)

Bit	Reset	Description
29:0	0xb71b00	**U3LFPS_RTY_TIMER**: **12000000** = INIT

T_XUSB_CFG_SSPX_CORE_CNT26_0

Offset: 0x678
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00b71b00 (0bxx00,0000,1011,0111,0001,1011,0000,0000)

Bit	Reset	Description
29:0	0xb71b00	**UXDET_RCV_TIMER**: **12000000** = INIT

T_XUSB_CFG_SSPX_CORE_CNT27_0

Offset: 0x67c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x016e3600 (0bxx00,0001,0110,1110,0011,0110,0000,0000)

Bit	Reset	Description
29:0	0x16e3600	**PING_LFPS_TRPT**: **24000000** = INIT

T_XUSB_CFG_SSPX_CORE_CNT28_0

Offset: 0x680
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x02255100 (0bxx00,0010,0010,0101,0101,0001,0000,0000)

Bit	Reset	Description
29:0	0x2255100	**U1_PING_TIMER**: **36000000** = INIT

T_XUSB_CFG_SSPX_CORE_CNT29_0

Offset: 0x684
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x02932e00 (0bxx00,0010,1001,0011,0010,1110,0000,0000)

Bit	Reset	Description
29:0	0x2932e00	**POLL_LFPS_TIMER**: **43200000** = INIT

T_XUSB_CFG_SSPX_CORE_CNT30_0

Offset: 0x688
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000960 (0bxxxx,xxxx,xxxx,0000,0000,1001,0110,0000)

Bit	Reset	Description
19:0	0x960	**LMPITP_TIMER**: **2400** = INIT

T_XUSB_CFG_SSPX_CORE_CNT31_0

Offset: 0x68c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x48252304 (0b0100,1000,0010,0101,0010,0011,0000,0100)

Bit	Reset	Description
31:24	0x48	**POLL_TBRST_MIN**: **72** = INIT
23:16	0x25	**U1EXIT_TBRST_MIN**: **37** = INIT
15:8	0x23	**PING_TBRST_MAX**: **35** = INIT

Bit	Reset	Description
7:0	0x4	**PING_TBRST_MIN**: **4** = INIT

T_XUSB_CFG_SSPX_CORE_CNT32_0

Offset: 0x690
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000168a9 (0bxxxx,0000,0000,0001,0110,1000,1010,1001)

Bit	Reset	Description
27:8	0x168	**UX_RESIDENCY**: **360** = INIT
7:0	0xa9	**POLL_TBRST_MAX**: **169** = INIT

T_XUSB_CFG_SSPX_CORE_CNT33_0

Offset: 0x694
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000260 (0bxxxx,xxxx,xxxx,0000,0000,0010,0110,0000)

Bit	Reset	Description
19:0	0x260	**POLL_TRPT_MIN**: **608** = INIT

T_XUSB_CFG_SSPX_CORE_CNT34_0

Offset: 0x698
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000006d6 (0bxxxx,xxxx,xxxx,0000,0000,0110,1101,0110)

Bit	Reset	Description
19:0	0x6d6	**POLL_TRPT_MAX**: **1750** = INIT

T_XUSB_CFG_SSPX_CORE_CNT35_0

Offset: 0x69c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000025 (0bxxxx,xxxx,xxxx,0000,0000,0000,0010,0101)

Bit	Reset	Description
19:0	0x25	**U1U2LBEXIT_TBRST_MIN**: **37** = INIT

T_XUSB_CFG_SSPX_CORE_CNT36_0

Offset: 0x6a0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00927d80 (0bxx00,0000,1001,0010,0111,1101,1000,0000)

Bit	Reset	Description
29:0	0x927d80	**TRST_TBRST_MIN**: **9600384** = INIT

T_XUSB_CFG_SSPX_CORE_CNT37_0

Offset: 0x6a4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29:0	0x0	**IDDQ_TWAIT**: **0** = INIT

T_XUSB_CFG_SSPX_CORE_CNT38_0

Offset: 0x6a8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00002710 (0bxxxx,xxxx,xxxx,0000,0010,0111,0001,0000)

Bit	Reset	Description
19:0	0x2710	**U2U3EXIT_TBRSTMIN**: **10000** = INIT

T_XUSB_CFG_SSPX_CORE_CTRL1_0

Offset: 0x6ac
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x60880004 (0b0110,0000,1000,1000,0000,0000,0000,0100)

Bit	Reset	Description
31:26	0x18	**LPBK_SKIP_HIGH**: **24** = INIT
25:20	0x8	**LPBK_SKIP_LOW**: **8** = INIT
19:16	0x8	**MAX_DETECT_ATTEMPT**: **8** = INIT
15:0	0x4	**MINTXTS2_RSTCLR_CNT**: **4** = INIT

T_XUSB_CFG_SSPX_CORE_CTRL2_0

Offset: 0x6b0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x44900001 (0b0100,0100,1001,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
31:29	0x2	**TSTATUS_WAIT_TIMEOUT_ADJ**: **2** = INIT
28:26	0x1	**U3_RCVDET_TIMER_DIV_VAL**: **0** = _0P5X **1** = _1X **2** = _2X **3** = _4X **4** = _8X **5** = _10X
25:23	0x1	**INACT_RCVDET_TIMER_DIV_VAL**: **0** = _0P5X **1** = _1X **2** = _2X **3** = _4X **4** = _8X **5** = _10X
22:20	0x1	**RXDET_RCVDET_TIMER_DIV_VAL**: **0** = _0P5X **1** = _1X **2** = _2X **3** = _4X **4** = _8X **5** = _10X
0	0x1	**CYA_POLLING_IGNORE_SCDTO**: **0** = FALSE **1** = TRUE

T_XUSB_CFG_SSBX_STATUS1_THREAD_0

This is an array of four identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_CFG_SSBX_STATUS1_THREAD_<i>, among which <i> belongs to <0..3>.
Offset: 0x6b0,..,0x6c8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:21	X	**EP_VF_ID**
20:18	X	**EP_TYPE**
17:13	X	**DCI**
12:5	X	**SLOT_ID**
4:0	X	**CTRL_FSM**

T_XUSB_CFG_HSBX_STATUS1_THREAD_0

This is an array of two identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_CFG_HSBX_STATUS1_THREAD_<i>, among which <i> belongs to <0..1>.
Offset: 0x6b0,0x6b8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:21	X	**EP_VF_ID**
20:18	X	**EP_TYPE**
17:13	X	**DCI**
12:5	X	**SLOT_ID**
4:0	X	**CTRL_FSM**

T_XUSB_CFG_FSBX_STATUS1_THREAD_0

This is an array of two identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_CFG_FSBX_STATUS1_THREAD_<i>, among which <i> belongs to <0..1>.
Offset: 0x6b0,0x6b8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
24:21	X	**EP_VF_ID**
20:18	X	**EP_TYPE**
17:13	X	**DCI**
12:5	X	**SLOT_ID**
4:0	X	**CTRL_FSM**

T_XUSB_CFG_SSPX_CORE_CNT40_0

Offset: 0x6b4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x007aaf82 (0bxx00,0000,0111,1010,1010,1111,1000,0010)

Bit	Reset	Description
29:0	0x7aaf82	**TRST_TBRST_FROM_U0_MIN**: **8040322** = INIT

T_XUSB_CFG_SSBX_STATUS2_THREAD_0

This is an array of four identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_CFG_SSBX_STATUS2_THREAD_<i>, among which <i> belongs to <0..3>.
Offset: 0x6b4,..,0x6cc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00XXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
21:14	X	**WORKQ_TYPE**
13:0	X	**CHECK_UFRAME**

T_XUSB_CFG_HSBX_STATUS2_THREAD_0

This is an array of two identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_CFG_HSBX_STATUS2_THREAD_<i>, among which <i> belongs to <0..1>.
Offset: 0x6b4,0x6bc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00XXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
21:14	X	**WORKQ_TYPE**
13:0	X	**CHECK_UFRAME**

T_XUSB_CFG_FSBX_STATUS2_THREAD_0

This is an array of two identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_CFG_FSBX_STATUS2_THREAD_<i>, among which <i> belongs to <0..1>.
Offset: 0x6b4,0x6bc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00XXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
21:14	X	**WORKQ_TYPE**
13:0	X	**CHECK_UFRAME**

T_XUSB_CFG_SSPX_CORE_CTRL3_0

Offset: 0x6b8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x001f3fff (0b0000,0000,0001,1111,0011,1111,1111,1111)

Bit	Reset	Description
31:16	0x1f	**TS1_TS2_SYNC_INTVL**: **31** = INIT
15:0	0x3fff	**TSEQ_SYNC_INTVL**: **16383** = INIT

T_XUSB_CFG_SSPX_CORE_CTRL4_0

Offset: 0x6bc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x03030404 (0b00xx,xx11,xxxx,x011,0xxx,0100,xxxx,0100)

Bit	Reset	Description
31	0x0	**DEV_PORT_REJECT_U1_ENTRY**: **0** = INIT
30	0x0	**DEV_PORT_REJECT_U2_ENTRY**: **0** = INIT

Bit	Reset	Description
25	0x1	**TXHBUF_HOLD_ROLLBACK_LGOOD_PENDING**: **1** = INIT
24	0x1	**ACCU_BUF_G1_WMARK_SEL**: **1** = INIT
18:15	0x6	**LINK2PHY_G1_TXBUF_FIFO_WR_LIMIT**: **6** = INIT
11:8	0x4	**LINK2PHY_TX_PIPE_DLY**: **4** = INIT
3:0	0x4	**PHY2LINK_RX_PIPE_DLY**: **4** = INIT

T_XUSB_CFG_SSPX_CORE_CNT42_0

Offset: 0x6c0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00a08c14 (0bxxxx,xxxx,1010,0000,1000,1100,0001,0100)

Bit	Reset	Description
23:16	0xa0	**SLFPS_U1EX_TBRSTMIN**: **160** = INIT
15:8	0x8c	**SLFPS_PING_TBRSTMAX**: **140** = INIT
7:0	0x14	**SLFPS_PING_TBRSTMIN**: **20** = INIT

T_XUSB_CFG_SSPX_CORE_CNT43_0

Offset: 0x6c4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x1f030000 (0b0001,1111,0000,0011,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	0x1f	**MAX_LFPS_HALFCYCLES**: **31** = INIT

Bit	Reset	Description
23:16	0x3	**MIN_LFPS_HALFCYCLES**: **3** = INIT

T_XUSB_CFG_SSPX_CORE_PHY_G2_CTRL1_0

Offset: 0x6c8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x90028608 (0b1xx1,0000,000x,0010,1000,0110,0000,1000)

Bit	Reset	Description
31	0x1	**TX_INVAL_CRC_IN_PNULL_DPP**: **1** = INIT
28:21	0x80	**TX_SKP_INS_THRES_TSEQ**: **128** = INIT
19:12	0x28	**TX_SKP_INS_THRES**: **40** = INIT
11:6	0x18	**LPBK_SKP_THRES_HI**: **24** = INIT
5:0	0x8	**LPBK_SKP_THRES_LO**: **8** = INIT

T_XUSB_CFG_SSPX_CORE_PHY_G2_CTRL2_0

Offset: 0x6cc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x002941ff (0bxxxx,xxxx,x010,1001,0100,0001,1111,1111)

Bit	Reset	Description
22:19	0x5	**PHY_TX_CG_HYST**: **5** = INIT
18:15	0x2	**TXBUF_POP_DELAY**: **2** = INIT
14:10	0x10	**NUM_EXTRA_IDLES**: **16** = INIT

Bit	Reset	Description
9:0	0x1ff	**DC_BALANCE_CNT_MAX**: **511** = INIT

T_XUSB_CFG_SSPX_CORE_CTRL5_0

Offset: 0x6d0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x02020410 (0bx000,0010,x000,0010,x000,0100,x001,0000)

Bit	Reset	Description
30:24	0x2	**POLL_MINRXLFPS**: **2** = INIT
22:16	0x2	**PING_MINTXLFPS**: **2** = INIT
14:8	0x4	**POLL_MINTXLFPS_RCV**: **4** = INIT
6:0	0x10	**POLL_MINTXLFPS**: **16** = INIT

T_XUSB_CFG_SSPX_CORE_UPHYCTL_RX7_0

Offset: 0x6d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x02000004 (0b0xxx,0010,xxxx,xxxx,xxxx,xxxx,xxxx,0100)

Bit	Reset	Description
31	0x0	**IDLE_STATUS_GLITCH_STRETCH_DISABLE**: **0** = INIT
27:24	0x2	**ASYNC_RX_WATERMARK**: **2** = INIT
3:0	0x4	**IDLE_STATUS_GLITCH_CNT_THRESHOLD**: **4** = INIT

T_XUSB_CFG_SSPX_CORE_CNT48_0

Offset: 0x6d8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000096 (0bxxxx,xxxx,xxxx,0000,0000,0000,1001,0110)

Bit	Reset	Description
19:0	0x96	**SLFPS_U2U3LBEX_TBRSTMIN**: **150** = INIT

T_XUSB_CFG_SSPX_CORE_CNT49_0

Offset: 0x6dc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x02faf080 (0bxx00,0010,1111,1010,1111,0000,1000,0000)

Bit	Reset	Description
29:0	0x2faf080	**SLFPS_TRST_TBRSTMIN**: **50000000** = INIT

T_XUSB_CFG_SSPX_CORE_ITP_0

Offset: 0x6e0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000001e0 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0001,1110,0000)

Bit	Reset	Description
12:0	0x1e0	**DELTA_THRES**: **480** = INIT

T_XUSB_CFG_SSPX_CORE_CNT50_0

Offset: 0x6e4
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00009624 (0bxxxx,xxxx,xxxx,xxxx,1001,0110,0010,0100)

Bit	Reset	Description
15:8	0x96	**SLFPS_U1_TBRSTMIN**: **150** = INIT
7:0	0x24	**PLFPS_U1_TBRSTMIN**: **36** = INIT

T_XUSB_CFG_SSPX_CORE_CNT51_0

Offset: 0x6e8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000025 (0bxxxx,xxxx,xxxx,0000,0000,0000,0010,0101)

Bit	Reset	Description
19:0	0x25	**PLFPS_U2U3LB_TBRSTMIN**: **37** = INIT

T_XUSB_CFG_SSPX_CORE_CNT52_0

Offset: 0x6ec
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000096 (0bxxxx,xxxx,xxxx,0000,0000,0000,1001,0110)

Bit	Reset	Description
19:0	0x96	**SLFPS_U2U3LB_TBRSTMIN**: **150** = INIT

T_XUSB_CFG_SSPX_CORE_CNT53_0

Offset: 0x6f0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00603840 (0bxxxx,0000,0110,0000,0011,1000,0100,0000)

Bit	Reset	Description
27:20	0x6	**MIN_RXEQ_TRAIN_EN_TIME**: This register configures the minimum time from LFPS end to RX data en assertion **6** = INIT
19:0	0x3840	**RXEQ_WAIT_CNT**: The minimum wait time between polling.LFPS to polling.RXEQ transition. Here we first wait for the LFPS to end. Once LFPS is ended, wait for this minimum time before rx_data_en is asserted. This is to prevent CDR lock failure (CDR should lock on HS data not on LFPS). **14400** = INIT

T_XUSB_CFG_SSPX_CORE_CNT54_0

Offset: 0x6f4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00600e10 (0bxxxx,0000,0110,0000,0000,1110,0001,0000)

Bit	Reset	Description
27:20	0x6	**RXEQ_RESET_TIME**: **6** = INIT
19:0	0xe10	**RXEQ_LOCK_CNT**: **3600** = INIT

T_XUSB_CFG_SSPX_CORE_CNT55_0

Offset: 0x6f8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x04140025 (0b0000,0100,0001,0100,xxxx,xxx0,0010,0101)

Bit	Reset	Description
31:24	0x4	**MIN_ELEC_IDLE_CNT_DIS_TO_EN**: **4** = INIT
23:16	0x14	**MIN_ELEC_IDLE_CNT_EN_TO_DIS**: **20** = INIT
8	0x0	**LATCH_U2_EXIT_LFPS_EN**: **0** = INIT

Bit	Reset	Description
7:0	0x25	**PING_LFPS_FILTER_TIMEOUT**: **37** = INIT

T_XUSB_CFG_SSPX_CORE_CNT56_0

Offset: 0x6fc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x01000438 (0bxxxx,x001,0000,0000,0000,0100,0011,1000)

Bit	Reset	Description
26:20	0x10	**SSPLUS_FAIL_POLL_BRST_CNT**: **16** = INIT
19:0	0x438	**SCD_BIT0_TRPT_MAX**: **1080** = INIT

T_XUSB_CFG_SSPX_CORE_CNT57_0

Offset: 0x700
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00100528 (0bxxxx,x000,0001,0000,0000,0101,0010,1000)

Bit	Reset	Description
26:20	0x1	**MIN_TX_SCD_CNT_PASS**: **1** = INIT
19:0	0x528	**SCD_BIT1_TRPT_MIN**: **1320** = INIT

T_XUSB_CFG_SSPX_CORE_CNT58_0

Offset: 0x704
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00100d20 (0bxxxx,x000,0001,0000,0000,1101,0010,0000)

Bit	Reset	Description
26:20	0x1	**MIN_SCD_CNT_POST_RX_FAIL**: **1** = INIT
19:0	0xd20	**SCD_END_TRPT_MIN**: **3360** = INIT

T_XUSB_CFG_SSPX_CORE_CNT59_0

Offset: 0x708
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00200036 (0bxxxx,x000,0010,0000,0000,0000,0011,0110)

Bit	Reset	Description
26:20	0x2	**MIN_RX_LBPM_CNT**: **2** = INIT
19:0	0x36	**LBPM_BIT0_TBRST_MIN**: **54** = INIT

T_XUSB_CFG_SSPX_CORE_CNT60_0

Offset: 0x70c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00100066 (0bxxxx,x000,0001,0000,0000,0000,0110,0110)

Bit	Reset	Description
26:20	0x1	**MIN_TX_LBPM_CNT**: **1** = INIT
19:0	0x66	**LBPM_BIT0_TBRST_MAX**: **102** = INIT

T_XUSB_CFG_SSPX_CORE_CNT61_0

Offset: 0x710
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x0040009a (0bxxxx,x000,0100,0000,0000,0000,1001,1010)

Bit	Reset	Description
26:20	0x4	**MIN_TX_LBPM_CNT_POST_RX**: **4** = INIT
19:0	0x9a	**LBPM_BIT1_TBRST_MIN**: **154** = INIT

T_XUSB_CFG_SSPX_CORE_CNT62_0

Offset: 0x714
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x002000de (0bxxxx,x000,0010,0000,0000,0000,1101,1110)

Bit	Reset	Description
26:20	0x2	**MIN_SCD_CNT_POST_RX_PASS**: **2** = INIT
19:0	0xde	**LBPM_BIT1_TBRST_MAX**: **222** = INIT

T_XUSB_CFG_SSPX_CORE_CNT63_0

Offset: 0x718
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x001000f0 (0bxxxx,x000,0001,0000,0000,0000,1111,0000)

Bit	Reset	Description
26:20	0x1	**MIN_TX_SCD_CNT_POLL**: **1** = INIT
19:0	0xf0	**LBPM_TPWM_MIN**: **240** = INIT

T_XUSB_CFG_SSPX_CORE_CNT64_0

Offset: 0x71c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00400120 (0bxxxx,x000,0100,0000,0000,0001,0010,0000)

Bit	Reset	Description
26:20	0x4	**MIN_TX_SCD_CNT_FAIL**: **4** = INIT
19:0	0x120	**LBPM_TPWM_MAX**: **288** = INIT

T_XUSB_CFG_SSPX_CORE_CNT65_0

Offset: 0x720
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00100e10 (0bxxxx,x000,0001,0000,0000,1110,0001,0000)

Bit	Reset	Description
26:20	0x1	**SSP_MIN_TX_POLL_LFPS**: **1** = INIT
19:0	0xe10	**TX_SCD_END_TRPT_MID**: **3600** = INIT

T_XUSB_CFG_SSPX_CORE_CNT66_0

Offset: 0x724
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x01400348 (0bxxxx,x001,0100,0000,0000,0011,0100,1000)

Bit	Reset	Description
26:20	0x14	**SSP_MIN_TX_POLL_LFPS_PLUS**: **20** = INIT
19:0	0x348	**TX_SCD_BIT0_TRPT_MID**: **840** = INIT

T_XUSB_CFG_SSPX_CORE_CNT67_0

Offset: 0x728
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x040005a0 (0bxxxx,x100,0000,0000,0000,0101,1010,0000)

Bit	Reset	Description
26:20	0x40	**POLL_LFPS_PLUS_FAIL_CNT**: **64** = INIT
19:0	0x5a0	**TX_SCD_BIT1_TRPT_MID**: **1440** = INIT

T_XUSB_CFG_SSPX_CORE_CNT68_0

Offset: 0x72c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000004e (0bxxxx,xxxx,xxxx,0000,0000,0000,0100,1110)

Bit	Reset	Description
19:0	0x4e	**TX_LBPM_BIT0_TBRST_MID**: **78** = INIT

T_XUSB_CFG_SSPX_CORE_CNT69_0

Offset: 0x730
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000000b4 (0bxxxx,xxxx,xxxx,0000,0000,0000,1011,0100)

Bit	Reset	Description
19:0	0xb4	**TX_LBPM_BIT1_TBRST_MID**: **180** = INIT

T_XUSB_CFG_SSPX_CORE_CNT70_0

Offset: 0x734
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000108 (0bxxxx,xxxx,xxxx,0000,0000,0001,0000,1000)

Bit	Reset	Description
19:0	0x108	**TX_LBPM_TPWM_MID**: **264** = INIT

T_XUSB_CFG_SSPX_CORE_CNT71_0

Offset: 0x738
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000108 (0bxxxx,xxxx,xxxx,0000,0000,0001,0000,1000)

Bit	Reset	Description
19:0	0x108	**LBPM_END_MIN_EIDLE**: **264** = INIT

T_XUSB_CFG_SSPX_CORE_CNT72_0

Offset: 0x73c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00001c21 (0bxxxx,xxxx,xxxx,0000,0001,1100,0010,0001)

Bit	Reset	Description
19:0	0x1c21	**SCD_LFPS_TIMEOUT**: **7201** = INIT

T_XUSB_CFG_SSPX_CORE_CNT73_0

Offset: 0x740
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x0015f93a (0bxx00,0000,0001,0101,1111,1001,0011,1010)

Bit	Reset	Description
29:0	0x15f93a	**POLL_LBPM_TIMEOUT**: **1440058** = INIT

T_XUSB_CFG_SSPX_CORE_UPHYCTL_0

Offset: 0x744
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000015bf (0bxxxx,xxxx,xxxx,xxxx,xxx1,0101,1011,1111)

Bit	Reset	Description
12	0x1	**CAL_COMPLIANCE**: **1** = INIT
11	0x0	**EQ_LOOPBACK_PERIODIC**: **0** = INIT
10	0x1	**EQ_U0_PERIODIC**: **1** = INIT
9	0x0	**EQ_RESET_U3EXIT**: **0** = INIT
8	0x1	**EQ_RESET_POLLING**: **1** = INIT
7	0x1	**CAL_U2EXIT**: **1** = INIT
6	0x0	**CAL_U1EXIT**: **0** = INIT
5	0x1	**EOM_U3EXIT**: **1** = INIT
4	0x1	**EOM_POLLING**: **1** = INIT
3	0x1	**EQ_U3EXIT**: **1** = INIT
2	0x1	**EQ_POLLING**: **1** = INIT
1	0x1	**CAL_U3EXIT**: **1** = INIT
0	0x1	**CAL_POLLING**: **1** = INIT

T_XUSB_CFG_SSPX_CORE_UPHYCTL_TX1_0

Offset: 0x748
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x07000aa3 (0b0000,0111,xxxx,xxxx,xxxx,1010,1010,0011)

Bit	Reset	Description
31:24	0x7	**CAL_DATAEN**: **7** = INIT
11:10	0x2	**SLPSTATE_RATE_CHANGE**: **2** = INIT
9:8	0x2	**SLPSTATE_U3**: **2** = INIT
7:6	0x2	**SLPSTATE_U2**: **2** = INIT
5:4	0x2	**SLPSTATE_U1**: **2** = INIT
3:2	0x0	**SLPSTATE_LFPS**: **0** = INIT
1:0	0x3	**SLPSTATE_DETECT**: **3** = INIT

T_XUSB_CFG_SSPX_CORE_UPHYCTL_TX2_0

Offset: 0x74c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x070d004e (0b0000,0111,0000,1101,xxxx,0000,0100,1110)

Bit	Reset	Description
31:24	0x7	**RATE_SLEEP**: **7** = INIT
23:16	0xd	**SLEEP_MIN**: **13** = INIT
11:0	0x4e	**SLEEP0_DATAEN**: **78** = INIT

T_XUSB_CFG_SSPX_CORE_UPHYCTL_TX3_0

Offset: 0x750
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x1f134e00 (0b0001,1111,0001,0011,0100,1110,xxxx,xxxx)

Bit	Reset	Description
31:24	0x1f	**SLEEP_RATE**: **31** = INIT
23:16	0x13	**SLEEP_BYPEN**: **19** = INIT
15:8	0x4e	**SLEEP_CAL**: **78** = INIT

T_XUSB_CFG_SSPX_CORE_UPHYCTL_RX1_0

Offset: 0x754
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00070aa3 (0b0000,0000,xxxx,x111,xx0x,1010,1010,0011)

Bit	Reset	Description
31:24	0x0	**CAL_DATAEN**: **0** = INIT
18:16	0x7	**U3_IDLE_DETECT_MOD_TH**: **7** = INIT
13	0x0	**WAIT_RX_EQ_BYPASS**: **0** = INIT
11:10	0x2	**SLPSTATE_RATE_CHANGE**: **2** = INIT
9:8	0x2	**SLPSTATE_U3**: **2** = INIT
7:6	0x2	**SLPSTATE_U2**: **2** = INIT
5:4	0x2	**SLPSTATE_U1**: **2** = INIT

Bit	Reset	Description
3:2	0x0	**SLPSTATE_LFPS**: **0** = INIT
1:0	0x3	**SLPSTATE_DETECT**: **3** = INIT

T_XUSB_CFG_SSPX_CORE_UPHYCTL_RX2_0

Offset: 0x758
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x070d004e (0b0000,0111,0000,1101,xxxx,0000,0100,1110)

Bit	Reset	Description
31:24	0x7	**RATE_SLEEP**: **7** = INIT
23:16	0xd	**SLEEP_MIN**: **13** = INIT
11:0	0x4e	**SLEEP0_DATAEN**: **78** = INIT

T_XUSB_CFG_SSPX_CORE_UPHYCTL_RX3_0

Offset: 0x75c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x1f79794e (0b0001,1111,0111,1001,0111,1001,0100,1110)

Bit	Reset	Description
31:24	0x1f	**SLEEP_RATE**: **31** = INIT
23:16	0x79	**TCAL_ABORT**: **121** = INIT
15:8	0x79	**TEQ_ABORT**: **121** = INIT
7:0	0x4e	**SLEEP_CAL**: **78** = INIT

T_XUSB_CFG_SSPX_CORE_XFER_CNT0_0

Offset: 0x760
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0001d4c0 (0b0000,0000,0000,0001,1101,0100,1100,0000)

Bit	Reset	Description
31:0	0x1d4c0	**RXACK**: **120000** = INIT

T_XUSB_CFG_SSPX_CORE_XFER_CNT1_0

Offset: 0x764
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0001d4c0 (0b0000,0000,0000,0001,1101,0100,1100,0000)

Bit	Reset	Description
31:0	0x1d4c0	**RXDP**: **120000** = INIT

T_XUSB_CFG_SSPX_CORE_INXFER_0

Offset: 0x768
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x009c0843 (0b0000,0000,1001,1100,x000,10xx,x100,0011)

Bit	Reset	Description
31:16	0x9c	**ACKQ_TMR**: **156** = DEFAULT
14:10	0x2	**PIPE_ISO_THRES**: **2** = INIT
6	0x1	**EXTRA_NUMP**: **1** = INIT

Bit	Reset	Description
5	0x0	**NUMP_MAX_ISOCH**: **0** = DISABLED **1** = ENABLED
4	0x0	**NUMP_MAX**: **0** = DISABLED **1** = ENABLED
3	0x0	**ISOBURST2**: **0** = DISABLED **1** = ENABLED
2	0x0	**ISOBURST1**: **0** = DISABLED **1** = ENABLED
1	0x1	**ACKQUEUE**: **0** = DISABLED **1** = ENABLED
0	0x1	**RETRY**: **0** = DISABLED **1** = ENABLED

T_XUSB_CFG_SSPX_CORE_ASYNC_0

Offset: 0x76c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**POST_ALL**: **0** = INIT

T_XUSB_CFG_SSPX_CORE_OUTXFER_0

Offset: 0x770
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	0x1	**CLAMP_NUMP_16**: **1** = INIT

T_XUSB_CFG_SSPX_CORE_BUBBLE_0

Offset: 0x774
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b00xx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**EN**: **0** = INIT
30	0x0	**TRAIN**: **0** = NOT_PEND **1** = SET
15:0	0x0	**CNT**: **0** = INIT

T_XUSB_CFG_SSPX_CORE_CTRL_LTSSM_0

Offset: 0x778
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00XX0000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	R/W	Reset	Description
20:16	RO	X	**COMPLIANCE_TYPE**
0	RW	0x0	**COMPLIANCE_TRIG**: **0** = INIT **1** = SET

T_XUSB_CFG_SSPX_CORE_UPHYCTL_HANDSHAKE_TIMEOUT_0

Offset: 0x77c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000ea79 (0b0000,0000,0000,0000,1110,1010,0111,1001)

Bit	Reset	Description
31:0	0xea79	**VAL**: **60025** = INIT

T_XUSB_CFG_SSPX_CORE_UPHYCTL_RX4_0

Offset: 0x780
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000003c (0bxxxx,xxxx,xxxx,0000,0000,0000,0011,1100)

Bit	Reset	Description
19:0	0x3c	**RXTERM_DATAEN**: **60** = INIT

T_XUSB_CFG_SSPX_CORE_UPHYCTL_RX5_0

Offset: 0x784
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00057e41 (0bxxxx,xxxx,xxxx,0101,0111,1110,0100,0001)

Bit	Reset	Description
19:0	0x57e41	**RXDAT_VLD_TIMEOUT**: **360001** = INIT

T_XUSB_CFG_SSPX_CORE_CNT74_0

Offset: 0x78c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000404 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0100,xxx0,0100)

Bit	Reset	Description
12	0x0	**CRD_C2_HP_TIMEOUT_LINK2INACTIVE_EN**: **0** = INIT

Bit	Reset	Description
11:8	0x4	**CRD_C2_HP_TIMEOUT_MAX_CNT**: **4** = INIT
4	0x0	**CRD_C1_HP_TIMEOUT_LINK2INACTIVE_EN**: **0** = INIT
3:0	0x4	**CRD_C1_HP_TIMEOUT_MAX_CNT**: **4** = INIT

T_XUSB_CFG_SSPX_CORE_MIN_TX4_0

Offset: 0x790
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00870010 (0bxxxx,xxxx,1000,0111,0000,0000,0001,0000)

Bit	Reset	Description
23:20	0x8	**MIN_RX_HOTRESET**: **8** = INIT
19:16	0x7	**MIN_RX_LOOPBACK**: **7** = INIT
15:0	0x10	**TS2_CNT_IN_LOOPBACK**: **16** = INIT

T_XUSB_CFG_SSPX_CORE_UPHYCTL_RX6_0

Offset: 0x794
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000e10 (0bxxxx,xxxx,xxxx,0000,0000,1110,0001,0000)

Bit	Reset	Description
19:0	0xe10	**RXDAT_VLD_TIMEOUT_U3**: **3600** = INIT

T_XUSB_CFG_SSPX_CORE_UPHYCTL_RX8_0

Offset: 0x7c0
Read/Write: R/W

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x80249f60 (0b1x00,0000,0010,0100,1001,1111,0110,0000)

Bit	Reset	Description
31	0x1	**PERIODIC_EQ_U02RECOVERY**: **1** = INIT
29:0	0x249f60	**PERIODIC_EQ_INTERVAL**: **2400096** = INIT

T_XUSB_CFG_SSPX_CORE_UPHYCTL_RX9_0

Offset: 0x7c4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000049 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0100,1001)

Bit	Reset	Description
7:0	0x49	**POLLING_FILTER_LFPS_TIMER**: **73** = INIT

T_XUSB_CFG_SSPX_CORE_UPHYCTL_RX10_0

Offset: 0x7c8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000e10 (0bxxxx,xxxx,xxxx,0000,0000,1110,0001,0000)

Bit	Reset	Description
19:0	0xe10	**RXDAT_VLD_TIMEOUT_ONCE_TRAINED**: **3600** = INIT

T_XUSB_CFG_SSPX_CORE_UPHYCTL_TX4_0

Offset: 0x7cc
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000002 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0010)

Bit	Reset	Description
3:0	0x2	**ASYNC_TX_WATERMARK**: **2** = INIT

9.2.4.4 XHCI Controller Memory-Mapped I/O Registers

T_XUSB_XHCI_CAP_REG0_0

Offset: 0x0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**HCIVERSION**: **288** = INIT
15:8	X	**RSVD**: **0** = _00
7:0	X	**CAPLENGTH**: **32** = INIT

T_XUSB_XHCI_CAP_HCSPARAMS1_0

Offset: 0x4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**MAXPORTS**: **16** = INIT
22:19	X	**RSVD0**: **0** = _00
18:8	X	**MAXINTRS**: **5** = INIT

Bit	Reset	Description
7:0	X	**MAXSLOTS**: **255** = INIT

T_XUSB_XHCI_CAP_HCSPARAMS2_0

Offset: 0x8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:27	X	**MAXSPBLO**: **0** = INIT
26	X	**SPR**: **0** = FALSE **1** = TRUE
25:21	X	**MAXSPBHI**: **0** = INIT
20:8	X	**RSVD**: **0** = _00
7:4	X	**ERST_MAX**: **15** = INIT
3:0	X	**IST**: **8** = INIT

T_XUSB_XHCI_CAP_HCSPARAMS3_0

Offset: 0xc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**U2LAT**: **598** = INIT
15:8	X	**RSVD**: **0** = _00

Bit	Reset	Description
7:0	X	**U1LAT**: **10** = INIT

T_XUSB_XHCI_CAP_HCCPARAMS1_0

Offset: 0x10
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**XECP**: **384** = USBLEGSUP **388** = SUPPROT_USB3 **391** = SUPPROT_USB2
15:12	X	**MAXPSASIZE**: **15** = INIT
11	X	**CFC**: **0** = FALSE **1** = TRUE
10	X	**SEC**: **0** = FALSE **1** = TRUE
9	X	**SPC**: **0** = FALSE **1** = TRUE
8	X	**PAE**: **0** = FALSE **1** = TRUE
7	X	**NSS**: **0** = FALSE **1** = TRUE
6	X	**LTC**: **0** = FALSE **1** = TRUE
5	X	**LHRC**: **0** = FALSE **1** = TRUE
4	X	**PIND**: **0** = FALSE **1** = TRUE

Bit	Reset	Description
3	X	**PPC**: **0** = FALSE **1** = TRUE
2	X	**CSZ**: **0** = _32B **1** = _64B
1	X	**BNC**: **0** = FALSE **1** = TRUE
0	X	**AC64**: **0** = FALSE **1** = TRUE

T_XUSB_XHCI_CAP_DBOFF_0

Offset: 0x14
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:2	X	**OFFSET**: **49152** = INIT
1:0	X	**RSVD**: **0** = _00

T_XUSB_XHCI_CAP_RTSOFF_0

Offset: 0x18
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:5	X	**OFFSET**: **4096** = INIT
4:0	X	**RSVD**: **0** = _00

T_XUSB_XHCI_CAP_HCCPARAMS2_0

Offset: 0x1c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:10	X	**RSVD**: **0** = INIT
9	X	**VTC**: **0** = FALSE **1** = TRUE
8	X	**GSC**: **0** = FALSE **1** = TRUE
7	X	**ETC_TSC**: **0** = FALSE **1** = TRUE
6	X	**ETC**: **0** = FALSE **1** = TRUE
5	X	**CIC**: **0** = FALSE **1** = TRUE
4	X	**LEC**: **0** = FALSE **1** = TRUE
3	X	**CTC**: **0** = FALSE **1** = TRUE
2	X	**FSC**: **0** = FALSE **1** = TRUE
1	X	**CMC**: **0** = FALSE **1** = TRUE
0	X	**U3C**: **0** = FALSE **1** = TRUE

T_XUSB_XHCI_OP_USBCMD_0

Offset: 0x20
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXX00X0 (0bxxxx,xxxx,xxxx,xxx0,000x,0000,0xxx,0000)

Bit	R/W	Reset	Description
31:17	RO	X	**RSVD1**: **0** = _00
16	RW	0x0	**VTIOE**: **0** = DISABLE **1** = ENABLE
15	RW	0x0	**TSC**: **0** = EN_DISABLE **1** = EN_ENABLE
14	RW	0x0	**ETE**: **0** = DISABLE **1** = ENABLE
13	RW	0x0	**CME**: **0** = DISABLE **1** = ENABLE
11	RW	0x0	**EU3S**: **0** = DISABLE **1** = ENABLE
10	RW	0x0	**EWE**: **0** = DISABLE **1** = ENABLE
9	RW	0x0	**CRS**: **0** = NOOP **1** = START
8	RW	0x0	**CSS**: **0** = NOOP **1** = START
7	RW	0x0	**LHCRST**: **0** = NOT_PENDING **1** = SET
6:4	RO	X	**RSVD0**: **0** = _00
3	RW	0x0	**HSEE**: **0** = DISABLE **1** = ENABLE
2	RW	0x0	**INTE**: **0** = DISABLE **1** = ENABLE

Bit	R/W	Reset	Description
1	RW	0x0	**HCRST**: **0** = NOT_PENDING **1** = SET
0	RW	0x0	**RS**: **0** = STOP **1** = RUN

T_XUSB_XHCI_OP_USBSTS_0

Offset: 0x24
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXX8XX (0bxxxx,xxxx,xxxx,xxxx,xxx0,1000,xxx0,00x1)

Bit	R/W	Reset	Description
31:13	RO	X	**RSVD2**: **0** = _00
12	RO	0x0	**HCE**: **0** = NO_ERROR **1** = ERROR
11	RO	0x1	**CNR**: **0** = READY **1** = NOT_READY
10	RW	0x0	**SRE**: **0** = NOT_PENDING **1** = CLEAR **1** = PENDING
9	RO	0x0	**RSS**: **0** = NOT_PENDING **1** = PENDING
8	RO	0x0	**SSS**: **0** = NOT_PENDING **1** = PENDING
7:5	RO	X	**RSVD1**: **0** = _00
4	RW	0x0	**PCD**: **0** = NOT_PENDING **1** = CLEAR **1** = PENDING
3	RW	0x0	**EINT**: **0** = NOT_PENDING **1** = CLEAR **1** = PENDING

Bit	R/W	Reset	Description
2	RW	0x0	**HSE**: **0** = NOT_PENDING **1** = CLEAR **1** = PENDING
1	RO	X	**RSVD0**: **0** = _0
0	RO	0x1	**HCH**: **0** = RUNNING **1** = HALTED

T_XUSB_XHCI_OP_PGSZ_0

Offset: 0x28
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXX0010 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0001,0000)

Bit	Reset	Description
31:16	X	**RSVD0**: **0** = _00
15:0	0x10	**PAGESIZE**: **16** = _64K

T_XUSB_XHCI_OP_DNCTRL_0

Offset: 0x34
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15	0x0	**N15**: **0** = DISABLE **1** = ENABLE
14	0x0	**N14**: **0** = DISABLE **1** = ENABLE
13	0x0	**N13**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
12	0x0	**N12**: **0** = DISABLE **1** = ENABLE
11	0x0	**N11**: **0** = DISABLE **1** = ENABLE
10	0x0	**N10**: **0** = DISABLE **1** = ENABLE
9	0x0	**N9**: **0** = DISABLE **1** = ENABLE
8	0x0	**N8**: **0** = DISABLE **1** = ENABLE
7	0x0	**N7**: **0** = DISABLE **1** = ENABLE
6	0x0	**N6**: **0** = DISABLE **1** = ENABLE
5	0x0	**N5**: **0** = DISABLE **1** = ENABLE
4	0x0	**N4**: **0** = DISABLE **1** = ENABLE
3	0x0	**N3**: **0** = DISABLE **1** = ENABLE
2	0x0	**N2**: **0** = DISABLE **1** = ENABLE
1	0x0	**N1**: **0** = DISABLE **1** = ENABLE
0	0x0	**N0**: **0** = DISABLE **1** = ENABLE

T_XUSB_XHCI_OP_CRCR0_0

Offset: 0x38
Read/Write: See table below
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x000000X0 (0b0000,0000,0000,0000,0000,0000,00xx,0000)

Bit	R/W	Reset	Description
31:6	WO	0x0	**CRPLO**: **0** = INIT
5:4	RO	X	**RSVD0**: **0** = _00
3	RO	0x0	**CRR**: **0** = STOPPED **1** = RUNNING
2	WO	0x0	**CA**: **0** = INIT **1** = ABORT
1	WO	0x0	**CS**: **0** = INIT **1** = STOP
0	WO	0x0	**RCS**: **0** = _0 **1** = _1

T_XUSB_XHCI_OP_CRCR1_0

Offset: 0x3c
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CRPHI**: **0** = INIT

T_XUSB_XHCI_OP_DCBAAP0_0

Offset: 0x50
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000000XX (0b0000,0000,0000,0000,0000,0000,00xx,xxxx)

Bit	R/W	Reset	Description
31:6	RW	0x0	**DCBAAPLO**: **0** = INIT
5:0	RO	X	**RSVD0**: **0** = _00

T_XUSB_XHCI_OP_DCBAAP1_0

Offset: 0x54
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DCBAAPHI**: **0** = INIT

T_XUSB_XHCI_OP_CONFIG_0

Offset: 0x58
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXX00 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	R/W	Reset	Description
31:10	RO	X	**RSVD0**: **0** = _00
9	RW	0x0	**CIE**: **0** = INIT
8	RW	0x0	**U3E**: **0** = INIT
7:0	RW	0x0	**MAXSLOTSEN**: **0** = INIT

T_XUSB_XHCI_OP_PORTSC_0

This is an array of 16 identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_XHCI_OP_PORTSC_<i>, among which <i> belongs to <0..15>.

Offset: 0x420,..,0x510
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xX00002aX (0b00xx,0000,0000,0000,0000,0010,1010,0x00)

Bit	R/W	Reset	Description
31	RO	0x0	**WPR**: **0** = NOT_PENDING **1** = SET
30	RO	0x0	**DR**: **0** = FALSE **1** = TRUE
29:28	RO	X	**RSVD2**: **0** = _00
27	RW	0x0	**WOE**: **0** = DISABLED **1** = ENABLED
26	RW	0x0	**WDE**: **0** = DISABLED **1** = ENABLED
25	RW	0x0	**WCE**: **0** = DISABLED **1** = ENABLED
24	RO	0x0	**CAS**: **0** = INIT
23	RW	0x0	**CEC**: **0** = NOT_PENDING **1** = CLEAR **1** = PENDING
22	RW	0x0	**PLC**: **0** = NOT_PENDING **1** = CLEAR **1** = PENDING
21	RW	0x0	**PRC**: **0** = NOT_PENDING **1** = CLEAR **1** = PENDING
20	RW	0x0	**OCC**: **0** = NOT_PENDING **1** = CLEAR **1** = PENDING
19	RW	0x0	**WRC**: **0** = NOT_PENDING **1** = CLEAR **1** = PENDING

Bit	R/W	Reset	Description
18	RW	0x0	**PEC**: **0** = NOT_PENDING **1** = CLEAR **1** = PENDING
17	RW	0x0	**CSC**: **0** = NOT_PENDING **1** = CLEAR **1** = PENDING
16	RW	0x0	**LWS**: **0** = DISABLED **1** = ENABLED
15:14	RW	0x0	**PIC**: **0** = OFF **1** = AMBER **2** = GREEN **3** = UNDEFINED
13:10	RO	0x0	**PSPD**: **0** = UNDEFINED **1** = FS **2** = LS **3** = HS **4** = SS **5** = SSP **15** = UNKNOWN
9	RW	0x1	**PP**: **0** = OFF **1** = ON
8:5	RW	0x5	**PLS**: **0** = U0 **1** = U1 **2** = U2 **3** = U3 **4** = DISABLED **5** = RXDETECT **6** = INACTIVE **7** = POLLING **8** = RECOVERY **9** = HOTRESET **10** = COMPLIANCE **11** = LOOPBACK **15** = RESUME
4	RW	0x0	**PR**: **0** = NOT_PENDING **1** = SET
3	RO	0x0	**OCA**: **0** = FALSE **1** = TRUE
2	RO	X	**RSVD0**: **0** = _0

Bit	R/W	Reset	Description
1	RW	0x0	**PED**: **0** = DISABLED **1** = CLEAR **1** = ENABLED
0	RO	0x0	**CCS**: **0** = NODEV **1** = DEV

T_XUSB_XHCI_OP_PORTPMSCSS_0

This is an array of 16 identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_XHCI_OP_PORTPMSCSS_<i>, among which <i> belongs to <0..15>.
Offset: 0x424,..,0x514
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXX0000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:17	RO	X	**RSVD0**: **0** = _00
16	RW	0x0	**FLA**: **0** = INIT
15:8	RW	0x0	**U2TIMEOUT**: **0** = INIT
7:0	RW	0x0	**U1TIMEOUT**: **0** = INIT

T_XUSB_XHCI_OP_PORTPMSCHS_0

This is an array of 16 identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_XHCI_OP_PORTPMSCHS_<i>, among which <i> belongs to <0..15>.
Offset: 0x424,..,0x514
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0XXX0000 (0b0000,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:28	RW	0x0	**TM**: **0** = DISABLE **1** = JSTATE **2** = KSTATE **3** = SE0NAK **4** = PACKET **5** = FORCEEN **15** = ERROR
27:17	RO	X	**RSVD0**: **0** = _00
16	RW	0x0	**HLE**: **0** = INIT
15:8	RW	0x0	**L1DS**: **0** = INIT
7:4	RW	0x0	**BESL**: **0** = INIT
3	RW	0x0	**RWE**: **0** = DISABLED **1** = ENABLED
2:0	RO	0x0	**L1S**: **0** = INVLD **1** = SUCCESS **2** = NYET **3** = STALL **4** = ERROR

T_XUSB_XHCI_OP_PORTLISC_0

This is an array of four identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_XHCI_OP_PORTLISC_<i>, among which <i> belongs to <0..3>.
Offset: 0x428,..,0x458
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXX0000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:24	RO	X	**RSVD0**: **0** = _00
23:20	RO	X	**TLC**: **0** = INIT
19:16	RO	X	**RLC**: **0** = INIT

Bit	R/W	Reset	Description
15:0	RW	0x0	**LEC**: **0** = INIT

T_XUSB_XHCI_OP_PORTHLPMC_0

This is an array of 16 identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_XHCI_OP_PORTHLPMC_<i>, among which <i> belongs to <0..15>.
Offset: 0x42c,..,0x51c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xx00,0000,0000,0000)

Bit	Reset	Description
13:10	0x0	**BESLD**: **0** = INIT
9:2	0x0	**L1_TIMEOUT**: **0** = INIT
1:0	0x0	**HIRDM**: **0** = INIT

T_XUSB_XHCI_OP_PORTHLPMCSS_0

This is an array of 16 identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_XHCI_OP_PORTHLPMCSS_<i>, among which <i> belongs to <0..15>.
Offset: 0x42c,..,0x51c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RSVD**: **0** = INIT
15:0	0x0	**LSEC**: **0** = INIT

T_XUSB_XHCI_EC_USBLEGSUP_0

Offset: 0x600
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxx0,xxxx,xxx0,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31:25	RO	X	**RSVD1**: **0** = _00
24	RW	0x0	**OSSEM**: **0** = INIT
23:17	RO	X	**RSVD0**: **0** = _00
16	RW	0x0	**BIOSSEM**: **0** = INIT
15:8	RO	X	**NEXT**: **4** = SUPPROT_USB3
7:0	RO	X	**CAPID**: **1** = USBLEGSUP

T_XUSB_XHCI_EC_USBLEGCTLSTS_0

Offset: 0x604
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0b000x,xxxx,xxx0,xxx0,000x,xxxx,xxx0,xxx0)

Bit	R/W	Reset	Description
31	RW	0x0	**BAR**: **0** = NOT_PENDING **1** = CLEAR **1** = PENDING
30	RW	0x0	**PCIC**: **0** = NOT_PENDING **1** = CLEAR **1** = PENDING
29	RW	0x0	**OSOC**: **0** = NOT_PENDING **1** = CLEAR **1** = PENDING
28:21	RO	X	**RSVD3**: **0** = _00

Bit	R/W	Reset	Description
20	RO	0x0	**HSE**: **0** = NOT_PENDING **1** = PENDING
19:17	RO	X	**RSVD2**: **0** = _00
16	RO	0x0	**EVI**: **0** = NOT_PENDING **1** = PENDING
15	RW	0x0	**BAREN**: **0** = DISABLED **1** = ENABLED
14	RW	0x0	**PCIEN**: **0** = DISABLED **1** = ENABLED
13	RW	0x0	**OSOEN**: **0** = DISABLED **1** = ENABLED
12:5	RO	X	**RSVD1**: **0** = _00
4	RW	0x0	**HSEEN**: **0** = DISABLED **1** = ENABLED
3:1	RO	X	**RSVD0**: **0** = _00
0	RW	0x0	**SMIEN**: **0** = DISABLED **1** = ENABLED

T_XUSB_XHCI_EC_SUPPROT_USB3_0_0

Offset: 0x610
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**MAJORREV**: **3** = _3
23:16	X	**MINORREV**: **16** = _0
15:8	X	**NEXT**: **6** = SUPPROT_USB2

Bit	Reset	Description
7:0	X	**CAPID**: **2** = SUPPROT_USB3

T_XUSB_XHCI_EC_SUPPROT_USB3_1_0

Offset: 0x614
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**NAMESTR**: **541217621** = USB

T_XUSB_XHCI_EC_SUPPROT_USB3_2_0

Offset: 0x618
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:28	X	**PSIC**: **2** = VAL
27:16	X	**RSVD0**: **0** = _00
15:8	X	**PORTCNT**
7:0	X	**PORTOFS**: **1** = VAL

T_XUSB_XHCI_EC_SUPPROT_USB3_3_0

Offset: 0x61c
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**SLOTTYPE**: **0** = VAL

T_XUSB_XHCI_EC_SUPPROT_USB3_PSI_G1_0

Offset: 0x620
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**PSIM**: **5** = VAL
15:14	X	**LP**: **0** = VAL
13:9	X	**RSVD**: **0** = VAL
8	X	**PFD**: **1** = VAL
7:6	X	**PLT**: **0** = VAL
5:4	X	**PSIE**: **3** = VAL
3:0	X	**PSIV**: **4** = VAL

T_XUSB_XHCI_EC_SUPPROT_USB3_PSI_G2_0

Offset: 0x624
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**PSIM**: **10** = VAL
15:14	X	**LP**: **1** = VAL
13:9	X	**RSVD**: **0** = VAL
8	X	**PFD**: **1** = VAL
7:6	X	**PLT**: **0** = VAL
5:4	X	**PSIE**: **3** = VAL
3:0	X	**PSIV**: **5** = VAL

T_XUSB_XHCI_EC_SUPPROT_USB2_0_0

Offset: 0x628
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**MAJORREV**: **2** = _3
23:16	X	**MINORREV**: **0** = _0
15:8	X	**NEXT**: **4** = XHCIIOV
7:0	X	**CAPID**: **2** = SUPPROT_USB3

T_XUSB_XHCI_EC_SUPPROT_USB2_1_0

Offset: 0x62c
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**NAMESTR**: **541217621** = USB

T_XUSB_XHCI_EC_SUPPROT_USB2_2_0

Offset: 0x630
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:28	X	**RSVD2**: **0** = _00
27:25	X	**MHD**: **0** = _00
24:21	X	**RSVD1**: **0** = _00
20	X	**BLC**: **1** = TRUE
19	X	**HLC**: **1** = TRUE
18	X	**IHI**: **0** = TRUE
17	X	**HSO**: **0** = TRUE
16	X	**RSVD0**: **0** = _00
15:8	X	**PORTCNT**
7:0	X	**PORTOFS**

T_XUSB_XHCI_EC_SUPPROT_USB2_3_0

Offset: 0x634
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**SLOTTYPE**: **0** = VAL

T_XUSB_XHCI_EC_XHCIIOV_HDR_0

Offset: 0x638
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000XXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
15:8	X	**NEXT**: **0** = NONE
7:0	X	**CAPID**: **4** = XHCIIOV

T_XUSB_XHCI_EC_XHCIIOV_INTR_RANGE_0

This is an array of four identical register entries; the register fields below apply to each entry. Full register list is: T_XUSB_XHCI_EC_XHCIIOV_INTR_RANGE_<i>, among which <i> belongs to <0..3>.
Offset: 0x63c,..,0x648
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00200000 (0bxxxx,xxxx,xx10,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
21	RO	0x1	**VFH**: **1** = INIT
20	RW	0x0	**VFR**: **0** = INIT
19:10	RW	0x0	**INTRCNT**: **0** = INIT
9:0	RW	0x0	**INTROFS**: **0** = INIT

T_XUSB_XHCI_EC_XHCIIOV_SLOT_ASSIGN_0

This is an array of 256 identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_XHCI_EC_XHCIIOV_SLOT_ASSIGN_<i>, among which <i> belongs to <0..255>.
Offset: 0x64c,..,0xa48
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000,0000)

Bit	Reset	Description
6	0x0	**SE**: **0** = INIT
5:0	0x0	**VF_ID**: **0** = INIT

T_XUSB_XHCI_RT_MFINDEX_0

Offset: 0x20000
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:14	X	**RSVD0**: **0** = _00
13:0	X	**MFINDEX**

T_XUSB_XHCI_RT_IMAN_0

This is an array of five identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_XHCI_RT_IMAN_<i>, among which <i> belongs to <0..4>.
Offset: 0x20020,..,0x200a0
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	R/W	Reset	Description
31:2	RO	X	**RSVD0**: **0** = _00
1	RW	0x0	**IE**: **0** = DISABLED **1** = ENABLED
0	RW	0x0	**IP**: **0** = NOT_PENDING **1** = CLEAR **1** = PENDING

T_XUSB_XHCI_RT_IMOD_0

This is an array of five identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_XHCI_RT_IMOD_<i>, among which <i> belongs to <0..4>.
Offset: 0x20024,..,0x200a4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXX0fa0 (0bxxxx,xxxx,xxxx,xxxx,0000,1111,1010,0000)

Bit	Reset	Description
31:16	X	**IMODC**
15:0	0xfa0	**IMODI**: **4000** = INIT

T_XUSB_XHCI_RT_ERSTSZ_0

This is an array of five identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_XHCI_RT_ERSTSZ_<i>, among which <i> belongs to <0..4>.
Offset: 0x20028,..,0x200a8
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXX0000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:16	RO	X	**RSVD0**: **0** = _00
15:0	RW	0x0	**SZ**: **0** = INIT

T_XUSB_XHCI_RT_ERRSVD_0

This is an array of five identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_XHCI_RT_ERRSVD_<i>, among which <i> belongs to <0..4>.
Offset: 0x2002c,..,0x200ac
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RSVD0**: **0** = _00

T_XUSB_XHCI_RT_ERSTBA0_0

This is an array of five identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_XHCI_RT_ERSTBA0_<i>, among which <i> belongs to <0..4>.
Offset: 0x20030,..,0x200b0
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0b0000,0000,0000,0000,0000,0000,0000,xxxx)

Bit	R/W	Reset	Description
31:4	RW	0x0	**ERSTBLO**: **0** = INIT
3:0	RO	X	**RSVD0**: **0** = _00

T_XUSB_XHCI_RT_ERSTBA1_0

This is an array of five identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_XHCI_RT_ERSTBA1_<i>, among which <i> belongs to <0..4>.
Offset: 0x20034,..,0x200b4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ERSTBHI**: **0** = INIT

T_XUSB_XHCI_RT_ERDP0_0

This is an array of five identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_XHCI_RT_ERDP0_<i>, among which <i> belongs to <0..4>.
Offset: 0x20038,..,0x200b8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:4	0x0	**DQPTRLO**: **0** = INIT
3	0x0	**EHB**: **0** = NOT_PENDING **1** = CLEAR **1** = PENDING
2:0	0x0	**DESI**: **0** = INIT

T_XUSB_XHCI_RT_ERDP1_0

This is an array of five identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_XHCI_RT_ERDP1_<i>, among which <i> belongs to <0..4>.
Offset: 0x2003c,..,0x200bc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DQPTRHI**: **0** = INIT

T_XUSB_XHCI_DB_0

This is an array of 256 identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_XHCI_DB_<i>, among which <i> belongs to <0..255>.
Offset: 0x30000,..,0x303fc
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000XX00 (0b0000,0000,0000,0000,xxxx,xxxx,0000,0000)

Bit	R/W	Reset	Description
31:16	WO	0x0	**STREAMID**: **0** = INIT
15:8	RO	X	**RSVD0**: **0** = _00
7:0	WO	0x0	**TARGET**: **0** = INIT

9.2.4.5 XUSB DEV Registers

T_XUSB_DEV_CFG_0_0

Offset: 0x0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:16	X	**DEVICE_ID_UNIT**: **8857** = XUSB_DEV
15:0	X	**VENDOR_ID**: **4318** = NVIDIA

T_XUSB_DEV_CFG_1_0

Offset: 0x4
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXX0XXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,x0xx,xxxx,x000)

Bit	R/W	Reset	Description
31	RO	X	**DETECTED_PERR**: The DETECTED_PERR bit indicates that the device has detected a parity error, even if parity error handing is disabled. (Bit 6 - T_CONFIG_FPCI_PERR_DISABLED) **0** = NOT_ACTIVE
30	RO	X	**SIGNALED_SERR**: The SIGNALED_SERR bit indicates that the device has asserted SERR#. **0** = NOT_ACTIVE
29	RO	X	**RECEIVED_MASTER**: The RECEIVED_MASTER bit indicates that a master device's transaction (except for Special Cycle) was terminated with a master-abort. This means that no device on the PCI bus responded to the address of the mastered transaction. All master devices must implement this bit. When this bit is set, an interrupt is signaled in the PBUS_INTR_0 register. **0** = NO_ABORT
28	RO	X	**RECEIVED_TARGET**: The RECEIVED_TARGET bit indicates that a master device's transaction was terminated with a target-abort. All master devices must implement this bit. When this bit is set, an interrupt is signaled in the PBUS_INTR_0 register. **0** = NO_ABORT
27	RO	X	**SIGNALED_TARGET**: The SIGNALED_TARGET bit indicates that the device has terminated a transaction with target-abort. Devices that will never signal target-abort do not need to implement this bit. When this bit is set, an interrupt is signaled in the PBUS_INTR_0 register. **0** = NO_ABORT
26:25	RO	X	**DEVSEL_TIMING**: The DEVSEL_TIMING bits contain the timing of DEVSEL#. There are three allowable timings for assertion of DEVSEL#. These are encoded as 00b for fast, 01b for medium, and 10b for slow (11b is reserved). These bits are read only and must indicate the slowest time that a device asserts DEVSEL# for any bus command except Configuration Read and Configuration Write. FPCI positive decode device are required to respond with fast DEVSEL# (0-cycle). Only the subtractive FPCI function responds with medium DEVSEL# to accept the cycle for the subtractive bus. **0** = FAST
24	RO	X	**MASTER_DATA_PERR**: **0** = NOT_ACTIVE
23	RO	X	**FAST_BACK2BACK**: The FAST_BACK2BACK bit indicates that the device is capable of handling back-to-back transfers when the transactions are not to the same agent. This bit can be set to 1 if the device can accept these transactions, and it must be set to 0 otherwise. **1** = CAPABLE
21	RO	X	**_66MHZ**: The 66 MHz bit indicates that the device is capable of 66 MHz PCI Bus operation. This value is initialized by a strapping bit. **1** = CAPABLE

Bit	R/W	Reset	Description
20	RO	X	**CAPLIST**: The CAPLIST bit indicates that the device configuration space includes a capabilities list starting at the offset indicated by T_XUSB_CFG_13. **1** = PRESENT
19	RO	X	**INTR_STATUS**: The INTR_STATUS bit is read-only and reflects the state of the interrupt in the device/function. Only when the INTR_DISABLE bit in the command register is a 0 and this INTR_STATUS bit is a 1, will the device's/function's INTx# signal be asserted. Setting the INTR_DISABLE bit to 1 has no effect on the state of this bit.
10	RW	0x0	**INTR_DISABLE**: The INTR_DISABLE bit indicates that it could disable the device/function from asserting INTx#. A value of 0 enables the assertion of INTx#, and a value of 1 disables the assertion of INTx# signal. The Device Status register is used to record status information for PCI bus related events. **0** = ON **1** = OFF
9	RO	X	**BACK2BACK**: **0** = DISABLED
8	RO	X	**SERR**: **0** = DISABLED
7	RO	X	**STEP**: **0** = DISABLED
6	RO	X	**PERR**: **0** = DISABLED
5	RO	X	**PALETTE_SNOOP**: The PALETTE_SNOOP bit indicates that VGA compatible devices should snoop their palette registers. When this bit is set, special palette snooping behavior is enabled (i.e., device must not respond). When the bit is reset, the device should treat palette accesses like all other accesses. VGA compatible devices should implement this bit. PALETTE_SNOOP is writable. **0** = DISABLED
4	RO	X	**WRITE_AND_INVAL**: The WRITE_AND_INVAL bit indicates that the device can use the Memory Write and Invalidate command when the transfer is aligned, and 16 bytes and the contents of the Cache Line Size Register is 4 DWORDS. When this bit is 1, masters may generate the command. When it is 0, Memory Write must be used instead. State after RST# is 0. This bit must be implemented by master devices that can generate the Memory Write and Invalidate command. **0** = DISABLED
3	RO	X	**SPECIAL_CYCLE**: **0** = DISABLED

Bit	R/W	Reset	Description
2	RW	0x0	**BUS_MASTER**: The BUS_MASTER bit indicates that the device can act as a master on the PCI bus. A value of 0 disables the device from generating PCI accesses. A value of 1 allows the device to behave as a bus master. BUS_MASTER is writable. **0** = DISABLED **1** = ENABLED
1	RW	0x0	**MEMORY_SPACE**: The MEMORY_SPACE bit indicates that the device will respond to memory space accesses. A value of 0 disables the device response. A value of 1 allows the device to respond to Memory space accesses. MEMORY_SPACE is writable. **0** = DISABLED **1** = ENABLED
0	RW	0x0	**IO_SPACE**: The IO_SPACE bit indicates that the device will respond to I/O space accesses. A value of 0 disables the device response. A value of 1 allows the device to respond to I/O space accesses. IO_SPACE is writable. **0** = DISABLED **1** = ENABLED

T_XUSB_DEV_CFG_2_0

Offset: 0x8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:24	X	**BASE_CLASS**: The CLASS_CODE bits identify the generic function of the device and (in some cases) a specific register-level programming interface. The register is broken into three byte-size fields. The upper byte (at offset 0BH) is a base class code which broadly classifies the type of function the device performs. The middle-byte (at offset 0BH) is a sub-class code which identifies more specifically the function of the device. The lower byte (at offset 09H) identifies a specific register-level programming interface (if any) so that device independent software can interact with the device. The Class Code and Revision ID are defined by parameters per block. **12** = SBC
23:16	X	**SUB_CLASS**: **3** = XUSB
15:8	X	**PROG_IF**: **254** = CUSTOM
7:0	X	**REVISION_ID**: The REVISION_ID bits specify a device specific revision identifier. The value is chosen by the vendor. Zero is an acceptable value. This field should be viewed as a vendor defined extension to the DEVICE_ID. **161** = VAL

T_XUSB_DEV_CFG_3_0

Offset: 0xc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00XXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
23	X	**HEADER_TYPE_FUNC**: **0** = SINGLE
22:16	X	**HEADER_TYPE_DEVICE**: The HEADER_TYPE bits identify the layout of the bytes 10h through 3FH in configuration space and also whether or not the device contains multiple functions. Bit 7 in this register is used to identify a multi-function device. If the bit is 0, then the device is single function. If the bit is 1, then the device has multiple functions. Bits 6 through 0 specify the layout of bytes 10h through 3Fh. The LATENCY_TIMER and HEADER_TYPE are defined by parameters per block. **0** = NON_BRIDGE
15:11	X	**LATENCY_TIMER**: The LATENCY_TIMER bits contain, in units of PCI bus clocks, the value of the Latency Timer for this PCI bus master. This register must be implemented as writable by any master that can burst more than two data phases. This register may be implemented as read-only for devices that burst two or fewer data phases, but the hardwired value must be limited to 16 or less. A typical implementation would be to build the five high-order bits (leaving the bottom three as read-only), resulting in a timer granularity of eight clocks. At reset, the register should be set to 0 (if programmable). LATENCY_TIMER bits are writable. **0** = _0_CLOCKS
7:0	X	**CACHE_LINE_SIZE**: **0** = _0

T_XUSB_DEV_CFG_4_0

Offset: 0x10
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000XXXX (0b0000,0000,0000,0000,0xxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31:15	RW	0x0	**BASE_ADDRESS**: The BASE_ADDRESS bits contain the base address of the device. The number of upper bits that a device actually implements depends on how much of the address space the device will respond to. A device that wants a 1 MB memory address space (using a 32-bit base address register) would build the top 12 bits of the address register, hardwiring the other bits to 0. Power-up software can determine how much address space the device required by writing a value of all 1's to the register and then reading the value back. The device will return 0's in all don't-care address bits, effectively specifying the address space required. **0** = DEFAULT
14:4	RO	X	**BAR_SIZE_32KB**: **0** = RSVD
3	RO	X	**PREFETCHABLE**: **1** = MERGABLE
2:1	RO	X	**ADDRESS_TYPE**: The ADDRESS_TYPE bits contain the type of the Base Address. It can be 32 bits, 20 bits, or 64 bits wide. **2** = _64_BIT
0	RO	X	**SPACE_TYPE**: The SPACE_TYPE bit indicates whether the register maps into Memory or I/O space. **0** = MEMORY

T_XUSB_DEV_CFG_5_0

Offset: 0x14
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**BASE_ADDRESSHI**: **0** = DEFAULT

T_XUSB_DEV_CFG_6_0

This is an array of five identical register entries; the register fields below apply to each entry.
Full register list is: T_XUSB_DEV_CFG_6_[i], among which [i] belongs to [0..4].
Offset: 0x18,..,0x28
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RSVD**: **0** = _00

T_XUSB_DEV_CFG_11_0

The SUBSYSTEM_VENDOR_ID bits and SUBSYSTEM_ID bits are used to uniquely identify the add-in board or subsystem where the device resides. When the device is on the motherboard, there is no serial ROM and the registers both initialize to NONE. The motherboard BIOS must set the values of the Subsystem ID and Subsystem Vendor ID by writing the proper values to the SUBSYSTEM_VENDOR_ID and SUBSYSTEM_ID bits in the PCI_T_16 register (NOT PCI_T_11).

Offset: 0x2c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**SUBSYSTEM_ID**: **0** = NONE
15:0	0x0	**SUBSYSTEM_VENDOR_ID**: **0** = NONE

T_XUSB_DEV_CFG_12_0

The Expansion ROM Base Address configuration register should not be used for any FPCI integrated blocks.

Offset: 0x30
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVED**: **0** = _0

T_XUSB_DEV_CFG_13_0

Offset: 0x34
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000000XX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
7:0	X	**CAP_PTR**: The CAP_PTR bits indicate the offset into configuration space where the capabilities list begins. This always points to 0x44 where at least the PCI-PM registers are expected to reside. **68** = DEFAULT

T_XUSB_DEV_CFG_14_0

Offset: 0x38
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:0	X	**RESERVED**: **0** = _0

T_XUSB_DEV_CFG_15_0

Offset: 0x3c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXX00 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	R/W	Reset	Description
31:24	RO	X	**MAX_LAT**: The MAX_LAT bits contain the maximum time the device requires to gain access to the CPI bus. This read-only register is used to specify the device's desired settings for Latency Timer values. The value specifies a period of time in units of 1/4 microsecond. Values of 0 indicate that the device has no major requirements for the settings of Latency Timers. MAX_LAT is nonzero. **0** = NO_REQUIREMENTS
23:16	RO	X	**MIN_GNT**: The MIN_GNT bits contain the length of the burst period a device needs assuming a clock rate of 33 MHz. This read-only register is used to specify the device's desired settings for Latency Timer values. The value specifies a period of time in units of 1/4 microsecond. Values of 0 indicate that the device has no major requirements for the settings of Latency Timers. MIN_GNT is nonzero. **0** = NO_REQUIREMENTS
15:8	RO	X	**INTR_PIN**: The INTR_PIN bits contain the interrupt pin the device (or device function) uses. A value of 1 corresponds to INTA#. A value of 2 corresponds to INTB#. A value of 3 corresponds to INTC#. A value of 4 corresponds to INTD#. Devices (or device functions) that don't use an interrupt pin must put a 0 in this register. This register is read-only. **1** = INTA
7:0	RW	0x0	**INTR_LINE**: The INTR_LINE bits contain the interrupt routing information. The register is read/write and must be implemented by any device (or device function) that uses an interrupt pin. POST software will write the routing information into this register as it initializes and configures the system. The value in this register tells which input of the system interrupt controller(s) the device's interrupt pin is connected to. Device drivers and operating systems can use this information to determine priority and vector information. INTR_LINE is initialized to 0xff (no connection) at reset. Some PCI BIOS can't handle aliased INTR_LINEs. Some PCI BIOS' can't handle INTR_LINE initialized to 0xff. **0** = IRQ0 **1** = IRQ1 **15** = IRQ15 **255** = UNKNOWN

T_XUSB_DEV_CFG_16_0

Offset: 0x40
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**SUBSYSTEM_ID**: **0** = NONE

Bit	Reset	Description
15:0	0x0	**SUBSYSTEM_VENDOR_ID**: **0** = NONE

T_XUSB_DEV_CFG_17_0

Offset: 0x44
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXXXXXXXX (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	X	**D3CPME_SUPPORT**: **1** = YES
30	X	**D3HPME_SUPPORT**: **1** = YES
29	X	**D2PME_SUPPORT**: **0** = NO
28	X	**D1PME_SUPPORT**: **0** = NO
27	X	**D0PME_SUPPORT**: **1** = YES
26	X	**D2_SUPPORT**: **0** = NO
25	X	**D1_SUPPORT**: **0** = NO
24:22	X	**AUXCUR**: **0** = SELF
21	X	**DSI**: **0** = NONE
20	X	**RSVD**: **0** = _0
19	X	**PMECLK**: **0** = NOT_REQUIRED
18:16	X	**VER**: **3** = _1P2
15:8	X	**NEXT_PTR**: **192** = DEFAULT
7:0	X	**CAP**: **1** = PCIPM

T_XUSB_DEV_CFG_18_PMCSR_0

Offset: 0x48
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00XXXXXX (0bxxxx,xxxx,xxxx,xxxx,0xxx,xxx0,xxxx,xx00)

Bit	R/W	Reset	Description
23:16	RO	X	**BSE_RSVD**: **0** = _00
15	RW	0x0	**PMESTATUS**: **0** = NOT_PENDING **1** = PENDING
14:13	RO	X	**DSCALE**: **0** = INIT
12:9	RO	X	**DSEL**: **0** = INIT
8	RW	0x0	**PME**: **0** = DISABLE **1** = ENABLE
7:4	RO	X	**RSVD1**: **0** = _00
3	RO	X	**NSR**: **1** = NORESET
2	RO	X	**RSVD0**: **0** = _0
1:0	RW	0x0	**PWRSTATE**: **0** = D0 **1** = D1 **2** = D2 **3** = D3H

T_XUSB_DEV_MSI_CTRL_0

Offset: 0xc0
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0080XXXX (0bxxxx,xxx0,1000,0000,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
24	RO	0x0	**VECTOR_MASK_CAP**: The VECTOR_MASK_CAP field indicates whether or not the controller supports MSI-per-vector masking. **0** = DEFAULT
23	RO	0x1	**_64_ADDR_CAP**: The 64_ADDR_CAP field indicates whether or not the controller is capable of generating a 64-bit message address. A value of 1 means the controller is capable of generating a 64-bit message address. **1** = DEFAULT
22:20	RW	0x0	**MULT_MSG_ENABLE**: System software writes to this field to indicate the number of allocated vectors (less than or equal to the number of vectors requested). The number of vectors is aligned as a power of two. When MSI is enabled, the controller is allocated at least one vector. **0** = DEFAULT
19:17	RO	0x0	**MULT_MSG_CAP**: System software reads this field to determine the number of requested vectors. The number of requested vectors must be aligned to a power of two. Values of 6 and 7 in this field are reserved. **0** = DEFAULT
16	RW	0x0	**MSI_ENABLE**: The MSI_ENABLE field enables the MSI capability. If MSI_ENABLE is written to a 1, the controller is permitted to use MSI to request service and is prohibited from using the legacy interrupt. System configuration software sets this bit to enable MSI. A device driver is prohibited from writing this bit to mask the controller's service request. If this bit is written to a 0, the controller is prohibited from using MSI to request service. **0** = DEFAULT
15:8	RO	X	**NEXT_PTR**: The NEXT_PTR field identifies the next item in the capabilities list. It is a read-only field. **0** = DEFAULT
7:0	RO	X	**CAP_ID**: **5** = MSI

T_XUSB_DEV_MSI_ADDR1_0

Offset: 0xc4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**MSG_ADDR**: System-specified message address. When MSI is enabled, this field specifies the d-word-aligned address for the MSI memory write transaction. **0** = DEFAULT

T_XUSB_DEV_MSI_ADDR2_0

Offset: 0xc8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSG_ADDR**: System-specified message address. When MSI is enabled, this field specifies the upper 32-bits of the address for the MSI memory write transaction. The contents of this register only apply when T_XUSB_CFG_MSI_CTRL_64_ADDR_CAP bit is set. **0** = DEFAULT

T_XUSB_DEV_MSI_DATA_0

Offset: 0xcc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**MSG_DATA**: System-specified message. When MSI is enabled, the message data is driven onto the lower 16-bits of the MSI memory write. The MULT_MSG_ENABLE field in configuration register 80h specifies the number of low order message data bits that the XHCI is permitted to modify to generate its system software allocated vectors. **0** = DEFAULT

T_XUSB_DEV_MSI_MASK_0

Offset: 0xd0
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**BIT**: For each mask bit that is set, the controller is prohibited from generating the associated message. bit 0 corresponds to MSI vector 0 bit 1 corresponds to MSI vector 1 bit 2 corresponds to MSI vector 2 bit 3 corresponds to MSI vector 3 bit 4 corresponds to MSI vector 4 bit 5 corresponds to MSI vector 5 bit 6 corresponds to MSI vector 6 bit 7 corresponds to MSI vector 7 Note: a value of 0 means to allow MSI generation for that vector; value of 1 means to prevent MSI generation for that vector. **0** = DEFAULT

T_XUSB_DEV_MSI_PEND_0

Offset: 0xd4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**BIT**: For each Pending bit that is set XUSB has a pending associated message. bit 0 corresponds to MSI vector 0 bit 1 corresponds to MSI vector 1 bit 2 corresponds to MSI vector 2 bit 3 corresponds to MSI vector 3 bit 4 corresponds to MSI vector 4 bit 5 corresponds to MSI vector 5 bit 6 corresponds to MSI vector 6 bit 7 corresponds to MSI vector 7 **0** = DEFAULT

T_XUSB_DEV_MSI_QUEUE_0

The MSI_QUEUE register is a private register. It specifies to which virtual channel queue (ISO or NON-ISO) that the MSI message is sent to before being sent out on FPCI bus.

Note: When the MSI message is created, the FPCI wrapper does not know which physical queue to put the MSI message into. The MSI_QUEUE setting determines this on a vector-by-vector basis. If the corresponding bit is set to 1, FPCI wrapper will put that MSI message into the ISO command/data queue; otherwise by default the message is placed in the NONISO queue. After the MSI message is popped from the ISO/NONISO queue, the settings in FPCICFG_ISOCMD, FPCICFG_PASSPW, and FPCICFG_RSPPASSPW apply. Since the XUSB FPCI wrapper, physically, only contains the NONISO command/data queues (from MCP8x and beyond), the MSI_QUEUE bits must be all set to 0 for all MSI vectors.

Offset: 0xd8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**BITS**: The BITS field specifies to which VC queue a particular MSI message will be sent to. bit 0 corresponds to MSI vector 0 bit 1 corresponds to MSI vector 1 bit 2 corresponds to MSI vector 2 bit 3 corresponds to MSI vector 3 bit 4 corresponds to MSI vector 4 bit 5 corresponds to MSI vector 5 bit 6 corresponds to MSI vector 6 bit 7 corresponds to MSI vector 7 **0** = ALL_NONISO **1** = ALL_ISO

T_XUSB_DEV_MSI_MAP_0

The MSI_MAP register is used to tell the OS that MSIs are supported in K8 mode. The register is for software information purposes only. This is the per-device HT MSI Capability Block.

Offset: 0xdc
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xXX0XXXXX (0bxxxx,xxxx,xxxx,xxx0,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31:27	RO	X	**CAP_TYPE**: The CAP_TYPE field is read only at 0x15 to indicate that this is an MSI Mapping Capability block. **21** = DEFAULT

Bit	R/W	Reset	Description
17	RO	X	**FIXD**: The FIXD bit is a read-only bit indicating if the the next two d-words for programmable address are present in the capability. If set, the address for mapping MSIs is fixed at 0x0000_0000_FEEx_xxxx and that this capability block is 1 d-word long. **1** = ON
16	RW	0x0	**EN**: The EN bit indicates if the mapping is active. It is cleared upon warm reset. **0** = DEFAULT
15:8	RO	X	**NEXT_PTR**: The NEXT_PTR field points to the next item in the capabilities list. It is a read-only field. **0** = DEFAULT
7:0	RO	X	**CAP_ID**: The CAP_ID field identifies that this is an HT capability list item. (Read-only as 0x8). **8** = MSI

T_XUSB_DEV_AXI_CFG_0

Offset: 0xf8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,xxx0,1010)

Bit	Reset	Description
8	0x0	**CLR_MSI_INTR**: **0** = NOT_PENDING **1** = PENDING **1** = TRIGGER
4	0x0	**ORDER_RSP_VS_WRITES**: **0** = FALSE **1** = TRUE
3	0x1	**AW_FORCE_NONCOH**: **0** = FALSE **1** = TRUE
2	0x0	**AW_FORCE_COH**: **0** = FALSE **1** = TRUE
1	0x1	**AR_FORCE_NONCOH**: **0** = FALSE **1** = TRUE
0	0x0	**AR_FORCE_COH**: **0** = FALSE **1** = TRUE

9.3 PCI Express (PCIe) Controller

9.3.1 Overview

The NVIDIA® Orin™ implements a PCI Express® (PCIe) controller that supports Gen4 PCIe transfer speeds, as well as both PCIe Root Port and Endpoint modes.

The PCIe Root Ports enable Orin to connect to external PCIe devices to expand the functionality of the platform through the wide range of PCIe solutions in the market or through custom FPGAs. Examples of devices connecting to Orin via PCIe include:

- NVIDIA discrete GPU.
- SATA or NVMe based SSD.
- Wi-Fi adapter.
- Non-transparent Switch for connecting multiple SoCs.

The PCIe Endpoint mode allows two NVIDIA SoCs to connect to each other directly. It also allows NVIDIA SoCs to connect to other SoCs either directly, or through standard switches. The integrated DMA engine in the Endpoint helps to offload the CPU workload by performing DMA data transfer.

Standards Supported

- PCI Express Base Specification Revision 4.0, Version 1. 0.
- PCI Express Card Electromechanical Specification Revision 4.0, Version 1.0.

9.3.1.1 Features

The Orin supports the following controller features:

- PCIe4.0-compliant Root Port controller IP integration
  - Supports Gen1, Gen2, Gen3, and Gen4 link speeds
  - Supports 256-byte maximum payload size
- PCIe Dual-mode Controller
  - Supports PCIe Endpoint mode operation
  - Supports integrated DMA engine
- PCIe controller configurations
  - x8 lane configuration
    - Supports x8, x4, x2, and x1 links
    - Supports both Root Port and Endpoint modes
    - Supports lane reversal for x8, x4, or x2 and lane flipping for x4, x2, or x1

    - x4 lane configuration
        - Supports x4, x2, and x1 links
        - Supports both Root Port and Endpoint modes
        - Supports lane reversal for x4 or x2 and lane flipping for x2 or x1
    - x1 lane configuration
        - Supports x1 link
        - Root Port operations only

Gen1 or Gen 2 speeds for a x1 link on a x4 or x8 controller are typically slower than a x1 controller (faster).

- PCIe messages and message signaled Interrupt (MSI/MSI-X)
    - Reporting received messages in either Root Port or Endpoint mode
    - Generating messages in either Root Port or Endpoint mode
    - Reporting received MSI/MSI-X in Root Port mode and generating MSI/MSI-X in Endpoint mode
- PCIe link low-power states supported
    - L0s, L1, L1 sub-states, and L2/3 link low power states
        - Controller dynamic clock gating in L0 and L0s
        - UPHY PLL and PLLE power down in L1 and L1 sub-states
        - Controller power-gating in L2/3
    - Link speed management
- PCIe capabilities and services supported
    - Virtual Channel 0 (VC0)
    - Hot-plug via virtual GPIO
    - Advanced Error Reporting (AER)
    - Alternative Routing ID interpretation (ARI)
    - Latency Tolerance Reporting (LTR)
    - Precision Timing Management (PTM)
    - Power State D0/D3
- PCIe capabilities and services not supported
    - Virtual Channel 1 through 7 (VC1 ~ VC7)
    - Power Budgeting Capability
    - Multicast
    - Atomic Operations
    - Address Translation Services (ATS)
    - Dynamic Power Allocation (DPA)
    - Readiness Notification (RN)

    - Function Readiness Status (FRS)
    - Device Readiness Status (DRS)
    - Immediate Readiness (IR)
    - Readiness Time Reporting (RTR)
  - Vital Product Data (VPD)
  - TLP (Transaction Layer Packet) Processing Hints (TPH)
  - Optimized Buffer Flush/Fill (OBFF)
  - Lightweight Notification (LN)
  - Power State D1/D2
  - Peer-to-Peer Transactions
  - Locked Transactions

9.3.1.2 External Connectivity Diagrams

The following diagrams illustrates the PCIe controllers in the HSIO cluster.

Figure 9.12 PCIe Controllers Connecting to x8 HSIO UPHY

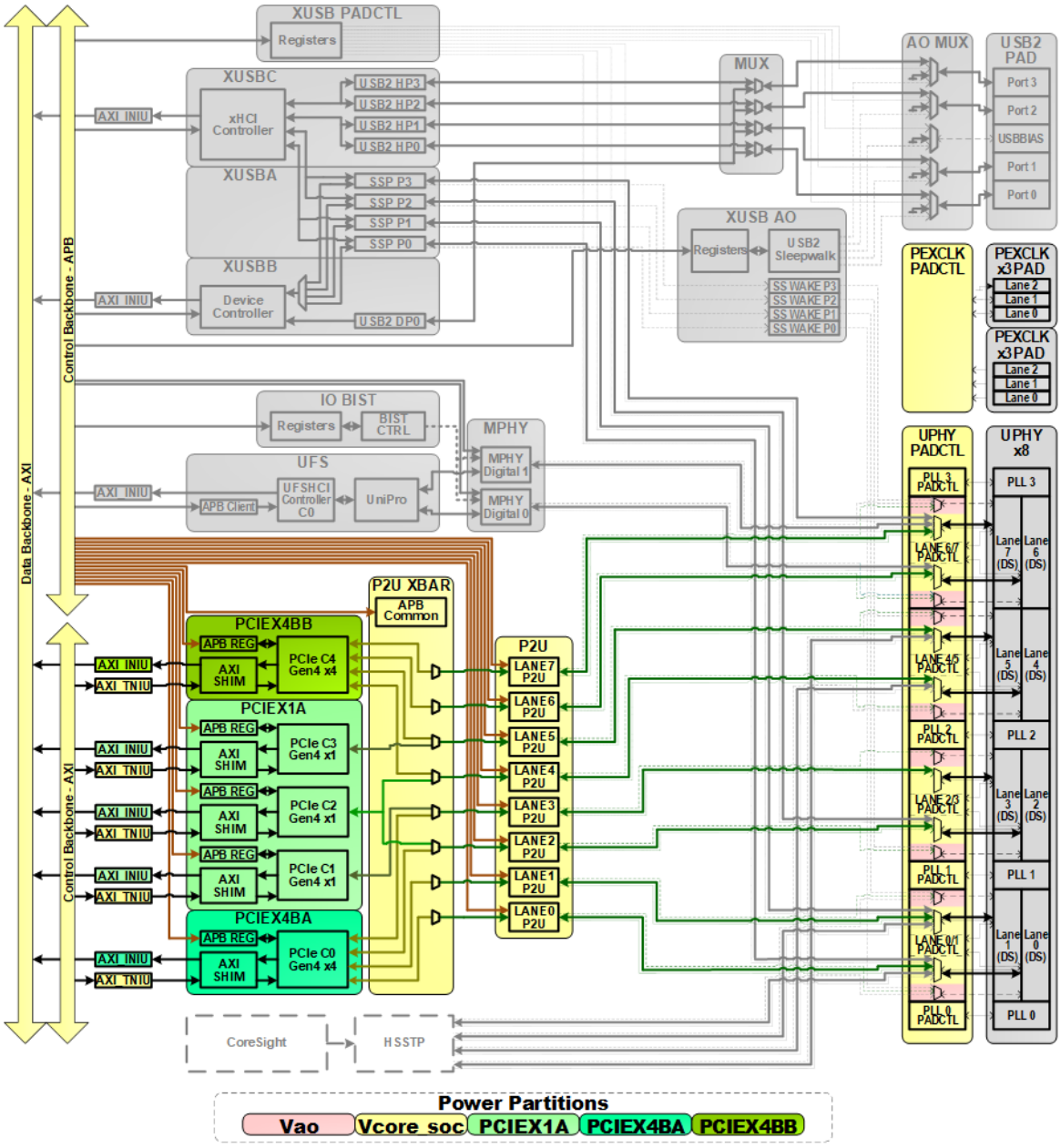


Figure 9.13 PCIe Controllers Connecting to x8 NHVS UPHY

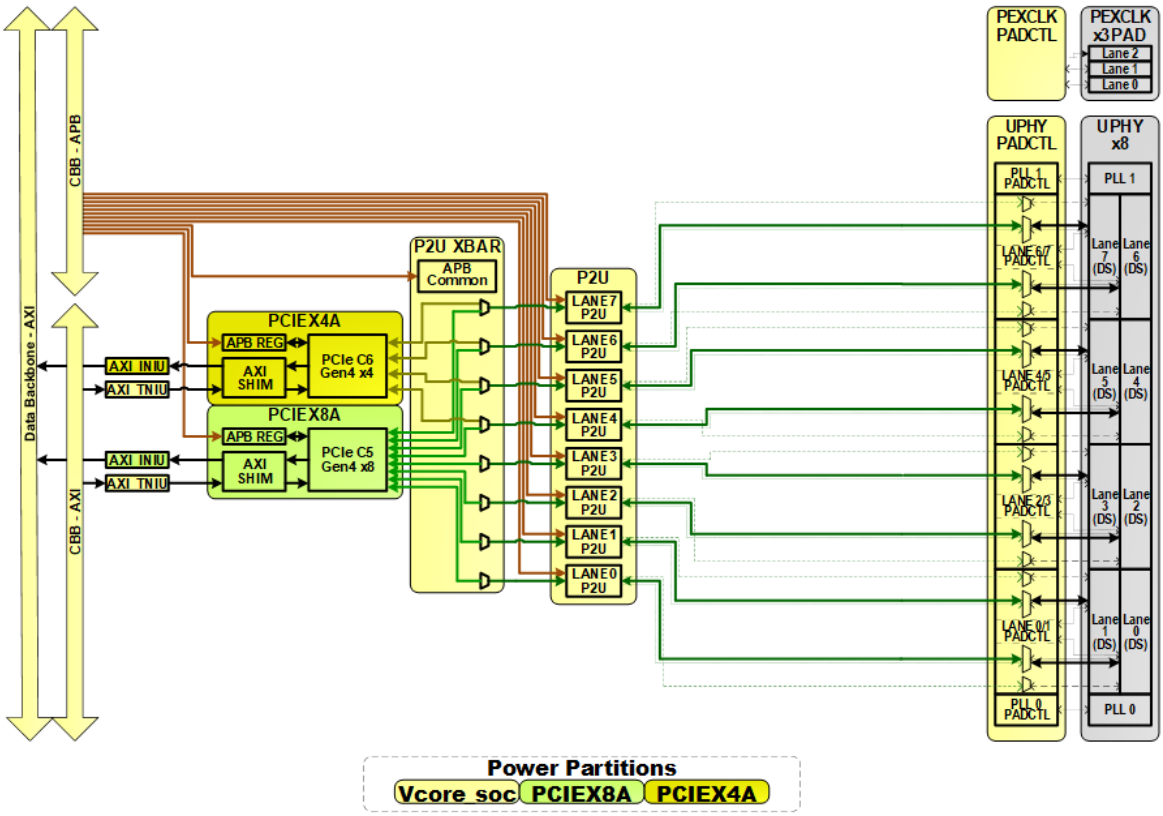
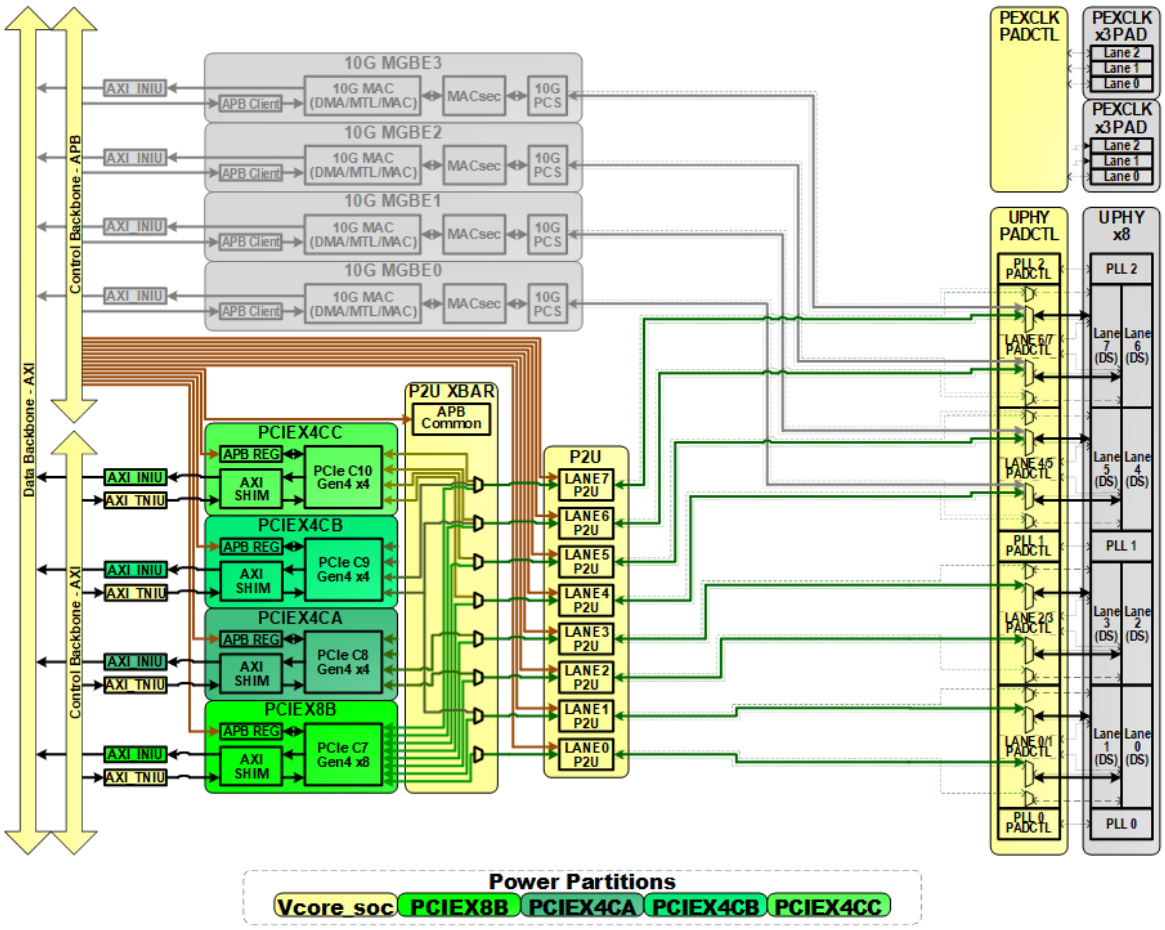


Figure 9.14 PCIe Controllers Connecting to x8 GBE UPHY



9.3.2 Interfaces

9.3.2.1 System Interconnect Interfaces

Each PCIe controller is encapsulated in its own AXI and APB wrappers for integrating into the Orin system interconnect. The AXI and APB wrappers implement the following system backbone interfaces:

- Data Backbone AXI Interface
  - Compliant with Socket 2.0
  - Traffic type:
    - DMA requests forwarded from external PCIe devices or integrated DMA engine
- Control Backbone AXI Interface
  - Compliant with Socket 2.0

    - Traffic type:
      - Control accesses targeting PCIe configuration registers internal to PCIe controller
      - Control accesses targeting PCIe memory, I/O, and configuration address spaces of external PCIe devices
- Control Backbone APB Interface
  - Compliant with Socket 2.0
  - Traffic type:
    - Control accesses targeting NVIDIA specific privates registers that configure PCIe controller operations
    - Control accesses targeting NVIDIA specific privates registers that control generation or receiving PCIe messages, MSI, and sideband signals

9.3.2.2 External I/O Interfaces

Each PCIe controller supports PIPE version 4.3 as its physical layer interface to PIPE2UPHY.

Each PCIe controller supports the following external sideband interfaces via pinmux.

- PEX_CLK*N*p/n
  - Root Port – differential reference clock output to external PCIe devices.
  - Endpoint – not used.
- PEX_REFCLK*N*p/n
  - Root Port – not used.
  - Endpoint – UPHY differential reference clock input.
- PEX_RST*N*_n
  - Root Port – active low reset output to external PCIe devices.
  - Endpoint – not used.
- PEX_CLKREQ*N*_n
  - Root Port – active low bidirectional signal indicating devices requesting output of reference clock, as well as for both Root Port and device to request and indicate wake from L1 sub-states.
  - Endpoint – active low bidirectional signal to request output of reference clock, as well as for both Root Port and device to request and indicate wake from L1 sub-states.

Each PCIe controller supports the following external sideband interfaces via virtual GPIO. Platform implementation can support presence detection via implementation specific mechanism and indicate the presence of PCIe device via APB Register Interface.

- PEX_PRSNTn
  - Root Port – active low input to indicate presence of PCIe device.
  - Endpoint – not used.

All PCIe controllers share one single signal for all PCIe devices to request wake from L2/3 state. The wake signal is not directly used by PCIe controllers; instead, this signal is directly controlled by PINMUX and PMC for handling PCIe wake events from PCIe devices.

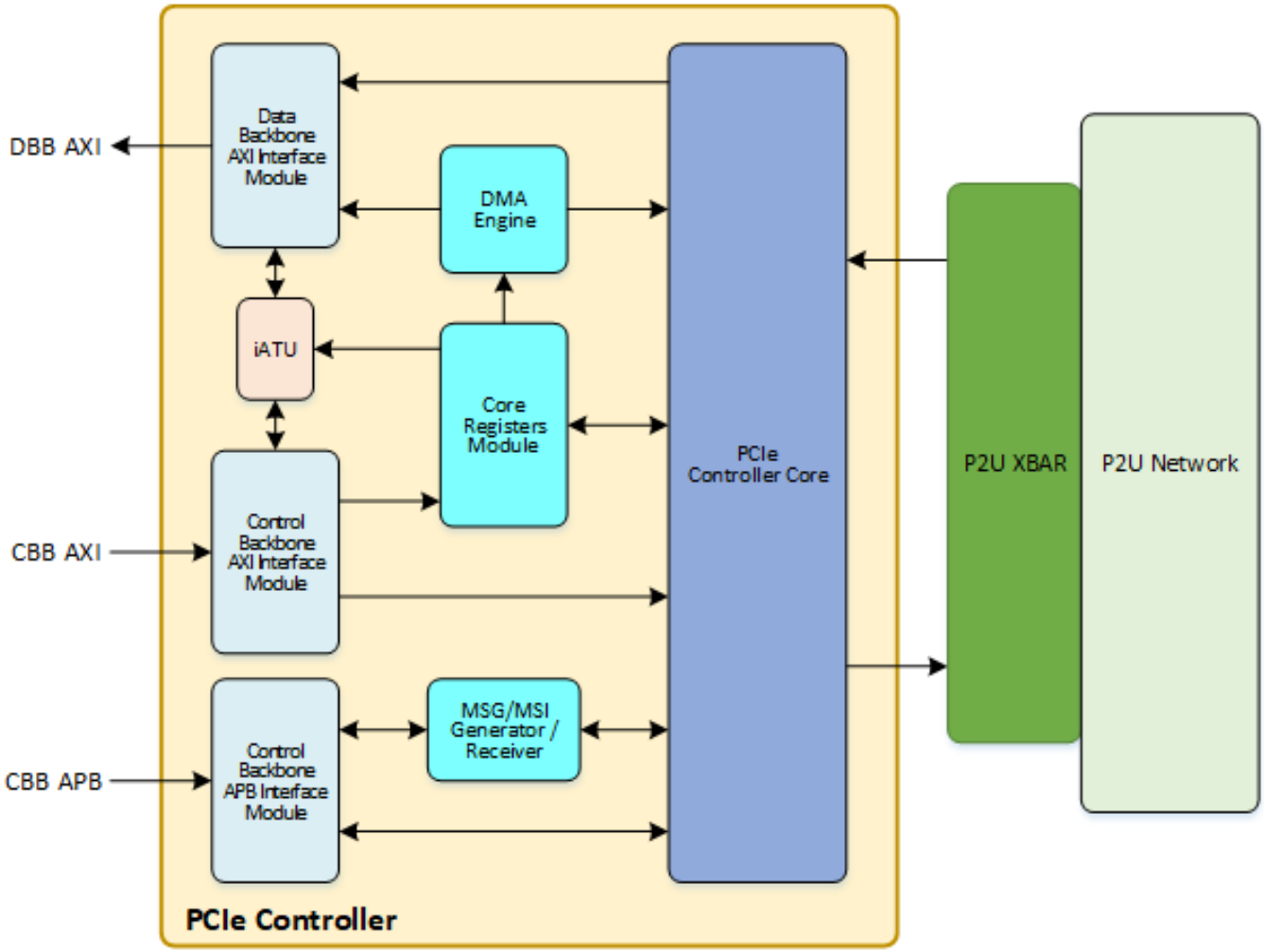
- PEX_WAKE_n
  - Root Port complex – active low input signal indicating devices requesting wake from L2/3 link power states.
  - Endpoint – not used.

Each PCIe Endpoint controller requires the following external sideband signal via virtual GPIOs. The Reset signal is not directly connected to PCIe controllers; when the Reset signal is asserted, software is required to use software override to reset the PCIe Endpoint controller. The wake signal is not directly used by PCIe controllers; instead, this signal is directly controlled by PINMUX and PMC for signaling PCIe wake events. Platform implementation allocates GPIOs as the Reset and wake signals for the PCIe Endpoint if the platform supports PCIe Endpoint.

- PEX_EP_RST_n
  - Endpoint – active low reset input from PCIe Root Port.
- PEX_EP_WAKE_n
  - Endpoint – active low output signal to request wake from L2/3 link power states if Endpoint mode needs to support wake from L2/3.

The PCIe controllers with x1 lane configuration support only Root Port operations, whereas the PCIe controllers with x4 and x8 lane configurations support both Root Port and Endpoint operations that need to be set up during controller initialization. The following diagram depicts the control and data paths of modules inside the PCIe controllers.

Figure 9.15 PCIe Controller Block Diagram



The functions of each module inside the PCIe Controller are given below.

- **PCIe Controller Core**

  The PCIe Controller Core handles PCIe protocols in the transport packet, data link, and physical layer to achieve the following:

  - Controlling PCIe link state transitions.
  - Transmitting and receiving PCIe packets.

- **Data BackboneAXI Interface Module**

  The Data Backbone AXI Interface Module converts requests forwarded by the PCIe Controller Core or generated by the DMA Engine to AXI protocols then issues these requests to the SoC Data Backbone.

- **Control Backbone AXI Interface Module**

  The Control Backbone AXI Interface Module processes requests received from the SoC Control Backbone then forwards them to the PCIe Controller Core to be sent to the external devices via PCIe Link or the Core Registers Module to access the registers inside.

- **Control Backbone APB Interface Module**

  The Control Backbone APB Interface Module contains the application registers for the following:

  - Setting up the PCIe Controller.
  - Accessing the status of the PCIe Controller.
  - Controlling the MSG/MSI Generator/Receiver.

- **MSG/MSI Generator/Receiver**

  The Messages (MSG) and Message Signaled Interrupts (MSI) Generator/Receiver performs the following:

  - Generating the PCI Express Messages (MSG) and Message Signaled Interrupts (MSI) as set up according to the values of the registers accessed through the APB Interface Module.
  - Receiving the PCI Express Messages (MSG) and Message Signaled Interrupts (MSI) and reflecting the status in the registers accessed through the APB Interface Module.

- **Core Registers Module**

  The Core Registers Module contains the PCIe configuration registers of the PCIe root and end point as well as the memory-mapped I/O registers for setting up the DMA Engine and iATU (internal Address Translation Unit).

- **DMA Engine**

  The DMA Engine can be set up to perform DMA operations for transferring data between the internal SoC memory and the memory buffers in the external devices connected to the SoC via PCIe Link. The DMA engine supports both Non-Linked-List Mode with DMA transfers directly through the memory-mapped I/O registers in the Core Registers Module and Linked-List Mode with DMA transfers through linked descriptor elements in the system memory buffer.

- **iATU**

  The iATU (internal Address Translation Unit) translates the requestaddresses between the AXI address space and the PCIe address spaces, including ECAM (Enhanced Configuration Access Mechanism) support that maps memory requests to configuration requests as defined in the PCIe specification.

External to the individual PCIe Controllers are two supporting modules that connect the PCIe Controllers to the UPHY as depicts in the external connectivity diagrams.

- **P2U (or PIPE2UPHY)**

  The P2U Module handles the following:

  - Symbol encoding/decoding for Gen1/Gen2
  - Data stream encoding/decoding for Gen3/Gen4
  - Receiver clock compensation for individual lanes, where the P2U is 1-to-1 mapped to a UPHY lane.

- **P2U XBAR (or PIPE2UPHY Crossbar)**

  The P2U XBAR cross connects the individual PCIe Controller to one or more P2U modules to support x1, x2, x4, or x8 connectivity as shown in the external connectivity diagrams.

9.3.3 Operation Models

9.3.3.1 Normal Root Port Operation

The PCIe host driver is responsible of the following tasks:

- Initialize PCIe Root Port controller, including coordinating with CAR and UPHY drivers to initialize and setup PLLE and UPHY operations.
- Program iATU to setup outbound address mappings to PCIe address spaces.
- Discover and enumerate the PCIe switch ports and Endpoints under each Root Port.
- Program integrated DMA engine to initialize and setup DMA operations if applicable (e.g. when the Root Port is used in SoC to SoC interconnect).

Th device-specific driver is responsible of the following tasks:

- Initialize and setup device specific operations following device specific programming model.

9.3.3.2 Normal Endpoint Operation

The PCIe Endpoint driver is responsible of the following tasks:

- Initialize PCIe Endpoint controller, including coordinating with CAR and UPHY drivers to initialize and setup PLLE and UPHY operations.
- Program iATU to setup inbound address mappings from PCIe memory address.
- Program integrated DMA engine to initialize and setup DMA operations.

9.3.4 Programming Guidelines

9.3.4.1 Initialization

9.3.4.1.1 PLL and PHY Initialization

The PAD driver enables PCIe related PLLs. For UPHY PLLs, PAD driver is required to setup the frequencies for these PLLs based on the lane muxing configuration and the targeted link speeds of the PCIe links. PAD driver enables platform specific regulators enable power rails to the pads. PAD driver de-asserts Reset to UPHY PAD and UPHY PAD Macro.

- Set the following CAR register field to '0' to de-assert Reset to UPHY Pad and UPHY Pad Macros.
  - (CLK_RST_CONTROLLER_RST_DEV_UPHY_0.SWR_UPHY_RST = 0)
  - (CLK_RST_CONTROLLER_RST_DEV_PEX_USB_UPHY_0.SWR_PEX_USB_UPHY_RST = 0)

The PAD driver programs the UPHY Pad Macro registers to assign the UPHY lanes, set static UPHY LANE and PLL parameters according to the platform specific configuration, and assign the static UPHY LANE parameters of lanes assigned to PCIe, according to the platform specific configuration. P2U driver de-asserts Reset to PIPE2UPHY Common APB.

- Set the following CAR register field to '0' to de-assert Reset to PIPE2UPHY Common APB.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_COMMON_APB_RST = 0)

The P2U driver programs the Common APB registers to map the P2U to PCIe controllers.

- APPL_COMMON_CONTROL.XBAR_CONFIG

9.3.4.1.2 Root Port

The PAD driver programs the PINMUX registers to setup the GPIO sideband signals for PCIe Root Port controllers.

- Set the following PINMUX register field to setup the WAKE# signal shared by all PCIe Root Ports.
  - (PADCTL_PEX_CTL_PEX_WAKE_N_0.E_INPUT = ENABLE)
- Set the following PINMUX register field to setup the RESET# and CLKREQ# signals for each PCIe controller configured as a Root Port.
  - (PADCTL_PEX_CTL_PEX_L0_CLKREQ_N_0.E_INPUT = ENABLE)
  - (PADCTL_PEX_CTL_PEX_L0_CLKREQ_N_0.TRISTATE = PASSTHROUGH)
  - (PADCTL_PEX_CTL_PEX_L0_RST_N_0.TRISTATE = PASSTHROUGH)
  - (PADCTL_PEX_CTL_PEX_L1_CLKREQ_N_0.E_INPUT = ENABLE)
  - (PADCTL_PEX_CTL_PEX_L1_CLKREQ_N_0.TRISTATE = PASSTHROUGH)

- (PADCTL_PEX_CTL_PEX_L1_RST_N_0.TRISTATE = PASSTHROUGH)
- (PADCTL_PEX_CTL_PEX_L2_CLKREQ_N_0.E_INPUT = ENABLE)
- (PADCTL_PEX_CTL_PEX_L2_CLKREQ_N_0.TRISTATE = PASSTHROUGH)
- (PADCTL_PEX_CTL_PEX_L2_RST_N_0.TRISTATE = PASSTHROUGH)
- (PADCTL_PEX_CTL_PEX_L3_CLKREQ_N_0.E_INPUT = ENABLE)
- (PADCTL_PEX_CTL_PEX_L3_CLKREQ_N_0.TRISTATE = PASSTHROUGH)
- (PADCTL_PEX_CTL_PEX_L3_RST_N_0.TRISTATE = PASSTHROUGH)
- PADCTL_PEX_CTL_PEX_L4_CLKREQ_N_0.E_INPUT = ENABLE)
- (PADCTL_PEX_CTL_PEX_L4_CLKREQ_N_0.TRISTATE = PASSTHROUGH)
- (PADCTL_PEX_CTL_PEX_L4_RST_N_0.TRISTATE = PASSTHROUGH)
- (PADCTL_PEX_CTL_PEX_L5_CLKREQ_N_0.E_INPUT = ENABLE)
- (PADCTL_PEX_CTL_PEX_L5_CLKREQ_N_0.TRISTATE = PASSTHROUGH)
- (PADCTL_PEX_CTL_PEX_L5_RST_N_0.TRISTATE = PASSTHROUGH)

9.3.4.1.3 Endpoint

The PAD driver programs the PINMUX registers to setup the GPIO sideband signals for PCIe Endpoint controllers.

- Set the following PINMUX register field to setup the CLKREQ# for each PCIe controller configured as an Endpoint.
  - (PADCTL_PEX_CTL_PEX_L0_CLKREQ_N_0.E_INPUT = ENABLE)
  - (PADCTL_PEX_CTL_PEX_L0_CLKREQ_N_0.TRISTATE = PASSTHROUGH)
  - (PADCTL_PEX_CTL_PEX_L0_RST_N_0.E_INPUT = DISABLE)
  - (PADCTL_PEX_CTL_PEX_L4_CLKREQ_N_0.E_INPUT = ENABLE)
  - (PADCTL_PEX_CTL_PEX_L4_CLKREQ_N_0.TRISTATE = PASSTHROUGH)
  - (PADCTL_PEX_CTL_PEX_L4_RST_N_0.E_INPUT = DISABLE)
  - (PADCTL_PEX_CTL_PEX_L5_CLKREQ_N_0.E_INPUT = ENABLE)
  - (PADCTL_PEX_CTL_PEX_L5_CLKREQ_N_0.TRISTATE = PASSTHROUGH)
  - (PADCTL_PEX_CTL_PEX_L5_RST_N_0.E_INPUT = DISABLE)

As the PCIe Endpoint controllers cannot use the per-controller RESET# output when configured in Endpoint mode, for Orin to connect to a PCIe Root Port, it allocates a GPIO pin under direct software control as the PCIe Endpoint's RESET#. The GPIO is required to be setup to generate Interrupt when the system is in SC0 and generate wake even when the system is in SC7.

The PCIe driver enables the allocated GPIO as an Interrupt source to signal Reset event from PCIe Root Port to the PCIe Endpoint controller.

- Enable the Interrupt event through GPIO registers

As the PCIe Endpoint controllers do not directly support WAKE#, if wake signaling is required when Orin is connected to a PCIe Root Port and the link is in L2, Orin allocates a GPIO pin under direct software control to drive WAKE#.

9.3.4.1.4 Programming CLKREQ

CLKREQ requires special programming to operate correctly.

CLKREQ# is an input-only open-drain signal for Endpoints supporting clock power management but not L1 sub-states, and is a bidirectional open-drain signal for Endpoints supporting L1 sub-states. Software needs to configure the pads after it checks the capabilities of the Endpoint after the link is up.

- Set the following field of the PCIe PADCTL registers to '1', if Endpoint supports clock power management but does not support L1 sub-states.
  - (PADCTL_PEX_CTL_PEX_L0_CLKREQ_N_0.TRISTATE = 1)
  - (PADCTL_PEX_CTL_PEX_L1_CLKREQ_N_0.TRISTATE = 1)
  - (PADCTL_PEX_CTL_PEX_L2_CLKREQ_N_0.TRISTATE = 1)
  - (PADCTL_PEX_CTL_PEX_L3_CLKREQ_N_0.TRISTATE = 1)
  - (PADCTL_PEX_CTL_PEX_L4_CLKREQ_N_0.TRISTATE = 1)
  - (PADCTL_PEX_CTL_2_PEX_L5_CLKREQ_N_0.TRISTATE = 1)
- Set the following field of the PCIe PADCTL registers to '0', if Endpoint supports L1 sub-states.
  - (PADCTL_PEX_CTL_PEX_L0_CLKREQ_N_0.TRISTATE = 0)
  - (PADCTL_PEX_CTL_PEX_L1_CLKREQ_N_0.TRISTATE = 0)
  - (PADCTL_PEX_CTL_PEX_L2_CLKREQ_N_0.TRISTATE = 0)
  - (PADCTL_PEX_CTL_PEX_L3_CLKREQ_N_0.TRISTATE = 0)
  - (PADCTL_PEX_CTL_PEX_L4_CLKREQ_N_0.TRISTATE = 0)
  - (PADCTL_PEX_CTL_2_PEX_L5_CLKREQ_N_0.TRISTATE = 0)

9.3.4.1.5 PCIe Controller Setup

The PCIe driver de-asserts Reset to NVHS rail domain if PCIe C5 is enabled in the platform.

- Set the following CAR register field to '0' to de-assert Reset to the NVHS power rail domain.
  - (CLK_RST_CONTROLLER_RST_DEV_NVHS_RAIL_0.SWR_NVHS_RAIL_RST = 0)

The PCIe driver de-asserts Reset to APB Client of each PCIe controller.

- Set the following CAR register field to '0' to de-assert Reset to the APB clients of PCIe controllers.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_0_APB_RST = 0)
  - (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_1_APB_RST = 0)

- (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_2_APB_RST = 0)
- (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_3_APB_RST = 0)
- (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_4_APB_RST = 0)
- (CLK_RST_CONTROLLER_RST_DEV_PEX1_CORE_0.SWR_PEX1_CORE_5_APB_RST = 0)

The PAD driver sets private PIPE2UPHY parameters according to the platform specific configuration according to the platform specific configuration.

The PCIe driver sets up operation modes of each PCIe controller through its APB Client.

- Set the following address registers to set up the address region of the local configuration registers of each PCIe controllers.
  - APPL_CFG_BASE_ADDR.CFG_BASE_ADDR
- Set the following general control register field to set up the device type configuration of PCIe controllers.
  - APPL_DM_TYPE.DEVICE_TYPE to 'END_POINT' or 'ROOT_PORT'
- Set the following electromechnical register field to set the device presence status of PCIe controllers.
  - APPL_CTRL.SYS_PRE_DET_STATE

The PCIe driver enables clocks to PCIe controller.

- Set the following CAR register field to '0' to enable clocks to the PCIe controllers.
  - (CLK_RST_CONTROLLER_CLK_OUT_ENB_PEX0_CORE_0.CLK_ENB_PEX0_CORE_0 = 0)
  - (CLK_RST_CONTROLLER_CLK_OUT_ENB_PEX0_CORE_0.CLK_ENB_PEX0_CORE_1 = 0)
  - (CLK_RST_CONTROLLER_CLK_OUT_ENB_PEX0_CORE_0.CLK_ENB_PEX0_CORE_2 = 0)
  - (CLK_RST_CONTROLLER_CLK_OUT_ENB_PEX0_CORE_0.CLK_ENB_PEX0_CORE_3 = 0)
  - (CLK_RST_CONTROLLER_CLK_OUT_ENB_PEX0_CORE_0.CLK_ENB_PEX0_CORE_4 = 0)
  - (CLK_RST_CONTROLLER_CLK_OUT_ENB_PEX1_CORE_0.CLK_ENB_PEX1_CORE_5 = 0)

The PCIe driver de-asserts Reset to PCIe controller

- Set the following CAR register field to '0' to de-assert Reset to the PCIe controllers.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_0_RST = 0)
  - (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_1_RST = 0)
  - (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_2_RST = 0)
  - (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_3_RST = 0)
  - (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_4_RST = 0)
  - (CLK_RST_CONTROLLER_RST_DEV_PEX1_CORE_0.SWR_PEX1_CORE_5_RST = 0)

The PCIe driver programs PCIe controller to set up the frequency of the AUX_CLK to match the frequency of oscillator clock.

- Set the following PCIe controller register

- PF0_PORT_LOGIC_AUX_CLK_FREQ_OFF_0.AUX_CLK_FREQ

The PCIe driver programs the iATU of each PCIe controller to set up the address mapping of outbound and inbound requests as briefly described below.

- For Root Port, setup the one 32-bit address mapping for outbound configuration requests through PCIe controller extended configuration registers. Each PCI bus requires 1 MB address region.
- For Root Port, setup the one 32-bit address mapping for outbound I/O requests through PCIe controller extended configuration registers if I/O register is required or expected.
- For Root Port and Endpoint, setup the one 32-bit address mapping for outbound memory requests through PCIe controller extended configuration registers.
- For Root Port and Endpoint, setup the one 64-bit address mapping for outbound memory requests through PCIe controller extended configuration registers.
- For Endpoint, setup the one 64-bit address mapping for inbound memory requests through PCIe controller extended configuration registers.

9.3.4.2 PCIe Device Discovery and Enumeration

9.3.4.2.1 Root Port

The PCIe driver enables reference clock output to connecting to each PCIe controller through its APB Client.

- Set the following sideband registers to enable reference clock output of each PCIe controllers for platforms supporting clock management.
  - (APPL_PINMUX.CLKREQ_OUT_OVERRIDE = 0)
  - (APPL_PINMUX.CLKREQ_OUT_OVERRIDE_EN = 0)
- Or set the following sideband registers to enable reference clock output of each PCIe controllers for platforms that do not support clock management.
  - (APPL_PINMUX.CLKREQ_OUT_OVERRIDE = 1)
  - (APPL_PINMUX.CLKREQ_OUT_OVERRIDE_EN = 1)
- Wait 100 µs

The PCIe driver enables LTSSM of PCIe controller through its APB Client.

- Set the following registers to enable LTSSM.
  - (APPL_CTRL.APP_LTSSM_ENABLE = 1)

The PCIe driver de-asserts RESET# to the PCIe fabric connecting to each PCIe controller through its APB Client.

- Set the following sideband registers to de-assert RESET# of each PCIe controllers.

- (APPL_PINMUX.PEX_RST_O_N = 1)

The PCIe driver follows standard PCIe device discovery and enumeration procedure via depth first search to identify and setup all devices attach to the PCIe controllers and initialize their CFG registers.

9.3.4.2.2 Endpoint

PCIe driver change the Device ID and Class Code by programming the following configuration registers:

- Set the following register to enable override of read only bits.
  - (EP_PF0_PORT_LOGIC_MISC_CONTROL_1_OFF_0.DBI_RO_WR_EN = 1)
- Set the following register to change the Device ID of x4 and x8 controllers following DeviceID_master_list
  - (EP_PF0_TYPE0_HDR_DEVICE_ID_VENDOR_ID_REG_0.PCI_TYPE0_DEVICE_ID
- Set the following register to change the class code of x4 and x8 controllers following DeviceID_master_list
  - (PCIE_X4_EP_PF0_TYPE0_HDR_CLASS_CODE_REVISION_ID_0.BASE_CLASS_CODE
  - (PCIE_X4_EP_PF0_TYPE0_HDR_CLASS_CODE_REVISION_ID_0.SUBCLASS_CODE
  - (PCIE_X4_EP_PF0_TYPE0_HDR_CLASS_CODE_REVISION_ID_0.PROGRAM_INTERFACE
- Set the following register to set the memory BAR size of x4 and x8 controllers. The maximum size of the memory BAR is 32GB.
  - (PCIE_X4_EP_PF0_TYPE0_HDR_CLASS_CODE_BAR0_MASK_REG_0
  - (PCIE_X4_EP_PF0_TYPE0_HDR_CLASS_CODE_BAR1_MASK_REG_0
- Clear the following register to disable override of read only bits.
  - (EP_PF0_PORT_LOGIC_MISC_CONTROL_1_OFF_0.DBI_RO_WR_EN = 0)

PCIe driver enables LTSSM of PCIe controller through its APB Client.

- Set the following register to enable LTSSM.
  - (APPL_CTRL.APP_LTSSM_ENABLE = 1)

Device discovery and enumeration is performed by driver running on the PCIe root complex. PCIe driver waits the Interrupt informing device enumeration completion when the Memory Enable and Bus Master Enable bits in the PCI command configuration register are set.

- Set the following sideband registers to enable Interrupt reporting enable bits in Command register status change.
  - (APPL_INTR_EN_L0.SYS_INTR_EN = 1)
  - (APPL_INTR_EN_L0.PCI_CMD_EN_INT_EN = 1)

PCIe driver programs the PCI and PCIe configuration registers of the Endpoint to their default values when RESET# is asserted.

9.3.4.3 PCIe Controller Hot Reset

9.3.4.3.1 Root Port

The PCIe driver checks if the link-down Reset is triggered by secondary bus Reset.

- Program the following APB Client register fields to identify link-down Reset is due to secondary bus Reset.
  - (APPL_INTR_STATUS_L1_0.LINK_REQ_RST_NOT_CHGED = 1)
  - (APPL_INTR_STATUS_L1_0.SURPRISE_DOWN_ERR_STATE = 0)

The PCIe driver handles secondary bus Reset by setting the Reset override to only toggle core Reset.

- Program the following APB Client register fields to assert core Reset.
  - (PCIE_RP_CAR_RESET_OVRD_0.CYA_OVERRIDE_CORE_RST_N = 0)
- Program the following APB Client register fields to de-assert core Reset.
  - (PCIE_RP_CAR_RESET_OVRD_0.CYA_OVERRIDE_CORE_RST_N = 1)

9.3.4.3.2 Endpoint

The PCIe driver sets the following bit to trigger hot Reset:

- Program the following register fields to enable hot Reset.
  - (APPL_CTRL.HW_HOT_RESET_MODE = 2'b01)
  - (APPL_CTRL.HW_HOT_RESET_EN = 1)

9.3.4.4 Interrupt and Message Handling

9.3.4.4.1 Legacy Interrupts and Messages

Root Port

The PCIe driver enables Interrupt generation for receiving legacy Interrupts, error messages, and PME messages for each PCIe Root Port controller through its APB Client.

- Program the following general message reception and Interrupt register fields to '1' to enable Interrupt generation of each PCIe controllers.
  - (APPL_INTR_EN_L0.SYS_INTR_EN = 1)
  - (APPL_INTR_EN_L0.LINK_STATE_INT_EN = 1)
  - (APPL_INTR_EN_L1_0.SMLH_LINK_UP_INT_EN = 1)
  - (APPL_INTR_EN_L1_0.LINK_REQ_RST_NOT_INT_EN = 1)

- (APPL_INTR_EN_L1_0.SMLH_REQ_RST_NOT_INT_EN = 1)
- (APPL_INTR_EN_L1_0.SURPRISE_DOWN_ERR_INT_EN = 1)
- (APPL_INTR_EN_L0.LTR_RCV_INT_EN = 1)
- (APPL_INTR_EN_L0.VEN_MSG_RCV_INT_EN)
- (APPL_INTR_EN_L0.INT_INT_EN = 1)
- (APPL_INTR_EN_L1_8.CFG_PME_INT_EN = 1)
- (APPL_INTR_EN_L1_8.EDMA_INT_EN = 1)
- (APPL_INTR_EN_L1_8.INTX_EN = 1)
- (APPL_INTR_EN_L0.PM_MSG_INT_EN = 1)
- (APPL_INTR_EN_L1_9.RADM_PM_PME_INT_EN = 1)
- (APPL_INTR_EN_L1_9.RADM_PM_TO_ACK_INT_EN = 1)

Endpoint

The PCIe driver enables Interrupt generation for receiving PCIe messages for each PCIe Endpoint controller through its APB Client

- Program the following general message reception and Interrupt register fields to '1' to enable Interrupt generation of each PCIe controllers.
  - (APPL_INTR_EN_L0.SYS_INTR_EN = 1)
  - (APPL_INTR_EN_L0.LINK_STATE_INT_EN = 1)
  - (APPL_INTR_EN_L1_0.SMLH_LINK_UP_INT_EN = 1)
  - (APPL_INTR_EN_L1_0.LINK_REQ_RST_NOT_INT_EN = 1)
  - (APPL_INTR_EN_L1_0.SMLH_REQ_RST_NOT_INT_EN = 1)
  - (APPL_INTR_EN_L0.VEN_MSG_RCV_INT_EN = 1)
  - (APPL_INTR_EN_L1_6.VEN_MSG_RCV_INT_EN = 1)
  - (APPL_INTR_EN_L0.PM_MSG_INT_EN = 1)
  - (APPL_INTR_EN_L1_9.RADM_PM_TURNOFF_INT_EN = 1)
  - (APPL_INTR_EN_L0.UNLOCK_MSG_INT_EN = 1)
  - (APPL_INTR_EN_L1_10.RADM_MSG_UNLOCK_INT_EN = 1)

9.3.4.4.2 MSI (Message Signaled Interrupt)

Root Port

The PCIe driver enables Interrupt generation of received MSI for each PCIe Root Port controller through its APB Client.

- Program the following MSI register field to '1' to enable Interrupt generation of each PCIe controllers.
  - (APPL_INTR_EN_L0.MSI_RCV_INT_EN = 1)

For platforms having MSI forwarded to GIC, the forwarded MSI must be initialized in the following sequence.

- The Boot firmware sets up both Internal and external MSI address.
  - Internal MSI address = MSI_BASE
  - External MSI address = GIC_BASE + GICA_SETSPI
  - When the SMMU is enabled, the Boot firmware also sets up the (MSI Base Stream ID = 0x7F) to ensure the encoded MSI packets bypasses SMMU.
    - IOMMU region with IOMMU_RESV_MSI type is reserved with (Physical Address = GIC_BASE + GIC_SET_SPI).
    - Similar step as HISILICON is required to use IORT table to reserve the IOMMU_RESV_MSI region.
  - The TZ-protected registers have dependency on the PCIe power and Reset states. Thus SMC call is needed to program them dynamically in the Boot firmware.
- The Boot firmware passes “GIC_BASE” as MSI Base Address in the acpi_madt table, where the GIC (IOB) v2m driver parses “GIC_BASE” from the acpi_madt table
  - SCF decodes MSI address and data, and targets WRITE to GIC_SETSPI with MSI data.
- The Boot Firmware sets up PCIe devices with (MSI address = GIC_BASE + GICA_SETSPI), where the IO virtual address equals to the physical address regardless whether SMMU is enabled or disabled.

9.3.4.4.3 Error Reporting

The PCIe driver always enables PCIe AER as defined in PCIe specification. The PCIe driver also enables the PCIe configuration register checking when register protection is required.

Root Port

The PCIe driver enables Interrupt and fault generation for received error messages and errors detected by each PCIe Root Port controller through its APB Client.

- Program the following Interrupt register fields to ‘1’ to enable Interrupt generation for detected errors of each PCIe controllers.
  - (APPL_INTR_EN_L0.ERROR_INT_EN = 1)
  - (APPL_INTR_EN_L1_1.CFG_SYS_ERR_RC_INT_EN = 1)
  - (APPL_INTR_EN_L1_1.RADM_CORRECTABLE_ERR_INT_EN = 1)
  - (APPL_INTR_EN_L1_1.RADM_NONFATAL_ERR_INT_EN = 1)
  - (APPL_INTR_EN_L1_1.RADM_FATAL_ERR_INT_EN = 1)
  - (APPL_INTR_EN_L0.INT_INT_EN = 1)
  - (APPL_INTR_EN_L1_8.CFG_AER_RC_ERR_INT_EN = 1)
  - (APPL_INTR_EN_L0.RASDP_INT_EN = 1)
  - (APPL_INTR_EN_L1_12.MSTR_RASDP_ERR_MODE_INT_EN = 1)

    - (APPL_INTR_EN_L1_12.SLV_RASDP_ERR_MODE_INT_EN = 1)
    - (APPL_INTR_EN_L0.CPL_TIMEOUT_INT_EN = 1)
    - (APPL_INTR_EN_L0.PARITY_ERR_INT_EN = 1)
    - (APPL_INTR_EN_L0.CDM_REG_CHK_INT_EN = 1)
    - (APPL_INTR_EN_L1_18.CDM_REG_CHK_CMP_ERR_INT_EN = 1)
    - (APPL_INTR_EN_L1_18.CDM_REG_CHK_LOGIC_ERR_INT_EN = 1)
- Alternatively, program the following fault register fields to '1' to enable fault generation for detected errors of each PCIe controllers.
- (APPL_FAULT_EN_L0.ERROR_FAULT_EN = 1)
- (APPL_FAULT_EN_L1_1.CFG_SYS_ERR_RC_FAULT_EN = 1)
- (APPL_FAULT_EN_L1_1.RADM_CORRECTABLE_ERR_FAULT_EN = 1)
- (APPL_FAULT_EN_L1_1.RADM_NONFATAL_ERR_FAULT_EN = 1)
- (APPL_FAULT_EN_L1_1.RADM_FATAL_ERR_FAULT_EN = 1)
- (APPL_FAULT_EN_L0.INT_FAULT_EN = 1)
- (APPL_FAULT_EN_L1_8.CFG_AER_RC_ERR_FAULT_EN = 1)
- (APPL_FAULT_EN_L0.RASDP_FAULT_EN = 1)
- (APPL_FAULT_EN_L1_12.MSTR_RASDP_ERR_MODE_FAULT_EN = 1)
- (APPL_FAULT_EN_L1_12.SLV_RASDP_ERR_MODE_FAULT_EN = 1)
- (APPL_FAULT_EN_L0.CPL_TIMEOUT_FAULT_EN = 1)
- (APPL_FAULT_EN_L0.PARITY_ERR_FAULT_EN = 1)
- (APPL_FAULT_EN_L0.CDM_REG_CHK_FAULT_EN = 1)
- (APPL_FAULT_EN_L1_18.CDM_REG_CHK_CMP_ERR_FAULT_EN = 1)
- (APPL_FAULT_EN_L1_18.CDM_REG_CHK_LOGIC_ERR_FAULT_EN = 1)

Endpoint

The PCIe driver always enables PCIe AER. PCIe driver enables Interrupt and fault generation for errors detected by each PCIe Endpoint controller through its APB Client.

- Program the following Interrupt register fields to '1' to enable Interrupt generation for detected errors of each PCIe controllers.
    - (APPL_INTR_EN_L0.ERROR_INT_EN = 1)
    - (APPL_INTR_EN_L1_1.CFG_SYS_ERR_RC_INT_EN = 1)
    - (APPL_INTR_EN_L0.RASDP_INT_EN = 1)
    - (APPL_INTR_EN_L1_12.MSTR_RASDP_ERR_MODE_INT_EN = 1)
    - (APPL_INTR_EN_L1_12.SLV_RASDP_ERR_MODE_INT_EN = 1)
    - (APPL_INTR_EN_L0.CPL_TIMEOUT_INT_EN = 1)
    - (APPL_INTR_EN_L0.PARITY_ERR_INT_EN = 1)
    - (APPL_INTR_EN_L0.CDM_REG_CHK_INT_EN = 1)

    - (APPL_INTR_EN_L1_18.CDM_REG_CHK_CMP_ERR_INT_EN = 1)
    - (APPL_INTR_EN_L1_18.CDM_REG_CHK_LOGIC_ERR_INT_EN = 1)
- Alternatively, program the following fault register fields to '1' to enable fault generation for detected errors of each PCIe controllers.
- (APPL_FAULT_EN_L0.ERROR_FAULT_EN = 1)
- (APPL_FAULT_EN_L1_1.CFG_SYS_ERR_RC_FAULT_EN = 1)
- (APPL_FAULT_EN_L0.RASDP_FAULT_EN = 1)
- (APPL_FAULT_EN_L1_12.MSTR_RASDP_ERR_MODE_FAULT_EN = 1)
- (APPL_FAULT_EN_L1_12.SLV_RASDP_ERR_MODE_FAULT_EN = 1)
- (APPL_FAULT_EN_L0.CPL_TIMEOUT_FAULT_EN = 1)
- (APPL_FAULT_EN_L0.PARITY_ERR_FAULT_EN = 1)
- (APPL_FAULT_EN_L0.CDM_REG_CHK_FAULT_EN = 1)
- (APPL_FAULT_EN_L1_18.CDM_REG_CHK_CMP_ERR_FAULT_EN = 1)
- (APPL_FAULT_EN_L1_18.CDM_REG_CHK_LOGIC_ERR_FAULT_EN = 1)

9.3.4.5 PCIe Hierarchy Enumeration

Following PCI/PCIe specification, the PCIe hierarchy enumeration is performed via depth first search to discover all devices attach to the PCIe controllers and initialize their CFG registers.

The following register fields must be set to enable PCIe to forward DMA requests and accept MMIO and IOIO requests from CPU. For each PCIe controller:

- (T_PCIE2_RP_DEV_CTRL.BUS_MASTER = 1).
- (T_PCIE2_RP_DEV_CTRL.MEMORY_SPACE = 1).
- (T_PCIE2_RP_DEV_CTRL.IO_SPACE = 1).

The configuration register space of the PCIe hierarchy below each PCIe Root Port is accesses by using the AFI address mapping to convert the Orin MMIO addresses to PCIe configuration addresses. The following PCIe registers must be programmed accordingly for access configuration register spaces below the Root Ports. For each PCIe controller:

- T_PCIE2_RP_BN_LT.SEC_BUS_NUMBER
- T_PCIE2_RP_BN_LT.SUB_BUS_NUMBER

The IOIO register space of the PCIe hierarchy below each PCIe Root Port is accesses by using the AFI address mapping to convert the Orin MMIO addresses to PCIe IOIO addresses. The following PCIe registers must be programmed accordingly for access IOIO register spaces below the Root Ports. For each PCIe controller:

- T_PCIE2_RP_IO_BL_SS.T_PCIE2_RP_IO_BL_SSIO_BASE
- T_PCIE2_RP_IO_BL_SS.IO_LIMIT

The IOIO register space of the PCIe hierarchy below each PCIe Root Port is accesses by using the AFI address mapping to shift the Orin MMIO addresses to PCIe MMIO addresses. The following PCIe registers must be programmed accordingly for access MMIO register spaces below the Root Ports. For each PCIe controller:

- T_PCIE2_RP_MEM_BL.MEM_BASE
- T_PCIE2_RP_MEM_BL.MEM_LIMIT
- T_PCIE2_RP_PRE_BL.PREFETCH_MEM_BASE
- T_PCIE2_RP_PRE_BL.PREFETCH_MEM_LIMIT

9.3.4.6 Partition Power-Gating

The PCIe controllers are grouped in different power partitions. A PCIe conroller can only be put to power-gating, when the PCIe link of all the controllers in the same power partition are not connected, disabled, or in L2/3 link state.

In the case of PCIe link in disabled or L2/3 link states, toggling of RESET# is required to bring the link back to L0, when the is re-enumerated. Thus, instead of context save and restore, the PCIe driver is required to reinitialize the PCIe controller and the PCIe fabric (i.e. all th PCIe controllers and their external connected switches and devices).

9.3.4.6.1 Power Partitions

The 11 PCIe controllers are placed into nine power partitions:

- PCIEX4BA:
  - PCIe Controller C0 x4.
- PCIEX1A:
  - PCIe Controller C1 x1.
  - PCIe Controller C2 x1.
  - PCIe Controller C3 x1.
  - System backbone interconnects for C0, C1, C2, C3, and C4.
- PCIEX4BB:
  - PCIe Controller C4 x4.
- PCIEX8A:
  - PCIe Controller C5 x8.
  - System backbone interconnects for C4 and C4.
- PCIEX4A:
  - PCIe Controller C6 x4.
- PCIEX8B:

    - PCIe Controller C7 x8.
    - System backbone interconnects for C7, C8, C9, and C10.
- PCIEX4CA:
    - PCIe Controller C8 x4.
- PCIEX4CB:
    - PCIe Controller C9 x4.
- PCIEX4CC:
    - PCIe Controller C10 x4.

Among them,

PCIEX4A, PCIEX4BA, PCIEX4BB, PCIEX4CA, PCIEX4CB and PCIEX4C can be power-gated when the sole PCIe controller in the partition has its link in either L2 or disabled link power state.

PCIEX1A can only be power-gated when both PCIEX4BA and PCIEX4BB are already power-gated and PCIe C1/C2/C3 all have their links in either L2 or disabled link power state. PCIEX1A must be powered first before PCIEX4BA and/or PCIEX4BB can be brought out of power-gating.

PCIEX8A can only be power-gated when PCIEX4A is already power-gated and PCIe C5 has its link in either L2 or disabled link power state. PCIEX8A must be powered first before PCIEX4A can be brought out of power-gating.

PCIEX8B can only be power-gated when all PCIEX4CA, PCIEX4CB, and PCIEX4CC are already power-gated and PCIe C7 has its link in either L2 or disabled link power state. PCIEX8B must be powered first before PCIEX4CA, PCIEX4CB and/or PCIEX4CC can be brought out of power-gating.

9.3.4.6.2 Power-Gating Entry

The PCIe driver enables PE_WAKE# as an Interrupt source to signal wake event from PCIe devices for all PCIe Root Ports.

- Enable the Interrupt event through the GPIO registers.

    Refer to the GPIO Output/Bidirectional Configuration section of the GPIO Controllers chapter for the programming steps with <Port No> = L and <N> = 02.

The PCIe driver puts the active link to L2 for each PCIe controller in Root Port mode. For PCIe controllers in Endpoint mode, the link must be put to L2 by the Root Port before they can be power-gated.

- Program the Power Management State register of all connected PCIe devices to put them to D3 state.
    - (PCIE_X<*i*>_RC_PF0_PM_CAP_CON_STATUS_REG_0**.**POWER_STATE = 3), where <*i*> = 1, 4, 8.

- Program the PM register of the APB Client registers to trigger the generation of PME_Turn_Off message.
- (PCIE_RP_APPL_RADM_STATUS0_0.APPS_PM_XMT_TURNOFF_STATE = 1).
- Wait 10 ms, or check the PM register of the APB Client registers to ensure receipt of PME_TO_ACK message.
  - (PCIE_RP_APPL_INTR_STATUS_L1_9_0.RADM_PM_TO_ACK_STATE == 1).
- Wait 20 µs after the link is in L2 then check the following field of the APB Client register is set to ensure UPHY is properly powered down.
  - (PCIE_RP_APPL_DEBUG_0.PM_LINKST_IN_L2_LAT == 1).
- Set the following fields of the APB Client registers to '1' to allow reference clock PLL to be powered down.
  - (PCIE_RP_APPL_PINMUX_0.CLKREQ_OVERRIDE_EN = 1).
  - (PCIE_RP_APPL_PINMUX_0.CLKREQ_OVERRIDE = 1).

The PCIe driver checks the link has entered L2, then forces UPHY to low power state using softare verrides.

- Check the following field of the APB Client registers to confirm the link has entered L2.
  - (PCIE_RP_APPL_DEBUG_0.PM_LINKST_IN_L2_LAT == 1).

The System Power Management driver flushes all the PCIe controllers in each PCIe power partition and their associated MCCIF (Memory Controller Connect InterFace).

- Set the following MC register fields to '1' to enable flush of PCIe C0 in PCIEX4BA.
  - (MC_CLIENT_HOTRESET_CTRL_2_0.PCIE0A_FLUSH_ENABLE = 1).
  - (MC_CLIENT_HOTRESET_CTRL_2_0.PCIE0A2_FLUSH_ENABLE = 1).
- Poll the following fields for '1' to ensure flushes are completed for PCIe C0 in PCIEX4BA.
  - (MC_CLIENT_HOTRESET_STATUS_2_0.PCIE0A_HOTRESET_STATUS == 1).
  - (MC_CLIENT_HOTRESET_STATUS_2_0.PCIE0A2_HOTRESET_STATUS == 1).
- Set the following MC register fields to '1' to enable flush for PCIe C1, C2, and C3 in PCIEX1A.
  - (MC_CLIENT_HOTRESET_CTRL_2_0.PCIE_FLUSH_ENABLE = 1).
  - (MC_CLIENT_HOTRESET_CTRL_2_0.PCIE2A_FLUSH_ENABLE = 1).
  - (MC_CLIENT_HOTRESET_CTRL_2_0.PCIE3_FLUSH_ENABLE = 1).
  - (MC_CLIENT_HOTRESET_CTRL_2_0.PCIE3A_FLUSH_ENABLE = 1).
- Poll the following fields for '1' to ensure flushes are completed for PCIe C1, C2, and C3 in PCIEX1A.
  - (MC_CLIENT_HOTRESET_STATUS_2_0.PCIE_HOTRESET_STATUS == 1).
  - (MC_CLIENT_HOTRESET_STATUS_2_0.PCIE2A_HOTRESET_STATUS == 1).
  - (MC_CLIENT_HOTRESET_STATUS_2_0.PCIE3_HOTRESET_STATUS == 1).
  - (MC_CLIENT_HOTRESET_STATUS_2_0.PCIE3A_HOTRESET_STATUS == 1).

- Set the following MC register field to '1' to enable flush for PCIe C4 in PCIEX4BB.
  - (MC_CLIENT_HOTRESET_CTRL_2_0.PCIE4A_FLUSH_ENABLE = 1).
- Poll the following field for '1' to ensure flushes are completed for PCIe C4 in PCIEX4BB.
  - (MC_CLIENT_HOTRESET_STATUS_2_0.PCIE4A_HOTRESET_STATUS == 1).
- Set the following MC register field to '1' to enable flush for PCIe C5 in PCIEX8A.
  - (MC_CLIENT_HOTRESET_CTRL_2_0.PCIE5A_FLUSH_ENABLE = 1).
- Poll the following field for '1' to ensure flushes are completed for PCIe C5 in PCIEX8A.
  - (MC_CLIENT_HOTRESET_STATUS_2_0.PCIE5A_HOTRESET_STATUS == 1).
- Set the following MC register field to '1' to enable flush for PCIe C6 in PCIEX4A.
  - (MC_CLIENT_HOTRESET_CTRL_3_0.PCIE6A_FLUSH_ENABLE = 1).
- Poll the following field for '1' to ensure flushes are completed for PCIe C6 in PCIEX4A.
  - (MC_CLIENT_HOTRESET_STATUS_3_0.PCIE6A_HOTRESET_STATUS == 1).
- Set the following MC register field to '1' to enable flush for PCIe C7 in PCIEX8B.
  - (MC_CLIENT_HOTRESET_CTRL_3_0.PCIE7A_FLUSH_ENABLE = 1).
- Poll the following field for '1' to ensure flushes are completed for PCIe C7 in PCIEX8B.
  - (MC_CLIENT_HOTRESET_STATUS_3_0.PCIE7A_HOTRESET_STATUS == 1).
- Set the following MC register field to '1' to enable flush for PCIe C8 in PCIEX4CA.
  - (MC_CLIENT_HOTRESET_CTRL_3_0.PCIE8A_FLUSH_ENABLE = 1).
- Poll the following field for '1' to ensure flushes are completed for PCIe C8 in PCIEX4CA.
  - (MC_CLIENT_HOTRESET_STATUS_3_0.PCIE8A_HOTRESET_STATUS == 1).
- Set the following MC register field to '1' to enable flush for PCIe C9 in PCIEX4CB.
  - (MC_CLIENT_HOTRESET_CTRL_3_0.PCIE9A_FLUSH_ENABLE = 1).
- Poll the following field for '1' to ensure flushes are completed for PCIe C9 in PCIEX4CB.
  - (MC_CLIENT_HOTRESET_STATUS_3_0.PCIE9A_HOTRESET_STATUS == 1).
- Set the following MC register field to '1' to enable flush for PCIe C10 in PCIEX4CC.
  - (MC_CLIENT_HOTRESET_CTRL_3_0.PCIE10A_FLUSH_ENABLE = 1).
- Poll the following field for '1' to ensure flushes are completed for PCIe C10 in PCIEX4CC.
  - (MC_CLIENT_HOTRESET_STATUS_3_0.PCIE10A_HOTRESET_STATUS == 1).

NOTE:

For details of the MC registers, please refer to the Memory SubSystem (MSS) chapter.

The System Power Management driver enables clamps of all the PCIe power partitions.

- Set the following PMC register field to '1' to enable clamps for PCIEX4BA.
  - (PMC_IMPL_PART_PCIEX4BA_CLAMP_CONTROL_0.CLAMP = 1).

- Set the following PMC register field to '1' to enable clamps for PCIEX1A.
  - PMC_IMPL_PART_PCIEX1A_CLAMP_CONTROL_0.CLAMP = 1).
- Set the following PMC register field to '1' to enable clamps for PCIEX4BB.
  - (PMC_IMPL_PART_PCIEX4BB_CLAMP_CONTROL_0.CLAMP = 1).
- Set the following PMC register field to '1' to enable clamps for PCIEX8A.
  - (PMC_IMPL_PART_PCIEX8A_CLAMP_CONTROL_0.CLAMP = 1).
- Set the following PMC register field to '1' to enable clamps for PCIEX4A.
  - (PMC_IMPL_PART_PCIEX4A_CLAMP_CONTROL_0.CLAMP = 1).
- Set the following PMC register field to '1' to enable clamps for PCIEX8B.
  - (PMC_IMPL_PART_PCIEX8B_CLAMP_CONTROL_0.CLAMP = 1).
- Set the following PMC register field to '1' to enable clamps for PCIEX4CA.
  - (PMC_IMPL_PART_PCIEX4CA_CLAMP_CONTROL_0.CLAMP = 1).
- Set the following PMC register field to '1' to enable clamps for PCIEX4CB.
  - (PMC_IMPL_PART_PCIEX4CB_CLAMP_CONTROL_0.CLAMP = 1).
- Set the following PMC register field to '1' to enable clamps for PCIEX4CC.
  - (PMC_IMPL_PART_PCIEX4C_CLAMP_CONTROL_0.CLAMP = 1).

NOTE:

For details of the PMC registers, please refer to the Power management Controller (PMC) chapter.

The System Power Management driver Wait 200 ns.

The System Power Management driver asserts Reset to PCIe then disables clocks for all the PCIe power partitions.

- Set the following CAR (Clock and Reset) register fields to '1' to assert Reset to PCIe C0 in PCIEX4BA.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_0_RST = 1).
  - (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_0_APB_RST = 1).
- Set the following CAR (Clock and Reset) register field to '1' to disable the clocks to PCIe C0 in PCIEX4BA.
  - (CLK_RST_CONTROLLER_CLK_OUT_ENB_PEX0_CORE_0.CLK_ENB_PEX0_CORE_0 = 1).
- Set the following CAR (Clock and Reset) register fields to '1' to assert Reset to PCIe C1, C2, and C3 in PCIEX1A.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_1_RST = 1).
  - (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_1_APB_RST = 1).
  - (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_2_RST = 1).
  - (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_2_APB_RST = 1).

  - (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_3_RST = 1).
  - (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_3_APB_RST = 1).
- Set the following CAR (Clock and Reset) register field to '1' to disable the clocks to PCIe C1, C2, and C3 in PCIEX1A.
  - (CLK_RST_CONTROLLER_CLK_OUT_ENB_PEX0_CORE_0.CLK_ENB_PEX0_CORE_1 = 1).
  - CLK_RST_CONTROLLER_CLK_OUT_ENB_PEX0_CORE_0.CLK_ENB_PEX0_CORE_2 = 1).
  - (CLK_RST_CONTROLLER_CLK_OUT_ENB_PEX0_CORE_0.CLK_ENB_PEX0_CORE_3 = 1).
- Set the following CAR (Clock and Reset) register fields to '1' to assert Reset to PCIe C4 in PCIEX4BB.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_4_RST = 1).
  - (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_4_APB_RST = 1).
- Set the following CAR (Clock and Reset) register field to '1' to disable the clocks to PCIe C4 in PCIEX4BB.
  - (CLK_RST_CONTROLLER_CLK_OUT_ENB_PEX0_CORE_0.CLK_ENB_PEX0_CORE_4 = 1).
- Set the following CAR (Clock and Reset) register fields to '1' to assert Reset to PCIe C5 in PCIEX8A.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX1_CORE_0.SWR_PEX1_CORE_5_RST = 1).
  - (CLK_RST_CONTROLLER_RST_DEV_PEX1_CORE_0.SWR_PEX1_CORE_5_APB_RST = 1).
- Set the following CAR (Clock and Reset) register field to '1' to disable the clocks to PCIe C5 in PCIEX8A.
  - (CLK_RST_CONTROLLER_CLK_OUT_ENB_PEX1_CORE_0.CLK_ENB_PEX1_CORE_5 = 1).
- Set the following CAR (Clock and Reset) register fields to '1' to assert Reset to PCIe C6 in PCIEX4A.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX1_CORE_0.SWR_PEX1_CORE_6_RST = 1).
  - (CLK_RST_CONTROLLER_RST_DEV_PEX1_CORE_0.SWR_PEX1_CORE_6_APB_RST = 1).
- Set the following CAR (Clock and Reset) register field to '1' to disable the clocks to PCIe C6 in PCIEX4A.
  - (CLK_RST_CONTROLLER_CLK_OUT_ENB_PEX1_CORE_0.CLK_ENB_PEX1_CORE_6 = 1).
- Set the following CAR (Clock and Reset) register fields to '1' to assert Reset to PCIE C7 in PCIEX8B.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX2_CORE_0.SWR_PEX2_CORE_7_RST = 1).
  - (CLK_RST_CONTROLLER_RST_DEV_PEX2_CORE_0.SWR_PEX2_CORE_7_APB_RST = 1).
- Set the following CAR (Clock and Reset) register field to '1' to disable the clocks to PCIe C7 in PCIEX8B.
  - (CLK_RST_CONTROLLER_CLK_OUT_ENB_PEX1_CORE_0.CLK_ENB_PEX2_CORE_7 = 1).
- Set the following CAR (Clock and Reset) register fields to '1' to assert Reset to PCIe C8 in PCIEX4CA.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX2_CORE_0.SWR_PEX2_CORE_8_RST = 1).

  - (CLK_RST_CONTROLLER_RST_DEV_PEX2_CORE_0.SWR_PEX2_CORE_8_APB_RST = 1).
- Set the following CAR (Clock and Reset) register field to '1' to disable the clocks to PCIe C8 in PCIEX4CA.
  - (CLK_RST_CONTROLLER_CLK_OUT_ENB_PEX1_CORE_0.CLK_ENB_PEX2_CORE_8 = 1).
- Set the following CAR (Clock and Reset) register fields to '1' to assert Reset to PCIe C9 in PCIEX4CB.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX2_CORE_0.SWR_PEX2_CORE_9_RST = 1).
  - (CLK_RST_CONTROLLER_RST_DEV_PEX2_CORE_0.SWR_PEX2_CORE_9_APB_RST = 1).
- Set the following CAR (Clock and Reset) register field to '1' to disable the clocks to PCIe C9 in PCIEX4CB.
  - (CLK_RST_CONTROLLER_CLK_OUT_ENB_PEX1_CORE_0.CLK_ENB_PEX2_CORE_9 = 1).
- Set the following CAR (Clock and Reset) register fields to '1' to assert Reset to PCIe C10 in PCIEX4CC.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX2_CORE_0.SWR_PEX2_CORE_10_RST = 1).
  - (CLK_RST_CONTROLLER_RST_DEV_PEX2_CORE_0.SWR_PEX2_CORE_10_APB_RST = 1).
- Set the following CAR (Clock and Reset)register field to '1' to disable the clocks to PCIe C10 in PCIEX4CC.
  - (CLK_RST_CONTROLLER_CLK_OUT_ENB_PEX1_CORE_0.CLK_ENB_PEX2_CORE_10 = 1).

NOTE:

For details of the CAR registers, please refer to the Clock and Reset Controller (CAR) chapter.

The System Power Management driver removes power to all the PCIe power partitions.

- Set the following PMC register fields to power-gate PCIEX4BA.
  - (PMC_IMPL_PART_PCIEX4BA_POWER_GATE_CONTROL_0.INTER_PART_DELAY_EN = ENABLE).
  - (PMC_IMPL_PART_PCIEX4BA_POWER_GATE_CONTROL_0.SRAM_SLEEP_EN = ON).
  - (PMC_IMPL_PART_PCIEX4BA_POWER_GATE_CONTROL_0.SRAM_RET_EN = OFF).
  - (PMC_IMPL_PART_PCIEX4BA_POWER_GATE_CONTROL_0.LOGIC_SLEEP = ON).
  - (PMC_IMPL_PART_PCIEX4BA_POWER_GATE_CONTROL_0.START = 1).
- Poll the following PMC register field to become '0' to ensure power rail is disabled for PCIEX4BA.
  - (PMC_IMPL_PART_PCIEX4BA_POWER_GATE_CONTROL_0.START == 0).
- Read the following PMC register fields to confirm the power-gating status of partition PCIEX4BB.
  - (PMC_IMPL_PART_PCIEX4BB_POWER_GATE_STATUS_0.SRAM_SLEEP_STS == ON).
  - (PMC_IMPL_PART_PCIEX4BB_POWER_GATE_CONTROL_0.SRAM_RET_STS == OFF).

  - (PMC_IMPL_PART_PCIEX4BB_POWER_GATE_CONTROL_0.LOGIC_SLEEP_STS == ON).
- Set the following PMC register fields to power-gate PCIEX8A.
  - (PMC_IMPL_PART_PCIEX8A_POWER_GATE_CONTROL_0.INTER_PART_DELAY_EN = ENABLE).
  - (PMC_IMPL_PART_PCIEX8A_POWER_GATE_CONTROL_0.SRAM_SLEEP_EN = ON).
  - (PMC_IMPL_PART_PCIEX8A_POWER_GATE_CONTROL_0.SRAM_RET_EN = OFF).
  - (PMC_IMPL_PART_PCIEX8A_POWER_GATE_CONTROL_0.LOGIC_SLEEP = ON).
  - (PMC_IMPL_PART_PCIEX8A_POWER_GATE_CONTROL_0.START = 1).
- Poll the following PMC register field to become '0' to ensure power rail is disabled for PCIEX8A.
  - (PMC_IMPL_PART_PCIEX8A_POWER_GATE_CONTROL_0.START == 0).
- Read the following PMC register fields to confirm the power-gating status of partition PCIEX8A.
  - (PMC_IMPL_PART_PCIEX8A_POWER_GATE_STATUS_0.SRAM_SLEEP_STS == ON).
  - (PMC_IMPL_PART_PCIEX8A_POWER_GATE_CONTROL_0.SRAM_RET_STS == OFF).
  - (PMC_IMPL_PART_PCIEX8A_POWER_GATE_CONTROL_0.LOGIC_SLEEP_STS == ON).
- Set the following PMC register fields to power-gate PCIEX4A.
  - (PMC_IMPL_PART_PCIEX4A_POWER_GATE_CONTROL_0.INTER_PART_DELAY_EN = ENABLE).
  - (PMC_IMPL_PART_PCIEX4A_POWER_GATE_CONTROL_0.SRAM_SLEEP_EN = ON).
  - (PMC_IMPL_PART_PCIEX4A_POWER_GATE_CONTROL_0.SRAM_RET_EN = OFF).
  - (PMC_IMPL_PART_PCIEX4A_POWER_GATE_CONTROL_0.LOGIC_SLEEP = ON).
  - (PMC_IMPL_PART_PCIEX4A_POWER_GATE_CONTROL_0.START = 1).
- Poll the following PMC register field to become '0' to ensure power rail is disabled for PCIEX4A.
  - (PMC_IMPL_PART_PCIEX4A_POWER_GATE_CONTROL_0.START == 0).
- Read the following PMC register fields to confirm the power-gating status of partition PCIEX4A.
  - (PMC_IMPL_PART_PCIEX4A_POWER_GATE_STATUS_0.SRAM_SLEEP_STS == ON).
  - (PMC_IMPL_PART_PCIEX4A_POWER_GATE_CONTROL_0.SRAM_RET_STS == OFF).
  - (PMC_IMPL_PART_PCIEX4A_POWER_GATE_CONTROL_0.LOGIC_SLEEP_STS == ON).
- Set the following PMC register fields to power-gate PCIEX8B.
  - (PMC_IMPL_PART_PCIEX8B_POWER_GATE_CONTROL_0.INTER_PART_DELAY_EN = ENABLE).
  - (PMC_IMPL_PART_PCIEX8B_POWER_GATE_CONTROL_0.SRAM_SLEEP_EN = ON).
  - (PMC_IMPL_PART_PCIEX8B_POWER_GATE_CONTROL_0.SRAM_RET_EN = OFF).
  - (PMC_IMPL_PART_PCIEX8B_POWER_GATE_CONTROL_0.LOGIC_SLEEP = ON).
  - (PMC_IMPL_PART_PCIEX8B_POWER_GATE_CONTROL_0.START = 1).

- Poll the following PMC register field to become '0' to ensure power rail is disabled for PCIEX8B.
  - (PMC_IMPL_PART_PCIEX8B_POWER_GATE_CONTROL_0.START == 0).
- Read the following PMC register fields to confirm the power-gating status of partition PCIEX8B.
  - (PMC_IMPL_PART_PCIEX8B_POWER_GATE_STATUS_0.SRAM_SLEEP_STS == ON).
  - (PMC_IMPL_PART_PCIEX8B_POWER_GATE_CONTROL_0.SRAM_RET_STS == OFF).
  - (PMC_IMPL_PART_PCIEX8B_POWER_GATE_CONTROL_0.LOGIC_SLEEP_STS == ON).
- Set the following PMC register fields to power-gate PCIEX4CA.
  - (PMC_IMPL_PART_PCIEX4CA_POWER_GATE_CONTROL_0.INTER_PART_DELAY_EN = ENABLE).
  - (PMC_IMPL_PART_PCIEX4CA_POWER_GATE_CONTROL_0.SRAM_SLEEP_EN = ON).
  - (PMC_IMPL_PART_PCIEX4CA_POWER_GATE_CONTROL_0.SRAM_RET_EN = OFF).
  - (PMC_IMPL_PART_PCIEX4CA_POWER_GATE_CONTROL_0.LOGIC_SLEEP = ON).
  - (PMC_IMPL_PART_PCIEX4CA_POWER_GATE_CONTROL_0.START = 1).
- Poll the following PMC register field to become '0' to ensure power rail is disabled for PCIEX4CA.
  - (PMC_IMPL_PART_PCIEX4CA_POWER_GATE_CONTROL_0.START == 0).
- Read the following PMC register fields to confirm the power-gating status of partition PCIEX4CA.
  - (PMC_IMPL_PART_PCIEX4CA_POWER_GATE_STATUS_0.SRAM_SLEEP_STS == ON).
  - (PMC_IMPL_PART_PCIEX4CA_POWER_GATE_CONTROL_0.SRAM_RET_STS == OFF).
  - (PMC_IMPL_PART_PCIEX4CA_POWER_GATE_CONTROL_0.LOGIC_SLEEP_STS == ON).
- Set the following PMC register fields to power-gate PCIEX4CB.
  - (PMC_IMPL_PART_PCIEX4CB_POWER_GATE_CONTROL_0.INTER_PART_DELAY_EN = ENABLE).
  - (PMC_IMPL_PART_PCIEX4CB_POWER_GATE_CONTROL_0.SRAM_SLEEP_EN = ON).
  - (PMC_IMPL_PART_PCIEX4CB_POWER_GATE_CONTROL_0.SRAM_RET_EN = OFF).
  - (PMC_IMPL_PART_PCIEX4CB_POWER_GATE_CONTROL_0.LOGIC_SLEEP = ON).
  - (PMC_IMPL_PART_PCIEX4CB_POWER_GATE_CONTROL_0.START = 1)'.
- Poll the following PMC register field to become '0' to ensure power rail is disabled for PCIEX4CB.
  - (PMC_IMPL_PART_PCIEX4CB_POWER_GATE_CONTROL_0.START == 0).
- Read the following PMC register fields to confirm the power-gating status of partition PCIEX4CB.
  - (PMC_IMPL_PART_PCIEX4CB_POWER_GATE_STATUS_0.SRAM_SLEEP_STS == ON).
  - (PMC_IMPL_PART_PCIEX4CB_POWER_GATE_CONTROL_0.SRAM_RET_STS == OFF).
  - (PMC_IMPL_PART_PCIEX4CB_POWER_GATE_CONTROL_0.LOGIC_SLEEP_STS == ON).

- Set the following PMC register fields to power-gate PCIEX4CC.
  - (PMC_IMPL_PART_PCIEX4CC_POWER_GATE_CONTROL_0.INTER_PART_DELAY_EN = ENABLE)
  - (PMC_IMPL_PART_PCIEX4CC_POWER_GATE_CONTROL_0.SRAM_SLEEP_EN = ON).
  - (PMC_IMPL_PART_PCIEX4CC_POWER_GATE_CONTROL_0.SRAM_RET_EN = OFF).
  - (PMC_IMPL_PART_PCIEX4CC_POWER_GATE_CONTROL_0.LOGIC_SLEEP = ON).
  - (PMC_IMPL_PART_PCIEX4CC_POWER_GATE_CONTROL_0.START = 1).
- Poll the following PMC register field to become '0' to ensure power rail is disabled for PCIEX4CC.
  - (PMC_IMPL_PART_PCIEX4CC_POWER_GATE_CONTROL_0.START == 0).
- Read the following PMC register fields to confirm the power-gating status of partition PCIEX4CC.
  - (PMC_IMPL_PART_PCIEX4CC_POWER_GATE_STATUS_0.SRAM_SLEEP_STS == ON).
  - (PMC_IMPL_PART_PCIEX4CC_POWER_GATE_CONTROL_0.SRAM_RET_STS == OFF).
  - (PMC_IMPL_PART_PCIEX4CC_POWER_GATE_CONTROL_0.LOGIC_SLEEP_STS == ON).

NOTE:

For details of the PMC registers, please refer to the Power management Controller (PMC) chapter.

9.3.4.6.3 Power-Gating Exit

The System Power Management driver restores power to all the PCIe power partitions.

- Set the following PMC register fields to ungate power to PCIe power partition PCIEX4BA.
  - (PMC_IMPL_PART_PCIEX4BA_POWER_GATE_CONTROL_0.SRAM_SLEEP_EN = OFF).
  - (PMC_IMPL_PART_PCIEX4BA_POWER_GATE_CONTROL_0.SRAM_RET_EN = OFF).
  - (PMC_IMPL_PART_PCIEX4BA_POWER_GATE_CONTROL_0.LOGIC_SLEEP = OFF).
  - (PMC_IMPL_PART_PCIEX4BA_POWER_GATE_CONTROL_0.START = 1).
- Poll the following PMC register field to become '0' to ensure power rail is enabled for PCIEX4BA.
  - (PMC_IMPL_PART_PCIEX4BA_POWER_GATE_CONTROL_0.START == 0).
- Read the following PMC register fields to confirm the power-gating status of partition PCIEX4BA.
  - (PMC_IMPL_PART_PCIEX4BA_POWER_GATE_STATUS_0.SRAM_SLEEP_STS == OFF).
  - (PMC_IMPL_PART_PCIEX4BA_POWER_GATE_CONTROL_0.SRAM_RET_STS == OFF).
  - (PMC_IMPL_PART_PCIEX4BA_POWER_GATE_CONTROL_0.LOGIC_SLEEP_STS == OFF).
- Set the following PMC register fields to ungate power to PCIe power partition PCIEX1A.
  - (PMC_IMPL_PART_PCIEX1A_POWER_GATE_CONTROL_0.SRAM_SLEEP_EN = OFF).

  - (PMC_IMPL_PART_PCIEX1A_POWER_GATE_CONTROL_0.SRAM_RET_EN = OFF).
  - (PMC_IMPL_PART_PCIEX1A_POWER_GATE_CONTROL_0.LOGIC_SLEEP = OFF).
  - (PMC_IMPL_PART_PCIEX1A_POWER_GATE_CONTROL_0.START = 1).
- Poll the following PMC register field to become '0' to ensure power rail is enabled for PCIEX1A.
  - (PMC_IMPL_PART_PCIEX1A_POWER_GATE_CONTROL_0.START == 0).
- Read the following PMC register fields to confirm the power-gating status of partition PCIEX1A.
  - (PMC_IMPL_PART_PCIEX1A_POWER_GATE_STATUS_0.SRAM_SLEEP_STS == OFF).
  - (PMC_IMPL_PART_PCIEX1A_POWER_GATE_CONTROL_0.SRAM_RET_STS == OFF).
  - (PMC_IMPL_PART_PCIEX1A_POWER_GATE_CONTROL_0.LOGIC_SLEEP_STS == OFF).
- Set the following PMC register fields to ungate power to PCIe power partition PCIEX4BB.
  - (PMC_IMPL_PART_PCIEX4BB_POWER_GATE_CONTROL_0.SRAM_SLEEP_EN = OFF).
  - (PMC_IMPL_PART_PCIEX4BB_POWER_GATE_CONTROL_0.SRAM_RET_EN = OFF).
  - (PMC_IMPL_PART_PCIEX4BB_POWER_GATE_CONTROL_0.LOGIC_SLEEP = OFF).
  - (PMC_IMPL_PART_PCIEX4BB_POWER_GATE_CONTROL_0.START = 1).
- Poll the following PMC register field to become '0' to ensure power rail is enabled for PCIEX4BB.
  - (PMC_IMPL_PART_PCIEX4BB_POWER_GATE_CONTROL_0.START == 0).
- Read the following PMC register fields to confirm the power-gating status of partition PCIEX4BB.
  - (PMC_IMPL_PART_PCIEX4BB_POWER_GATE_STATUS_0.SRAM_SLEEP_STS == OFF).
  - (PMC_IMPL_PART_PCIEX4BB_POWER_GATE_CONTROL_0.SRAM_RET_STS == OFF).
  - (PMC_IMPL_PART_PCIEX4BB_POWER_GATE_CONTROL_0.LOGIC_SLEEP_STS == OFF).
- Set the following PMC register fields to ungate power to PCIe power partition PCIEX8A.
  - (PMC_IMPL_PART_PCIEX8A_POWER_GATE_CONTROL_0.SRAM_SLEEP_EN = OFF).
  - (PMC_IMPL_PART_PCIEX8A_POWER_GATE_CONTROL_0.SRAM_RET_EN = OFF).
  - (PMC_IMPL_PART_PCIEX8A_POWER_GATE_CONTROL_0.LOGIC_SLEEP = OFF).
  - (PMC_IMPL_PART_PCIEX8A_POWER_GATE_CONTROL_0.START = 1).
- Poll the following PMC register field to become '0' to ensure power rail is enabled for PCIEX8A.
  - (PMC_IMPL_PART_PCIEX8A_POWER_GATE_CONTROL_0.START == 0).
- Read the following PMC register fields to confirm the power-gating status of partition PCIEX8A.
  - (PMC_IMPL_PART_PCIEX8A_POWER_GATE_STATUS_0.SRAM_SLEEP_STS == OFF).
  - (PMC_IMPL_PART_PCIEX8A_POWER_GATE_CONTROL_0.SRAM_RET_STS == OFF).
  - (PMC_IMPL_PART_PCIEX8A_POWER_GATE_CONTROL_0.LOGIC_SLEEP_STS == OFF).

- Set the following PMC register fields to ungate power to PCIe power partition PCIEX4A.
  - (PMC_IMPL_PART_PCIEX4A_POWER_GATE_CONTROL_0.SRAM_SLEEP_EN = OFF).
  - (PMC_IMPL_PART_PCIEX4A_POWER_GATE_CONTROL_0.SRAM_RET_EN = OFF).
  - (PMC_IMPL_PART_PCIEX4A_POWER_GATE_CONTROL_0.LOGIC_SLEEP = OFF).
  - (PMC_IMPL_PART_PCIEX4A_POWER_GATE_CONTROL_0.START = 1).
- Poll the following PMC register field to become '0' to ensure power rail is enabled for PCIEX4A.
  - (PMC_IMPL_PART_PCIEX4A_POWER_GATE_CONTROL_0.START == 0).
- Read the following PMC register fields to confirm the power-gating status of partition PCIEX4A.
  - (PMC_IMPL_PART_PCIEX4A_POWER_GATE_STATUS_0.SRAM_SLEEP_STS == OFF).
  - (PMC_IMPL_PART_PCIEX4A_POWER_GATE_CONTROL_0.SRAM_RET_STS == OFF).
  - (PMC_IMPL_PART_PCIEX4A_POWER_GATE_CONTROL_0.LOGIC_SLEEP_STS == OFF).
- Set the following PMC register fields to ungate power to PCIe power partition PCIEX8B.
  - (PMC_IMPL_PART_PCIEX8B_POWER_GATE_CONTROL_0.SRAM_SLEEP_EN = OFF).
  - (PMC_IMPL_PART_PCIEX8B_POWER_GATE_CONTROL_0.SRAM_RET_EN = OFF).
  - (PMC_IMPL_PART_PCIEX8B_POWER_GATE_CONTROL_0.LOGIC_SLEEP = OFF).
  - (PMC_IMPL_PART_PCIEX8B_POWER_GATE_CONTROL_0.START = 1).
- Poll the following PMC register field to become '0' to ensure power rail is enabled for PCIEX8B.
  - (PMC_IMPL_PART_PCIEX8B_POWER_GATE_CONTROL_0.START == 0).
- Read the following PMC register fields to confirm the power-gating status of partition PCIEX8B.
  - (PMC_IMPL_PART_PCIEX8B_POWER_GATE_STATUS_0.SRAM_SLEEP_STS == OFF).
  - (PMC_IMPL_PART_PCIEX8B_POWER_GATE_CONTROL_0.SRAM_RET_STS == OFF).
  - (PMC_IMPL_PART_PCIEX8B_POWER_GATE_CONTROL_0.LOGIC_SLEEP_STS == OFF).
- Set the following PMC register fields to ungate power to PCIe power partition PCIEX4CA.
  - (PMC_IMPL_PART_PCIEX4CA_POWER_GATE_CONTROL_0.SRAM_SLEEP_EN = OFF).
  - (PMC_IMPL_PART_PCIEX4CA_POWER_GATE_CONTROL_0.SRAM_RET_EN = OFF).
  - (PMC_IMPL_PART_PCIEX4CA_POWER_GATE_CONTROL_0.LOGIC_SLEEP = OFF).
  - (PMC_IMPL_PART_PCIEX4CA_POWER_GATE_CONTROL_0.START = 1).
- Poll the following PMC register field to become '0' to ensure power rail is enabled for PCIEX4CA.
  - (PMC_IMPL_PART_PCIEX4CA_POWER_GATE_CONTROL_0.START == 0).
- Read the following PMC register fields to confirm the power-gating status of partition PCIEX4CA.
  - (PMC_IMPL_PART_PCIEX4CA_POWER_GATE_STATUS_0.SRAM_SLEEP_STS == OFF).
  - (PMC_IMPL_PART_PCIEX4CA_POWER_GATE_CONTROL_0.SRAM_RET_STS == OFF).

  - (PMC_IMPL_PART_PCIEX4CA_POWER_GATE_CONTROL_0.LOGIC_SLEEP_STS == OFF).
- Set the following PMC register fields to ungate power to PCIe power partition PCIEX4CB.
  - (PMC_IMPL_PART_PCIEX4CB_POWER_GATE_CONTROL_0.SRAM_SLEEP_EN = OFF).
  - (PMC_IMPL_PART_PCIEX4CB_POWER_GATE_CONTROL_0.SRAM_RET_EN = OFF).
  - (PMC_IMPL_PART_PCIEX4CB_POWER_GATE_CONTROL_0.LOGIC_SLEEP = OFF).
  - (PMC_IMPL_PART_PCIEX4CB_POWER_GATE_CONTROL_0.START = 1).
- Poll the following PMC register field to become '0' to ensure power rail is enabled for PCIEX4CB.
  - (PMC_IMPL_PART_PCIEX4CB_POWER_GATE_CONTROL_0.START == 0).
- Read the following PMC register fields to confirm the power-gating status of partition PCIEX4CB.
  - (PMC_IMPL_PART_PCIEX4CB_POWER_GATE_STATUS_0.SRAM_SLEEP_STS == OFF).
  - (PMC_IMPL_PART_PCIEX4CB_POWER_GATE_CONTROL_0.SRAM_RET_STS == OFF).
  - (PMC_IMPL_PART_PCIEX4CB_POWER_GATE_CONTROL_0.LOGIC_SLEEP_STS == OFF).
- Set the following PMC register fields to ungate power to PCIe power partition PCIEX4CC.
  - (PMC_IMPL_PART_PCIEX4CC_POWER_GATE_CONTROL_0.SRAM_SLEEP_EN = OFF).
  - (PMC_IMPL_PART_PCIEX4CC_POWER_GATE_CONTROL_0.SRAM_RET_EN = OFF).
  - (PMC_IMPL_PART_PCIEX4CC_POWER_GATE_CONTROL_0.LOGIC_SLEEP = OFF).
  - (PMC_IMPL_PART_PCIEX4CC_POWER_GATE_CONTROL_0.START = 1).
- Poll the following PMC register field to become '0' to ensure power rail is enabled for PCIEX4CC.
  - (PMC_IMPL_PART_PCIEX4CC_POWER_GATE_CONTROL_0.START == 0).
- Read the following PMC register fields to confirm the power-gating status of partition PCIEX4CC.
  - (PMC_IMPL_PART_PCIEX4CC_POWER_GATE_STATUS_0.SRAM_SLEEP_STS == OFF).
  - (PMC_IMPL_PART_PCIEX4CC_POWER_GATE_CONTROL_0.SRAM_RET_STS == OFF).
  - (PMC_IMPL_PART_PCIEX4CC_POWER_GATE_CONTROL_0.LOGIC_SLEEP_STS == OFF).

NOTE:

For details of the PMC registers, please refer to the Power management Controller (PMC) chapter.

The System Power Management driver enables clocks to all the PCIe power partitions.

- Set the following CAR (Clock and Reset) register field to '1' to enable the clocks to PCIe C0 in PCIEX4BA.
  - (CLK_RST_CONTROLLER_CLK_OUT_ENB_PEX0_CORE_0.CLK_ENB_PEX0_CORE_0 = 1).
- Set the following CAR (Clock and Reset) register fields to '1' to enable the clocks to PCIe C1, C2, and C3 in PCIEX1A.

  - (CLK_RST_CONTROLLER_CLK_OUT_ENB_PEX0_CORE_0.CLK_ENB_PEX0_CORE_1 = 1).
  - (CLK_RST_CONTROLLER_CLK_OUT_ENB_PEX0_CORE_0.CLK_ENB_PEX0_CORE_2 = 1)
  - (CLK_RST_CONTROLLER_CLK_OUT_ENB_PEX0_CORE_0.CLK_ENB_PEX0_CORE_3 = 1).
- Set the following CAR (Clock and Reset) register field to '1' to enable the clocks to PCIe C4 in PCIEX4BB.
  - (CLK_RST_CONTROLLER_CLK_OUT_ENB_PEX0_CORE_0.CLK_ENB_PEX0_CORE_4 = 1).
- Set the following CAR (Clock and Reset) register field to '1' to enable the clocks to PCIe C5 in PCIEX8A.
  - (CLK_RST_CONTROLLER_CLK_OUT_ENB_PEX1_CORE_0.CLK_ENB_PEX1_CORE_5 = 1).
- Set the following CAR (Clock and Reset) register field to '1' to enable the clocks to PCIe C6 in PCIEX4A.
  - (CLK_RST_CONTROLLER_CLK_OUT_ENB_PEX1_CORE_0.CLK_ENB_PEX1_CORE_6 = 1).
- Set the following CAR (Clock and Reset) register field to '1' to enable the clocks to PCIe C7 in PCIEX8B.
  - (CLK_RST_CONTROLLER_CLK_OUT_ENB_PEX2_CORE_0.CLK_ENB_PEX2_CORE_7 = 1).
- Set the following CAR (Clock and Reset) register field to '1' to enable the clocks to PCIe C8 in PCIEX4CA.
  - (CLK_RST_CONTROLLER_CLK_OUT_ENB_PEX2_CORE_0.CLK_ENB_PEX2_CORE_8 = 1).
- Set the following CAR (Clock and Reset) register field to '1' to enable the clocks to PCIe C9 in PCIEX4CB.
  - (CLK_RST_CONTROLLER_CLK_OUT_ENB_PEX2_CORE_0.CLK_ENB_PEX2_CORE_9 = 1).
- Set the following CAR (Clock and Reset) register field to '1' to enable the clocks to PCIe C10 in PCIEX4CC.
  - (CLK_RST_CONTROLLER_CLK_OUT_ENB_PEX2_CORE_0.CLK_ENB_PEX2_CORE_10 = 1).

NOTE:

For details of the CAR registers, please refer to the Clock and Reset Controller (CAR) chapter.

The System Power Management driver waits 2 µs.

The System Power Management driver removes clamps of all the PCIe power partitions.

- Set the following PMC register field to '0' to disable clamps of PCIEX4BA.
  - (PMC_IMPL_PART_PCIEX4BA_CLAMP_CONTROL_0.CLAMP = 0).
- Set the following PMC register field to '0' to disable clamps of PCIEX1A.
  - (PMC_IMPL_PART_PCIEX1A_CLAMP_CONTROL_0.CLAMP = 0).
- Set the following PMC register field to '0' to disable clamps of PCIEX4BB.
  - (PMC_IMPL_PART_PCIEX4BB_CLAMP_CONTROL_0.CLAMP = 0).

- Set the following PMC register field to '0' to disable clamps of PCIEX8A.
  - (PMC_IMPL_PART_PCIEX8A_CLAMP_CONTROL_0.CLAMP = 0).
- Set the following PMC register field to '0' to disable clamps of PCIEX4A.
  - (PMC_IMPL_PART_PCIEX4A_CLAMP_CONTROL_0.CLAMP = 0).
- Set the following PMC register field to '0' to disable clamps of PCIEX8B.
  - (PMC_IMPL_PART_PCIEX8B_CLAMP_CONTROL_0.CLAMP = 0).
- Set the following PMC register field to '0' to disable clamps of PCIEX4CA.
  - (PMC_IMPL_PART_PCIEX4CA_CLAMP_CONTROL_0.CLAMP = 0).
- Set the following PMC register field to '0' to disable clamps of PCIEX4CB.
  - (PMC_IMPL_PART_PCIEX4CB_CLAMP_CONTROL_0.CLAMP = 0).
- Set the following PMC register field to '0' to disable clamps of PCIEX4CC.
  - (PMC_IMPL_PART_PCIEX4CC_CLAMP_CONTROL_0.CLAMP = 0).

NOTE:

For details of the PMC registers, please refer to the Power management Controller (PMC) chapter.

The System Power Management driver waits 2 µs.

The System Power Management driver de-asserts Reset to APB Client of each PCIe controller.

- Set the following CAR (Clock and Reset) register field to '0' to de-assert Reset to the APB clients of PCIe C0 in PCIEX4BA.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_0_APB_RST = 0).
- Set the following CAR (Clock and Reset) register fields to '0' to de-assert Reset to the APB clients of PCIe C1, C2, and C3 in PCIEX1A.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_1_APB_RST = 0).
  - (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_2_APB_RST = 0).
  - (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_3_APB_RST = 0).
- Set the following CAR (Clock and Reset) register field to '0' to de-assert Reset to the APB clients of PCIe C4 in PCIEX4BB.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_4_APB_RST = 0).
- Set the following CAR (Clock and Reset) register field to '0' to de-assert Reset to the APB clients of PCIe C5 in PCIEX8A.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX1_CORE_0.SWR_PEX1_CORE_5_APB_RST = 0).
- Set the following CAR (Clock and Reset) register field to '0' to de-assert Reset to the APB clients of PCIe C6 in PCIEX4A.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX1_CORE_0.SWR_PEX1_CORE_6_APB_RST = 0).

- Set the following CAR (Clock and Reset) register field to '0' to de-assert Reset to the APB clients of PCIe C7 in PCIEX8B.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX2_CORE_0.SWR_PEX2_CORE_7_APB_RST = 0).
- Set the following CAR (Clock and Reset) register field to '0' to de-assert Reset to the APB clients of PCIe C8 in PCIEX4CA.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX2_CORE_0.SWR_PEX2_CORE_8_APB_RST = 0).
- Set the following CAR (Clock and Reset) register field to '0' to de-assert Reset to the APB clients of PCIe C9 in PCIEX4CB.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX2_CORE_0.SWR_PEX2_CORE_9_APB_RST = 0).
- Set the following CAR (Clock and Reset)register field to '0' to de-assert Reset to the APB clients of PCIe C10 in PCIEX4CC.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX2_CORE_0.SWR_PEX2_CORE_10_APB_RST = 0).

NOTE:

For details of the CAR registers, please refer to the Clock and Reset Controller (CAR) chapter.

The PCIe driver sets up PCIe controller configurations following the programming sequences in the PCIe Controller Setup section.

The System Power Management driver de-asserts Reset to PCIe controller.

- Set the following CAR (Clock and Reset) register field to '0' to de-assert Reset to PCIe C0 in PCIEX4BA.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_0_RST = 0).
- Set the following CAR (Clock and Reset) register fields to '0' to de-assert Reset to PCIe C1, C2, and C3 in PCIEX1A.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_1_RST = 0).
  - (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_2_RST = 0).
  - (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_3_RST = 0).
- Set the following CAR (Clock and Reset) register field to '0' to de-assert Reset to PCIe C4 in PCIEX4BB.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX0_CORE_0.SWR_PEX0_CORE_4_RST = 0).
- Set the following CAR (Clock and Reset) register field to '0' to de-assert Reset to PCIe C5 in PCIEX8A.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX1_CORE_0.SWR_PEX1_CORE_5_RST = 0).
- Set the following CAR (Clock and Reset) register field to '0' to de-assert Reset to PCIe C6 in PCIEX4A.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX1_CORE_0.SWR_PEX1_CORE_6_RST = 0).

- Set the following CAR (Clock and Reset) register field to ‘0’ to de-assert Reset PCIe C7 in PCIEX8B.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX2_CORE_0.SWR_PEX2_CORE_7_RST = 0).
- Set the following CAR (Clock and Reset) register field to ‘0’ to de-assert Reset to PCIe C8 in PCIEX4CA.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX2_CORE_0.SWR_PEX2_CORE_8_RST = 0).
- Set the following CAR (Clock and Reset) register field to ‘0’ to de-assert Reset to PCIe C9 in PCIEX4CB.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX2_CORE_0.SWR_PEX2_CORE_9_RST = 0).
- Set the following CAR(Clock and Reset) register field to ‘0’ to de-assert Reset to PCIe C10 in PCIEX4CC.
  - (CLK_RST_CONTROLLER_RST_DEV_PEX2_CORE_0.SWR_PEX2_CORE_10_RST = 0).

NOTE:

For details of the CAR registers, please refer to the Clock and Reset Controller (CAR) chapter.

The System Power Management driver waits 1 µs.

The System Power Management driver disables the flush of all the PCIe controllers in each PCIe power partition and their associated MCCIF (Memory Controler Connect InterFace).

- Set the following MC register fields to ‘0’ to disable flush of PCIe C0 in PCIEX4BA.
  - (MC_CLIENT_HOTRESET_CTRL_2_0.PCIE0A_FLUSH_ENABLE = 0).
  - (MC_CLIENT_HOTRESET_CTRL_2_0.PCIE0A2_FLUSH_ENABLE = 0).
- Poll the following fields for '0' to ensure flush disable completed for C0 in PCIEX4BA.
  - (MC_CLIENT_HOTRESET_STATUS_2_0.PCIE0A_HOTRESET_STATUS == 0).
  - (MC_CLIENT_HOTRESET_STATUS_2_0.PCIE0A2_HOTRESET_STATUS == 0).
- Set the following MC register fields to ‘0’ to disable flush of PCIe C1, C2, C3 in PCIEX1A.
  - (MC_CLIENT_HOTRESET_CTRL_2_0.PCIE_FLUSH_ENABLE = 0).
  - (MC_CLIENT_HOTRESET_CTRL_2_0.PCIE2A_FLUSH_ENABLE = 0).
  - (MC_CLIENT_HOTRESET_CTRL_2_0.PCIE3_FLUSH_ENABLE = 0).
  - (MC_CLIENT_HOTRESET_CTRL_2_0.PCIE3A_FLUSH_ENABLE = 0).
- Poll the following fields for '0' to ensure flush disable completed for PCIe C1, C2, C3 in PCIEX1A.
  - (MC_CLIENT_HOTRESET_STATUS_2_0.PCIE_HOTRESET_STATUS == 0).
  - (MC_CLIENT_HOTRESET_STATUS_2_0.PCIE2A_HOTRESET_STATUS == 0).
  - (MC_CLIENT_HOTRESET_STATUS_2_0.PCIE3_HOTRESET_STATUS == 0).
  - (MC_CLIENT_HOTRESET_STATUS_2_0.PCIE3A_HOTRESET_STATUS == 0).

- Set the following MC register field to ‘0’ to disable flush of PCIE C4 in PCIEX4BB.
  - (MC_CLIENT_HOTRESET_CTRL_2_0.PCIE4A_FLUSH_ENABLE = 0).
- Poll the following field for '0' to ensure flush disable completed for PCIe C4 in PCIEX4BB.
  - (MC_CLIENT_HOTRESET_STATUS_2_0.PCIE4A_HOTRESET_STATUS == 0).
- Set the following MC register field to ‘0’ to disable flush of PCIe C5 in PCIEX8A.
  - (MC_CLIENT_HOTRESET_CTRL_2_0.PCIE5A_FLUSH_ENABLE = 0).
- Poll the following field for '0' to ensure flush disable completed for PCIe C5 in PCIEX8A.
  - (MC_CLIENT_HOTRESET_STATUS_2_0.PCIE5A_HOTRESET_STATUS == 0).
- Set the following MC register field to ‘0’ to disable flush of PCIe C6 in PCIEX4A.
  - (MC_CLIENT_HOTRESET_CTRL_3_0.PCIE6A_FLUSH_ENABLE = 0).
- Poll the following field for '0' to ensure flush disable completed for PCIe C6 in PCIEX4A.
  - (MC_CLIENT_HOTRESET_STATUS_2_0.PCIE6A_HOTRESET_STATUS == 0).
- Set the following MC register field to ‘0’ to disable flush oc PCIe C7 in PCIEX8B.
  - (MC_CLIENT_HOTRESET_CTRL_3_0.PCIE7A_FLUSH_ENABLE = 0).
- Poll the following field for '0' to ensure flush disable completed for PCIe C7 in PCIEX8B.
  - (MC_CLIENT_HOTRESET_STATUS_2_0.PCIE7A_HOTRESET_STATUS == 0).
- Set the following MC register field to ‘0’ to disable flush of PCIe C8 in PCIEX4CA.
  - (MC_CLIENT_HOTRESET_CTRL_3_0.PCIE8A_FLUSH_ENABLE = 0).
- Poll the following field for '0' to ensure flush disable completed for PCIe C8 in PCIEX4CA.
  - (MC_CLIENT_HOTRESET_STATUS_2_0.PCIE8_HOTRESET_STATUS == 0).
- Set the following MC register field to ‘0’ to disable flush of PCIe C9 in PCIEX4CB.
  - (MC_CLIENT_HOTRESET_CTRL_3_0.PCIE9A_FLUSH_ENABLE = 0).
- Poll the following field for '0' to ensure flush disable completed for PCIe C9 in PCIEX4CB.
  - (MC_CLIENT_HOTRESET_STATUS_2_0.PCIE9A_HOTRESET_STATUS == 0).
- Set the following MC register field to ‘0’ to disable flush of PCIe C10 in PCIEX4CC.
  - (MC_CLIENT_HOTRESET_CTRL_3_0.PCIE10A_FLUSH_ENABLE = 0).
- Poll the following field for '0' to ensure flush disable completed for PCIe C10 in PCIEX4CC.
  - (MC_CLIENT_HOTRESET_STATUS_2_0.PCIE10A_HOTRESET_STATUS == 0).

NOTE:

For details of the MC registers, please refer to the Memory SubSystem (MSS) chapter.

9.3.4.7 SC7

All the PCIe power partition must be put to partition power-gating before the system can enter SC7. The PCIe partition can either be brought out of power-gating once the system exits SC7 or kept under power-gating until the connected devices signal a wake event.

Assertion of PE_WAKE# must be set as an SC7 wake event for all PCIe Root Ports controllers.

The PCIe driver enables PE_WAKE# as a wake event before the system enters SC7.

- Set the LEVEL filed of the particular PMC WAKE register corresponding to the WAKE event to '0' to set the wake signal active level to 'LOW'.
  - (WAKE_AOWAKE_CNTRL_1**.**LEVEL = 0).
- Set the MASK field of the particular PMC WAKE register corresponding to the WAKE event to '1' to enable PCIe wake events.
  - (WAKE_AOWAKE_MASK_W_1**.**MASK = 1).

The PCIe driver disables PE_WAKE# as a wake event after the system exits SC7.

- Set the MASK field of the particular PMC WAKE register corresponding to the WAKE event to '0' to disable PCIe wake events.
  - (WAKE_AOWAKE_MASK_W_1**.**MASK = 0).

Assertion of RESET# must be set as an SC7 wake event for the PCIe Endpoint controller connected to a PCIe Root Port.

ThePCIe driver enables the Endpoint Reset GPIO as a wake event before the system enters SC7.

- Set the LEVEL field of the software-identified platform-specific PMC wake register to '0' to set the wake signal active level to 'LOW'.
  - WAKE_AOWAKE_CNTRL_<*i*>**.**LEVEL = 0), where <*i*> = 0, 1, ..., 95.
- Set the MASK field of the software-identified platform-specific PMC wake register to '1' to enable PCIe Endpoint Reset events.
  - WAKE_AOWAKE_MASK_W_<*i*>**.**MASK = 1), where <*i*> = 0, 1, ..., 95.

Th PCIe driver disables the Endpoint Reset GPIO as a wake event after the system exits SC7.

- Set the MASK field of the software-identified platform-specific PMC wake register to '0' to disable PCIe Endpoint Reset events.
  - WAKE_AOWAKE_MASK_W_<*i*>**.**MASK = 0), where <*i*> = 0, 1, ..., 95.

NOTE:

- Each GPIO pin capable of generating wake event is assigned with a fixed number as a wake source. Some GPIO pins have fixed assignment such as PE_WAKE# for PCIes. Meanwhile, some other GPIO pins without pre-defined functions are reserved to allow platform implementations as needed. RESET# takes advantage of the platform implementation

flexibility, since RESET# as input is not required by all platform implementations but only those with PCIe Endpoint controller enabled.

- PE_WAKE# is used by all PCIe Root Ports for connected devices to signal wake when the system is in SC7. This is a fixed function that is supported by all platforms.
- RESET# as a wake source is only used by platforms with enabled PCIe Endpoint. This is optional and different from PE_WAKE#, since in additional to signaling wake from the connected external Root Port, it's also used by that Root Port to reset the Endpoint.
- Actual GPIO pin/PMC register for wake is based on platform GPIO designation, hence, platform-specific PMC wake register.
- For details of the WAKE_AOWAKE_* registers, please refer to the "PMC Wake Registers" in the PMC chapter.

9.3.4.8 Fault Handling

For PCIe controller in Root Port mode that reports fault occurrence, software needs to read the error statuses in APB Client to get information of the type of error. To recover from the fault, software needs to reset the PCIe controller through CAR register programming, followed by reinitializing the PCIe fabric.

For PCIe controller in Endpoint mode that reports fault occurrence, software needs to read the error statuses in APB Client to get information of the type of error. To recover from the fault, software needs to reset the PCIe controller through CAR register programming, which causes the link to be disconnected and reported as Surprise Down error by the link partner.

9.3.4.9 Shutdown and Reboot

Before system shutdown or reboot, the PCIe driver needs to put all PCIe links to Disabled state, followed by asserting RESET# of all Root Ports then reset UPHY, to ensure PCIe devices and UPHY are in proper states before power rails are toggled. To avoid power surge when numerous logic are powering down at the same time, it is recommended that thePCIe driver also puts all the PCIe partitions into power-gating prior to shutdown or reboot.

9.3.5 PCIe Registers

9.3.5.1 PCIe iATU

NOTE:

There are 11 instances for each of the registers as listed, one for each of the PCIe module instances, namely C0, C1, C2, C3, C4, C5, C6, C7, C8, C9, and C10.

For the base addresses of these different register instances, please refer to the System Address Map.

NOTE:

The binary “x” is turned to “0” for the Reset and PROD hex value. Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as “x” or “Undefined” actually means that the bit could be either “0” or “1” after Reset, whereas a bit PROD value as “x” or “Don’t-care” means that either “0” or “1” can be written to the bit by software Initialization.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_REGION_CTRL_1_OFF_OUTBOUND_0_0

Description: This register controls the iATU outbound region access based on the optional iATU outbound features enabled using iATU Region Control 2 Register.

Offset: 0x0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,x000,xxxx,xx0x,x000,0000,0000)

Bit	Reset	Description
22:20	0x0	**CTRL_1_FUNC_NUM**: Function Number. - When the address of an outbound TLP is matched to this region and the FUNC_BYPASS field in the "iATU Region Control 2 Register" is '0', then the function number used in generating the function part of the requester ID (RID) field of the TLP is taken from this 5-bit register. The value in this register must be 0x0 unless multifunction operation in the controller is enabled (CX_NFUNC > 1). - When you are using the AXI Bridge, then this field is swapped before AXI decomposition occurs so that the correct "Max_Read_Request_Size" and "Max_Payload_Size" values are used. Note: This register field is sticky.
13	0x0	**INCREASE_REGION_SIZE**: Increase the maximum ATU Region size. When set, the maximum ATU Region size is determined by CX_ATU_MAX_REGION_SIZE. When clear, the maximum ATU Region size is 4 GB (default). Note: This register field is sticky.
10:9	0x0	**ATTR**: When the address of an outbound TLP is matched to this region, then the ATTR field of the TLP is changed to the value in this register. Note: This register field is sticky.
8	0x0	**TD**: This is a reserved field. Do not use. Note: This register field is sticky.
7:5	0x0	**TC**: When the address of an outbound TLP is matched to this region, then the TC field of the TLP is changed to the value in this register. Note: This register field is sticky.

Bit	Reset	Description
4:0	0x0	**TYPE**: When the address of an outbound TLP is matched to this region, then the TYPE field of the TLP is changed to the value in this register. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_REGION_CTRL_2_OFF_OUTBOUND_0_0

Description: Using this register you can enable/disable the outbound iATU optional features.

Offset: 0x4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0x00,0xxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**REGION_EN**: Region Enable. This bit must be set to '1' for address translation to take place. Note: This register field is sticky.
29	0x0	**INVERT_MODE**: Invert Mode. When set the address matching region is inverted. Therefore, an address match occurs when the untranslated address is in the region outside the defined range (Base Address to Limit Address). Note: This register field is sticky.
28	0x0	**CFG_SHIFT_MODE**: CFG Shift Mode. The iATU uses bits [27:12] of the untranslated address (on the XALI0/1/2 interface or AXI slave interface address) to form the BDF number of the outgoing CFG TLP. This supports the Enhanced Configuration Address Mapping (ECAM) mechanism (Section 7.2.2 of the PCI Express Base 3.1 Specification, revision 1.0) by allowing all outgoing I/O and MEM TLPs (that have been translated to CFG) to be mapped from memory space into any 256 MB region of the PCIe configuration space. Note: This register field is sticky.
27	0x0	**DMA_BYPASS**: DMA Bypass Mode. Allows request TLPs which are initiated by the DMA controller to pass through the iATU untranslated. Note: This field is reserved for the SW product. You must set it to '0'. Note: This register field is sticky.
23	0x0	**HEADER_SUBSTITUTE_EN**: Header Substitute Enable. When enabled and region address is matched, the iATU fully substitutes bytes 8-11 (for 3 DWORD header) or bytes 12-15 (for 4 DWORD header) of the outbound TLP header with the contents of the LWR_TARGET_RW field in IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i. Encodings are as above. Note: This register field is sticky.
22	0x0	**INHIBIT_PAYLOAD**: Inhibit TLP Payload Data for TLP's in Matched Region; assign iATU region to be TLP without data. When enabled and region address is matched, the iATU marks all TLPs as having no payload data by forcing the TLP header Fmt[1] bit =0, regardless of the application inputs such as slv_wstrb. Encoding are define as above. Note: This register field is sticky.

Bit	Reset	Description
21	0x0	**TLP_HEADER_FIELDS_BYPASS**: TLP Header Fields Translation Bypass. In this mode header fields of the translated TLP is taken from your application transmit interface or, if AMBA is configured, from the AMBA sideband bus (slv_awmisc_info) and not from the corresponding fields of the IATU_REGION_CTRL_1_OFF_OUTBOUND_i or IATU_REGION_CTRL_2_OFF_OUTBOUND_i registers. The header fields are - TC - PH - TH - ST - AT - Attr (IDO, RO and NS). Note: This register field is sticky.
20	0x0	**SNP**: Serialize Non-Posted Requests. In this mode, when the AXI Bridge is populated, same AXI ID Non-Posted Read/Write Requests are transmitted on the wire if there are no other same ID Non-Posted Requests outstanding. Note: This register field is sticky.
19	0x0	**FUNC_BYPASS**: Function Number Translation Bypass. In this mode, the function number of the translated TLP is taken from your application transmit interface and not from the CTRL_1_FUNC_NUM field of the "iATU Region Control 1 Register" or the VF_NUMBER field of the "iATU Control 3 Register." Note: This register field is sticky.
18:17	0x0	**MSB2BITS_TAG**: TAG. Two most significant bits of the substituted TAG field in the outgoing TLP header when TAG_SUBSTITUTE_EN is set. Note: This register field is sticky.
16	0x0	**TAG_SUBSTITUTE_EN**: TAG Substitute Enable. When enabled and region address is matched, the iATU substitutes the TAG field of the outbound TLP header with the contents of the TAG field in this register. The expected usage scenario is translation from AXI MWr to Vendor Defined Msg/MsgD. Note (CX_TPH_ENABLE=1): TAG substitution for MWr will not occur because this field (byte 6) in the TLP header is the ST field. ST substitution can still take place using the MSG_CODE field in IATU_REGION_CTRL_2_OFF_OUTBOUND_i. Note: This register field is sticky.
15:8	0x0	**TAG**: TAG. The substituted TAG field (byte 6) in the outgoing TLP header when TAG_SUBSTITUTE_EN is set. Note: This register field is sticky.
7:0	0x0	**MSG_CODE**: MSG TLPs (Message Code). When the address of an outbound TLP is matched to this region, and the translated TLP TYPE field is Msg or MsgD; then the message field of the TLP is changed to the value in this register. Memory TLPs: (ST: Steering Tag). When the ST field of an outbound TLP is matched to this region, and the translated TLP TYPE field targets memory space; then the ST field of the TLP is changed to the value in this register. Only Valid when the CX_TPH_ENABLE configuration parameter is 1. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LWR_BASE_ADDR_OFF_OUTBOUND_0_0

Description: The CX_ATU_MIN_REGION_SIZE configuration parameter (Value Range: 4 KiB, 8 KiB, 16 KiB, 32 KiB, 64 KiB defaults to 64 KiB) specifies the minimum size of an address translation region. For example, if set to 64 KiB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned on 64 KiB boundaries. More precisely, the lower log2(CX_ATU_MIN_REGION_SIZE) bits are zero.

Offset: 0x8
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:16	RW	0x0	**LWR_BASE_RW**: Forms bits [31:n] of the start address of the address region to be translated. n is log2(CX_ATU_MIN_REGION_SIZE) Note: This register field is sticky.
15:0	RO	0x0	**LWR_BASE_HW**: Forms bits [n-1:0] of the start address of the address region to be translated. The start address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. n is log2(CX_ATU_MIN_REGION_SIZE)

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_0_0

Description: This register holds the upper 32-bits of the start (and end) address of the address region to be translated.

Offset: 0xc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UPPER_BASE_RW**: Forms bits [63:32] of the start (and end) address of the address region to be translated. In systems with a 32-bit address space, this register is not used and therefore writing to this register has no effect. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LIMIT_ADDR_OFF_OUTBOUND_0_0

Description: This register holds the end address of the address region to be translated.

Offset: 0x10
Read/Write: See table below
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	R/W	Reset	Description
31:16	RW	0x0	**LIMIT_ADDR_RW**: Forms upper bits of the end address of the address region to be translated. Note: This register field is sticky.
15:4	RO	0xfff	**LIMIT_ADDR_HW**: Forms lower bits of the end address of the address region to be translated. The end address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so these bits are always all ones. A write to this location is ignored by the PCIe controller. Note: This register field is sticky.
3:0	RO	0xf	**CBUF_INCR**: Circular Buffer. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_0_0

Description: This register holds the Lower Target part of the new address of the translated region, or the outbound TLP header information, depending on the setting of HEADER_SUBSTITUTE_EN field of IATU_REGION_CTRL_2_VIEWPORT_OFF_OUTBOUND_i register.

Offset: 0x14
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LWR_TARGET_RW_OUTBOUND**: When HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_ is '0' (normal operation): - LWR_TARGET_RW[31:n] forms MSB's of the Lower Target part of the new address of the translated region; - LWR_TARGET_RW[n-1:0] are not used. (The start address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so the lower bits of the start address of the new address of the translated region (bits n-1:0) are always '0'). - n is log2(CX_ATU_MIN_REGION_SIZE). When HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_i is '1': - LWR_TARGET_RW[31:0] forms bytes 8-11 (for 3 dword header) or bytes 12-15 (for 4 dword header) of the outbound TLP header. Usage scenarios include the transmission of Vendor Defined Messages where the controller determines the content of bytes 12 to 15 of the TLP header. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_0_0

Description: This register holds the upper 32 bits of the start address (Upper Target part) of the new address of the translated region.

Offset: 0x18
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UPPER_TARGET_RW**: Forms bits [63:32] of the start address (Upper Target part) of the new address of the translated region. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPR_LIMIT_ADDR_OFF_OUTBOUND_0_0

Description: The CX_MAX_ATU_REGION_SIZE configuration parameter (Value Range: 0->32, 8=1TB) specifies the maximum region size of an address translation region. This register is only used when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_OUTBOUND_i is '1'.

Offset: 0x20
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:3	RO	0x0	**UPPR_LIMIT_ADDR_HW**: Forms MSB's of the Upper Limit part of the region "end address" to be translated. Only applies to 64-bit systems and when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_OUTBOUND_i is '1'. These bits are always '0'.
2:0	RW	0x0	**UPPR_LIMIT_ADDR_RW**: Forms the LSB's of the Upper Limit part of the region "end address" to be translated. Only applies to 64-bit systems and when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_OUTBOUND_i is '1' Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_REGION_CTRL_1_OFF_INBOUND_0_0

Description: This register controls the iATU inbound region access based on the optional iATU inbound features enabled using iATU Region Control 2 Register.

Offset: 0x100
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,x000,xxxx,xx0x,x000,0000,0000)

Bit	Reset	Description
22:20	0x0	**CTRL_1_FUNC_NUM**: Function Number. - MEM-I/O: When the Address and BAR matching logic in the controller indicate that a MEM-I/O transaction matches a BAR in the function corresponding to this value, then address translation proceeds. This check is only performed if the "Function Number Match Enable" bit of the "iATU Region Control 2 Register" is set. - CFG0/CFG1: When the destination function number as specified in the routing ID of the TLP header matches the function, then address translation proceeds. This check is only performed if the "Function Number Match Enable" bit of the "iATU Region Control 2 Register" is set. Note: This register field is sticky.
13	0x0	**INCREASE_REGION_SIZE**: Increase the maximum ATU Region size. When set, the maximum ATU Region size is determined by CX_ATU_MAX_REGION_SIZE. When clear, the maximum ATU Region size is 4 GB (default). Note: This register field is sticky.
10:9	0x0	**ATTR**: When the ATTR field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "ATTR Match Enable" bit of the "iATU Region Control 2 Register" is set. Note: This register field is sticky.
8	0x0	**TD**: When the TD field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "TD Match Enable" bit of the "iATU Region Control 2 Register" is set. Note: This register field is sticky.
7:5	0x0	**TC**: When the TC field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "TC Match Enable" bit of the "iATU Region Control 2 Register" is set. Note: This register field is sticky.
4:0	0x0	**TYPE**: When the TYPE field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_REGION_CTRL_2_OFF_INBOUND_0_0

Description: Using this register you can enable/disable the inbound iATU optional features.

Offset: 0x104
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0x00,0x0x,0xx0,000x,x000,0000,0000)

Bit	Reset	Description
31	0x0	**REGION_EN**: Region Enable. This bit must be set to '1' for address translation to take place. Note: This register field is sticky.
30	0x0	**MATCH_MODE**: Match Mode. Determines Inbound matching mode for TLPs. The mode depends on the type of TLP that is received as follows: For MEM-I/O TLPs, this field is interpreted as follows: -0: Address Match Mode. The iATU operates using addresses as in the outbound direction. The Region Base and Limit Registers must be setup. -1: BAR Match Mode. BAR matching is used. The "BAR Number" field is relevant. Not used for RC. For CFG0 TLPs, this field is interpreted as follows: -0: Routing ID Match Mode. The iATU interprets the Routing ID (Bytes 8 to 11 of TLP header) as an address. This corresponds to the upper 16 bits of the address in MEM-I/O transactions. The Routing ID of the TLP must be within the base and limit of the iATU region for matching to proceed. -1: Accept Mode. The iATU accepts all CFG0 transactions as address matches. The routing ID in the CFG0 TLP is ignored. This is useful as all received CFG0 TLPs should be processed regardless of the Bus number. For MSG/MSGD TLPs, this field is interpreted as follows: -0: Address Match Mode. The iATU treats the third dword and fourth dword of the inbound MSG/MSGD TLP as an address and it is matched against the Region Base and Limit Registers. -1: Vendor ID Match Mode. This mode is relevant for ID-routed Vendor Defined Messages. The iATU ignores the Routing ID (Bus, Device, Function) in bits [31:16] of the third dword of the TLP header, but matches against the Vendor ID in bits [15:0] of the third dword of the TLP header. Bits [15:0] of the Region Upper Base register should be programmed with the required Vendor ID. The lower Base and Limit Register should be programmed to translate TLPs based on vendor specific information in the fourth dword of the TLP header. - If SINGLE_ADDRESS_LOCATION_TRANSLATE_EN = 1 AND MSG_TYPE_MATCH_MODE =1, then Match Mode is ignored. Note: This register field is sticky.
29	0x0	**INVERT_MODE**: Invert Mode Enable. When set the address matching region is inverted. Therefore, an address match occurs when the untranslated address is in the region outside the defined range (Base Address to Limit Address). When set all regions of that type must use address match mode. Note: This register field is sticky.
28	0x0	**CFG_SHIFT_MODE**: CFG Shift Enable. This is useful for CFG transactions where the PCIe configuration mechanism maps bits [27:12] of the address to the bus/device and function number. This allows a CFG configuration space to be located in any 256MB window of your application memory space using a 28-bit effective address. Shifts bits [31:16] of the untranslated address to form bits [27:12] of the translated address. Note: This register field is sticky.
27	0x0	**FUZZY_TYPE_MATCH_CODE**: Fuzzy Type Match Enable. When enabled, the iATU relaxes the matching of the TLP TYPE field against the expected TYPE field so that - CfgRd0 and CfgRd1 TLPs are seen as identical. Similarly with CfgWr0 and CfgWr1. - MWr, MRd, and MRdLk TLPs are seen as identical - The Routing field of Msg/MsgD TLPs is ignored - FetchAdd, Swap, and CAS are seen as identical. For example, CFG0 in the TYPE field in the "iATU Region Control 1 Register" matches against an inbound CfgRd0, CfgRd1, CfgWr0, or CfgWr1 TLP. Note: This register field is sticky.

Bit	Reset	Description
25:24	0x0	**RESPONSE_CODE**: Response Code. Defines the type of response to give for accesses matching this region. This overrides the normal RADM filter response. Note that this feature is not available for any region where Single Address Location Translate is enabled. Note: This register field is sticky.
23	0x0	**SINGLE_ADDR_LOC_TRANS_EN**: Single Address Location Translate Enable. When enabled, Rx TLPs can be translated to a single address location as determined by the target address register of the iATU region. The main usage scenario is translation of Messages (such as Vendor Defined or ATS Messages) to MWr TLPs when the AXI bridge is enabled. Note: This register field is sticky.
21	0x0	**MSG_CODE_MATCH_EN**: Message Code Match Enable (Msg TLPS). Ensures that a successful message Code TLP field comparison match (see Message Code field of the "iATU Region Control 2 Register") occurs (in MSG transactions) for address translation to proceed. ST Match Enable (Mem TLPs). Ensures that a successful ST TLP field comparison match (see ST field of the "iATU Region Control 2 Register") occurs (in MEM transactions) for address translation to proceed. Only Valid when the CX_TPH_ENABLE configuration parameter is '1'. Note: This register field is sticky.
19	0x0	**FUNC_NUM_MATCH_EN**: Function Number Match Enable. Ensures that a successful Function Number TLP field comparison match (see Function Number field of the "iATU Region Control 1 Register") occurs (in MEM-I/O and CFG0/CFG1 transactions) for address translation to proceed. Note: This register field is sticky.
16	0x0	**ATTR_MATCH_EN**: ATTR Match Enable. Ensures that a successful ATTR TLP field comparison match (see ATTR field of the "iATU Region Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.
15	0x0	**TD_MATCH_EN**: TD Match Enable. Ensures that a successful TD TLP field comparison match (see TD field of the "iATU Region Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.
14	0x0	**TC_MATCH_EN**: TC Match Enable. Ensures that a successful TC TLP field comparison match (see TC field of the "iATU Region Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.
13	0x0	**MSG_TYPE_MATCH_MODE**: Message Type Match Mode. When enabled, and if single address location translate enable is set, then inbound TLPs of type MSG/MSGd which match the type field of the IATU_REGION_CTRL_1_VIEWPORT_OFF_INBOUND_i register (TYPE[4:3]=2'b10) will be translated. Message type match mode overrides any value of MATCH_MODE field in this register. Usage scenarios for this are translation of VDM or ATS messages when AXI bridge is configured on client interface. Note: This register field is sticky.
10:8	0x0	**BAR_NUM**: BAR Number. When the BAR number of an inbound MEM or IO TLP " that is matched by the normal internal BAR address matching mechanism " is the same as this field, address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "Match Mode" bit of the "iATU Region Control 2 Register" is set. IO translation would require either 00100b or 00101b in the inbound TLP TYPE; the BAR Number set in the range 000b - 101b and that BAR configured as an IO BAR. Note: This register field is sticky.

Bit	Reset	Description
7:0	0x0	**MSG_CODE**: MSG TLPs: (Message Code). When the TYPE field of an inbound Msg/MsgD TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "Message Code Match Enable" bit of the "iATU Region Control 2 Register" is set. Memory TLPs: (ST: Steering Tag). When the ST field of an inbound TLP is matched to this value, then address translation proceeds. This check is only performed if the "ST Match Enable" bit of the "iATU Region Control 2 Register" is set. The setting is independent of the setting of the TH field. Only Valid when the CX_TPH_ENABLE configuration parameter is '1'. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LWR_BASE_ADDR_OFF_INBOUND_0_0

Description: The CX_ATU_MIN_REGION_SIZE configuration parameter (Value Range: 4 KiB, 8 KiB, 16 KiB, 32 KiB, 64 KiB defaults to 64 KiB) specifies the minimum size of an address translation region. For example, if set to 64 KiB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned on 64 KiB boundaries. More precisely, the lower log2(CX_ATU_MIN_REGION_SIZE) bits are zero.

Offset: 0x108
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:16	RW	0x0	**LWR_BASE_RW**: Forms bits [31:n] of the start address of the address region to be translated. n is log2(CX_ATU_MIN_REGION_SIZE) Note: This register field is sticky.
15:0	RO	0x0	**LWR_BASE_HW**: Forms bits [n-1:0] of the start address of the address region to be translated. The start address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. n is log2(CX_ATU_MIN_REGION_SIZE)

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPER_BASE_ADDR_OFF_INBOUND_0_0

Description: This register holds the upper 32 bits of the start (and end) address of the address region to be translated.

Offset: 0x10c
Read/Write: R/W

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UPPER_BASE_RW**: Forms bits [63:32] of the start (and end) address of the address region to be translated. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LIMIT_ADDR_OFF_INBOUND_0_0

Description: This register holds the end address of the address region to be translated.

Offset: 0x110
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000fff0 (0b0000,0000,0000,0000,1111,1111,1111,0000)

Bit	R/W	Reset	Description
31:16	RW	0x0	**LIMIT_ADDR_RW**: Forms upper bits of the end address of the address region to be translated. When SINGLE_ADDR_LOC_TRANS_EN = 1, MSG_TYPE_MATCH_MODE = 1, and CBUF_INCR > 0 then this field forms the upper bits of the limit address for the circular buffer. Note: This register field is sticky.
15:4	RO	0xfff	**LIMIT_ADDR_HW**: Forms lower bits of the end address of the address region to be translated. The end address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so these bits are always all ones. When SINGLE_ADDR_LOC_TRANS_EN = 1, MSG_TYPE_MATCH_MODE = 1, and CBUF_INCR > 0 then this field forms lower bits of the limit address for the circular buffer. A write to this location is ignored by the PCIe controller. Note: This register field is sticky.

Bit	R/W	Reset	Description
3:0	RW	0x0	**CBUF_INCR**: Circular Buffer Increment. When CX_ATU_SLOC_CBUF = 0, then this field is Read-only and forms the lowest bits of the end address of the address region to be translated. When CX_ATU_SLOC_CBUF = 1, then this field is R/W and forms the upper bits of the Circular Buffer Increment size (CBUF_INCR) field for Single Location Address translation. The increment value (in bytes) is decoded as follows: - 0000b: 0 (Default; legacy Single Address Location mode) - 0001b: 4 - 0010b: 8 - 0011b: 16 - 0100b: 32 - 0101b: 64 - 0110b: 128 - 0111b: 256 - 1000b: 512 - 1001b: 1024 - 1010b: 2048 - 1011b: 4096 - 1100b: 8192 - 1101b: rsvd. - 1110b: rsvd. - 1111b: rsvd. Note: A write to any bit in the CBUF_INCR field resets the circular buffer pointer - that is, the next matched received Message will be translated to the start address of the Circular Buffer. This field must be written to AFTER the target and limit registers have been updated. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R (sticky) Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LWR_TARGET_ADDR_OFF_INBOUND_0_0

Description: This register holds the Lower Target part of the new address of the translated region.

Offset: 0x114
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:16	RW	0x0	**LWR_TARGET_RW**: Forms MSB's of the Lower Target part of the new address of the translated region. These bits are always '0'. - Field size depends on log2(CX_ATU_MIN_REGION_SIZE) in address match mode. - Field size depends on log2(BAR_MASK+1) in BAR match mode. Note: This register field is sticky.
15:0	RO	0x0	**LWR_TARGET_HW**: Forms the LSB's of the Lower Target part of the new address of the translated region. The start address must be aligned to a CX_ATU_MIN_REGION_SIZE KiB boundary (in address match mode); and to the Bar size boundary (in BAR match mode) so that these bits are always '0'. If the BAR is smaller than the iATU region size, then the iATU target address must align to the iATU region size; otherwise it must align to the BAR size. A write to this location is ignored by the PCIe controller. - Field size depends on log2(CX_ATU_MIN_REGION_SIZE) in address match mode. - Field size depends on log2(BAR_MASK+1) in BAR match mode.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPER_TARGET_ADDR_OFF_INBOUND_0_0

Description: This register holds the upper 32 bits of the start address (Upper Target part) of the new address of the translated region.

Offset: 0x118
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UPPER_TARGET_RW**: Forms bits [63:32] of the start address (Upper Target part) of the new address of the translated region. In systems with a 32-bit address space, this register is not used and therefore writing to this register has no effect. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPR_LIMIT_ADDR_OFF_INBOUND_0_0

Description: The CX_MAX_ATU_REGION_SIZE configuration parameter (Value Range: 0(4 GiB) to 32(16 GiB)) specifies the maximum region size of an address translation region. This register is only used when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_INBOUND_i is '1'.

Offset: 0x120
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:3	RO	0x0	**UPPR_LIMIT_ADDR_HW**: Forms MSB's of the Upper Limit part of the region "end address" to be translated. Only applies to 64-bit systems and when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_INBOUND_i is '1'. These bits are always '0'.
2:0	RW	0x0	**UPPR_LIMIT_ADDR_RW**: Forms the LSB's of the Upper Limit part of the region "end address" to be translated. Only applies to 64-bit systems and when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_INBOUND_i is '1' Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_REGION_CTRL_1_OFF_OUTBOUND_1_0

Description: This register controls the iATU outbound region access based on the optional iATU outbound features enabled using iATU Region Control 2 Register.

Offset: 0x200
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,x000,xxxx,xx0x,x000,0000,0000)

Bit	Reset	Description
22:20	0x0	**CTRL_1_FUNC_NUM**: Function Number. - When the address of an outbound TLP is matched to this region and the FUNC_BYPASS field in the "iATU Region Control 2 Register" is '0', then the function number used in generating the function part of the requester ID (RID) field of the TLP is taken from this 5-bit register. The value in this register must be 0x0 unless multifunction operation in the controller is enabled (CX_NFUNC > 1). - When you are using the AXI Bridge, then this field is swapped before AXI decomposition occurs so that the correct "Max_Read_Request_Size" and "Max_Payload_Size" values are used. Note: This register field is sticky.
13	0x0	**INCREASE_REGION_SIZE**: Increase the maximum ATU Region size. When set, the maximum ATU Region size is determined by CX_ATU_MAX_REGION_SIZE. When clear, the maximum ATU Region size is 4 GB (default). Note: This register field is sticky.
10:9	0x0	**ATTR**: When the address of an outbound TLP is matched to this region, then the ATTR field of the TLP is changed to the value in this register. Note: This register field is sticky.
8	0x0	**TD**: This is a reserved field. Do not use. Note: This register field is sticky.
7:5	0x0	**TC**: When the address of an outbound TLP is matched to this region, then the TC field of the TLP is changed to the value in this register. Note: This register field is sticky.
4:0	0x0	**TYPE**: When the address of an outbound TLP is matched to this region, then the TYPE field of the TLP is changed to the value in this register. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_REGION_CTRL_2_OFF_OUTBOUND_1_0

Description: Using this register you can enable/disable the outbound iATU optional features.

Offset: 0x204
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0x00,0xxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**REGION_EN**: Region Enable. This bit must be set to '1' for address translation to take place. Note: This register field is sticky.
29	0x0	**INVERT_MODE**: Invert Mode. When set the address matching region is inverted. Therefore, an address match occurs when the untranslated address is in the region outside the defined range (Base Address to Limit Address). Note: This register field is sticky.
28	0x0	**CFG_SHIFT_MODE**: CFG Shift Mode. The iATU uses bits [27:12] of the untranslated address (on the XALI0/1/2 interface or AXI slave interface address) to form the BDF number of the outgoing CFG TLP. This supports the Enhanced Configuration Address Mapping (ECAM) mechanism (Section 7.2.2 of the PCI Express Base 3.1 Specification, revision 1.0) by allowing all outgoing I/O and MEM TLPs (that have been translated to CFG) to be mapped from memory space into any 256 MB region of the PCIe configuration space. Note: This register field is sticky.
27	0x0	**DMA_BYPASS**: DMA Bypass Mode. Allows request TLPs which are initiated by the DMA controller to pass through the iATU untranslated. Note: This field is reserved for the SW product. You must set it to '0'. Note: This register field is sticky.
23	0x0	**HEADER_SUBSTITUTE_EN**: Header Substitute Enable. When enabled and region address is matched, the iATU fully substitutes bytes 8-11 (for 3 DWORD header) or bytes 12-15 (for 4 DWORD header) of the outbound TLP header with the contents of the LWR_TARGET_RW field in IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i. Encodings are as above. Note: This register field is sticky.
22	0x0	**INHIBIT_PAYLOAD**: Inhibit TLP Payload Data for TLP's in Matched Region; assign iATU region to be TLP without data. When enabled and region address is matched, the iATU marks all TLPs as having no payload data by forcing the TLP header Fmt[1] bit =0, regardless of the application inputs such as slv_wstrb. Encoding are define as above. Note: This register field is sticky.
21	0x0	**TLP_HEADER_FIELDS_BYPASS**: TLP Header Fields Translation Bypass. In this mode header fields of the translated TLP is taken from your application transmit interface or, if AMBA is configured, from the AMBA sideband bus (slv_awmisc_info) and not from the corresponding fields of the IATU_REGION_CTRL_1_OFF_OUTBOUND_i or IATU_REGION_CTRL_2_OFF_OUTBOUND_i registers. The header fields are - TC - PH - TH - ST - AT - Attr (IDO, RO and NS). Note: This register field is sticky.
20	0x0	**SNP**: Serialize Non-Posted Requests. In this mode, when the AXI Bridge is populated, same AXI ID Non-Posted Read/Write Requests are transmitted on the wire if there are no other same ID Non-Posted Requests outstanding. Note: This register field is sticky.
19	0x0	**FUNC_BYPASS**: Function Number Translation Bypass. In this mode, the function number of the translated TLP is taken from your application transmit interface and not from the CTRL_1_FUNC_NUM field of the "iATU Region Control 1 Register" or the VF_NUMBER field of the "iATU Control 3 Register." Note: This register field is sticky.

Bit	Reset	Description
18:17	0x0	**MSB2BITS_TAG**: TAG. Two most significant bits of the substituted TAG field in the outgoing TLP header when TAG_SUBSTITUTE_EN is set. Note: This register field is sticky.
16	0x0	**TAG_SUBSTITUTE_EN**: TAG Substitute Enable. When enabled and region address is matched, the iATU substitutes the TAG field of the outbound TLP header with the contents of the TAG field in this register. The expected usage scenario is translation from AXI MWr to Vendor Defined Msg/MsgD. Note (CX_TPH_ENABLE=1): TAG substitution for MWr will not occur because this field (byte 6) in the TLP header is the ST field. ST substitution can still take place using the MSG_CODE field in IATU_REGION_CTRL_2_OFF_OUTBOUND_i. Note: This register field is sticky.
15:8	0x0	**TAG**: TAG. The substituted TAG field (byte 6) in the outgoing TLP header when TAG_SUBSTITUTE_EN is set. Note: This register field is sticky.
7:0	0x0	**MSG_CODE**: MSG TLPs (Message Code). When the address of an outbound TLP is matched to this region, and the translated TLP TYPE field is Msg or MsgD; then the message field of the TLP is changed to the value in this register. Memory TLPs: (ST: Steering Tag). When the ST field of an outbound TLP is matched to this region, and the translated TLP TYPE field targets memory space; then the ST field of the TLP is changed to the value in this register. Only Valid when the CX_TPH_ENABLE configuration parameter is 1. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LWR_BASE_ADDR_OFF_OUTBOUND_1_0

Description: The CX_ATU_MIN_REGION_SIZE configuration parameter (Value Range: 4 kB, 8 kB, 16 kB, 32 kB, 64 kB defaults to 64 kB) specifies the minimum size of an address translation region. For example, if set to 64 kB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned on 64 kB boundaries. More precisely, the lower log2(CX_ATU_MIN_REGION_SIZE) bits are zero.

Offset: 0x208
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:16	RW	0x0	**LWR_BASE_RW**: Forms bits [31:n] of the start address of the address region to be translated. n is log2(CX_ATU_MIN_REGION_SIZE) Note: This register field is sticky.

Bit	R/W	Reset	Description
15:0	RO	0x0	**LWR_BASE_HW**: Forms bits [n-1:0] of the start address of the address region to be translated. The start address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. n is log2(CX_ATU_MIN_REGION_SIZE)

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_1_0

Description: This register holds the upper 32-bits of the start (and end) address of the address region to be translated.

Offset: 0x20c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UPPER_BASE_RW**: Forms bits [63:32] of the start (and end) address of the address region to be translated. In systems with a 32-bit address space, this register is not used and therefore writing to this register has no effect. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LIMIT_ADDR_OFF_OUTBOUND_1_0

Description: This register holds the end address of the address region to be translated.

Offset: 0x210
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	R/W	Reset	Description
31:16	RW	0x0	**LIMIT_ADDR_RW**: Forms upper bits of the end address of the address region to be translated. Note: This register field is sticky.

Bit	R/W	Reset	Description
15:4	RO	0xfff	**LIMIT_ADDR_HW**: Forms lower bits of the end address of the address region to be translated. The end address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so these bits are always all ones. A write to this location is ignored by the PCIe controller. Note: This register field is sticky.
3:0	RO	0xf	**CBUF_INCR**: Circular Buffer. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_1_0

Description: This register holds the Lower Target part of the new address of the translated region, or the outbound TLP header information, depending on the setting of HEADER_SUBSTITUTE_EN field of IATU_REGION_CTRL_2_VIEWPORT_OFF_OUTBOUND_i register.

Offset: 0x214
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LWR_TARGET_RW_OUTBOUND**: When HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_ is '0' (normal operation): - LWR_TARGET_RW[31:n] forms MSB's of the Lower Target part of the new address of the translated region; - LWR_TARGET_RW[n-1:0] are not used. (The start address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so the lower bits of the start address of the new address of the translated region (bits n-1:0) are always '0'). - n is log2(CX_ATU_MIN_REGION_SIZE). When HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_i is '1': - LWR_TARGET_RW[31:0] forms bytes 8-11 (for 3 dword header) or bytes 12-15 (for 4 dword header) of the outbound TLP header. Usage scenarios include the transmission of Vendor Defined Messages where the controller determines the content of bytes 12 to 15 of the TLP header. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_1_0

Description: This register holds the upper 32 bits of the start address (Upper Target part) of the new address of the translated region.

Offset: 0x218
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UPPER_TARGET_RW**: Forms bits [63:32] of the start address (Upper Target part) of the new address of the translated region. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPR_LIMIT_ADDR_OFF_OUTBOUND_1_0

Description: The CX_MAX_ATU_REGION_SIZE configuration parameter (Value Range: 0->32, 8=1TB) specifies the maximum region size of an address translation region. This register is only used when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_OUTBOUND_i is '1'.

Offset: 0x220
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:3	RO	0x0	**UPPR_LIMIT_ADDR_HW**: Forms MSB's of the Upper Limit part of the region "end address" to be translated. Only applies to 64-bit systems and when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_OUTBOUND_i is '1'. These bits are always '0'.
2:0	RW	0x0	**UPPR_LIMIT_ADDR_RW**: Forms the LSB's of the Upper Limit part of the region "end address" to be translated. Only applies to 64-bit systems and when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_OUTBOUND_i is '1' Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_REGION_CTRL_1_OFF_INBOUND_1_0

Description: This register controls the iATU inbound region access based on the optional iATU inbound features enabled using iATU Region Control 2 Register.

Offset: 0x300
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,x000,xxxx,xx0x,x000,0000,0000)

Bit	Reset	Description
22:20	0x0	**CTRL_1_FUNC_NUM**: Function Number. - MEM-I/O: When the Address and BAR matching logic in the controller indicate that a MEM-I/O transaction matches a BAR in the function corresponding to this value, then address translation proceeds. This check is only performed if the "Function Number Match Enable" bit of the "iATU Region Control 2 Register" is set. - CFG0/CFG1: When the destination function number as specified in the routing ID of the TLP header matches the function, then address translation proceeds. This check is only performed if the "Function Number Match Enable" bit of the "iATU Region Control 2 Register" is set. Note: This register field is sticky.
13	0x0	**INCREASE_REGION_SIZE**: Increase the maximum ATU Region size. When set, the maximum ATU Region size is determined by CX_ATU_MAX_REGION_SIZE. When clear, the maximum ATU Region size is 4 GB (default). Note: This register field is sticky.
10:9	0x0	**ATTR**: When the ATTR field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "ATTR Match Enable" bit of the "iATU Region Control 2 Register" is set. Note: This register field is sticky.
8	0x0	**TD**: When the TD field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "TD Match Enable" bit of the "iATU Region Control 2 Register" is set. Note: This register field is sticky.
7:5	0x0	**TC**: When the TC field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "TC Match Enable" bit of the "iATU Region Control 2 Register" is set. Note: This register field is sticky.
4:0	0x0	**TYPE**: When the TYPE field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_REGION_CTRL_2_OFF_INBOUND_1_0

Description: Using this register you can enable/disable the inbound iATU optional features.

Offset: 0x304
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0x00,0x0x,0xx0,000x,x000,0000,0000)

Bit	Reset	Description
31	0x0	**REGION_EN**: Region Enable. This bit must be set to '1' for address translation to take place. Note: This register field is sticky.
30	0x0	**MATCH_MODE**: Match Mode. Determines Inbound matching mode for TLPs. The mode depends on the type of TLP that is received as follows: For MEM-I/O TLPs, this field is interpreted as follows: -0: Address Match Mode. The iATU operates using addresses as in the outbound direction. The Region Base and Limit Registers must be setup. -1: BAR Match Mode. BAR matching is used. The "BAR Number" field is relevant. Not used for RC. For CFG0 TLPs, this field is interpreted as follows: -0: Routing ID Match Mode. The iATU interprets the Routing ID (Bytes 8 to 11 of TLP header) as an address. This corresponds to the upper 16 bits of the address in MEM-I/O transactions. The Routing ID of the TLP must be within the base and limit of the iATU region for matching to proceed. -1: Accept Mode. The iATU accepts all CFG0 transactions as address matches. The routing ID in the CFG0 TLP is ignored. This is useful as all received CFG0 TLPs should be processed regardless of the Bus number. For MSG/MSGD TLPs, this field is interpreted as follows: -0: Address Match Mode. The iATU treats the third dword and fourth dword of the inbound MSG/MSGD TLP as an address and it is matched against the Region Base and Limit Registers. -1: Vendor ID Match Mode. This mode is relevant for ID-routed Vendor Defined Messages. The iATU ignores the Routing ID (Bus, Device, Function) in bits [31:16] of the third dword of the TLP header, but matches against the Vendor ID in bits [15:0] of the third dword of the TLP header. Bits [15:0] of the Region Upper Base register should be programmed with the required Vendor ID. The lower Base and Limit Register should be programmed to translate TLPs based on vendor specific information in the fourth dword of the TLP header. - If SINGLE_ADDRESS_LOCATION_TRANSLATE_EN = 1 AND MSG_TYPE_MATCH_MODE =1, then Match Mode is ignored. Note: This register field is sticky.
29	0x0	**INVERT_MODE**: Invert Mode Enable. When set the address matching region is inverted. Therefore, an address match occurs when the untranslated address is in the region outside the defined range (Base Address to Limit Address). When set all regions of that type must use address match mode. Note: This register field is sticky.
28	0x0	**CFG_SHIFT_MODE**: CFG Shift Enable. This is useful for CFG transactions where the PCIe configuration mechanism maps bits [27:12] of the address to the bus/device and function number. This allows a CFG configuration space to be located in any 256MB window of your application memory space using a 28-bit effective address. Shifts bits [31:16] of the untranslated address to form bits [27:12] of the translated address. Note: This register field is sticky.
27	0x0	**FUZZY_TYPE_MATCH_CODE**: Fuzzy Type Match Enable. When enabled, the iATU relaxes the matching of the TLP TYPE field against the expected TYPE field so that - CfgRd0 and CfgRd1 TLPs are seen as identical. Similarly with CfgWr0 and CfgWr1. - MWr, MRd, and MRdLk TLPs are seen as identical - The Routing field of Msg/MsgD TLPs is ignored - FetchAdd, Swap, and CAS are seen as identical. For example, CFG0 in the TYPE field in the "iATU Region Control 1 Register" matches against an inbound CfgRd0, CfgRd1, CfgWr0, or CfgWr1 TLP. Note: This register field is sticky.
25:24	0x0	**RESPONSE_CODE**: Response Code. Defines the type of response to give for accesses matching this region. This overrides the normal RADM filter response. Note that this feature is not available for any region where Single Address Location Translate is enabled. Note: This register field is sticky.

Bit	Reset	Description
23	0x0	**SINGLE_ADDR_LOC_TRANS_EN**: Single Address Location Translate Enable. When enabled, Rx TLPs can be translated to a single address location as determined by the target address register of the iATU region. The main usage scenario is translation of Messages (such as Vendor Defined or ATS Messages) to MWr TLPs when the AXI bridge is enabled. Note: This register field is sticky.
21	0x0	**MSG_CODE_MATCH_EN**: Message Code Match Enable (Msg TLPS). Ensures that a successful message Code TLP field comparison match (see Message Code field of the "iATU Region Control 2 Register") occurs (in MSG transactions) for address translation to proceed. ST Match Enable (Mem TLPs). Ensures that a successful ST TLP field comparison match (see ST field of the "iATU Region Control 2 Register") occurs (in MEM transactions) for address translation to proceed. Only Valid when the CX_TPH_ENABLE configuration parameter is '1'. Note: This register field is sticky.
19	0x0	**FUNC_NUM_MATCH_EN**: Function Number Match Enable. Ensures that a successful Function Number TLP field comparison match (see Function Number field of the "iATU Region Control 1 Register") occurs (in MEM-I/O and CFG0/CFG1 transactions) for address translation to proceed. Note: This register field is sticky.
16	0x0	**ATTR_MATCH_EN**: ATTR Match Enable. Ensures that a successful ATTR TLP field comparison match (see ATTR field of the "iATU Region Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.
15	0x0	**TD_MATCH_EN**: TD Match Enable. Ensures that a successful TD TLP field comparison match (see TD field of the "iATU Region Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.
14	0x0	**TC_MATCH_EN**: TC Match Enable. Ensures that a successful TC TLP field comparison match (see TC field of the "iATU Region Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.
13	0x0	**MSG_TYPE_MATCH_MODE**: Message Type Match Mode. When enabled, and if single address location translate enable is set, then inbound TLPs of type MSG/MSGd which match the type field of the IATU_REGION_CTRL_1_VIEWPORT_OFF_INBOUND_i register (TYPE[4:3]=2'b10) will be translated. Message type match mode overrides any value of MATCH_MODE field in this register. Usage scenarios for this are translation of VDM or ATS messages when AXI bridge is configured on client interface. Note: This register field is sticky.
10:8	0x0	**BAR_NUM**: BAR Number. When the BAR number of an inbound MEM or IO TLP " that is matched by the normal internal BAR address matching mechanism " is the same as this field, address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "Match Mode" bit of the "iATU Region Control 2 Register" is set. IO translation would require either 00100b or 00101b in the inbound TLP TYPE; the BAR Number set in the range 000b - 101b and that BAR configured as an IO BAR. Note: This register field is sticky.

Bit	Reset	Description
7:0	0x0	**MSG_CODE**: MSG TLPs: (Message Code). When the TYPE field of an inbound Msg/MsgD TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "Message Code Match Enable" bit of the "iATU Region Control 2 Register" is set. Memory TLPs: (ST: Steering Tag). When the ST field of an inbound TLP is matched to this value, then address translation proceeds. This check is only performed if the "ST Match Enable" bit of the "iATU Region Control 2 Register" is set. The setting is independent of the setting of the TH field. Only Valid when the CX_TPH_ENABLE configuration parameter is '1'. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LWR_BASE_ADDR_OFF_INBOUND_1_0

Description: The CX_ATU_MIN_REGION_SIZE configuration parameter (Value Range: 4 KiB, 8 KiB, 16 KiB, 32 KiB, 64 KiB defaults to 64 KiB) specifies the minimum size of an address translation region. For example, if set to 64 KiB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned on 64 KiB boundaries. More precisely, the lower log2(CX_ATU_MIN_REGION_SIZE) bits are zero.

Offset: 0x308
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:16	RW	0x0	**LWR_BASE_RW**: Forms bits [31:n] of the start address of the address region to be translated. n is log2(CX_ATU_MIN_REGION_SIZE) Note: This register field is sticky.
15:0	RO	0x0	**LWR_BASE_HW**: Forms bits [n-1:0] of the start address of the address region to be translated. The start address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. n is log2(CX_ATU_MIN_REGION_SIZE)

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPER_BASE_ADDR_OFF_INBOUND_1_0

Description: This register holds the upper 32 bits of the start (and end) address of the address region to be translated.

Offset: 0x30c
Read/Write: R/W

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UPPER_BASE_RW**: Forms bits [63:32] of the start (and end) address of the address region to be translated. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LIMIT_ADDR_OFF_INBOUND_1_0

Description: This register holds the end address of the address region to be translated.

Offset: 0x310
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000fff0 (0b0000,0000,0000,0000,1111,1111,1111,0000)

Bit	R/W	Reset	Description
31:16	RW	0x0	**LIMIT_ADDR_RW**: Forms upper bits of the end address of the address region to be translated. When SINGLE_ADDR_LOC_TRANS_EN = 1, MSG_TYPE_MATCH_MODE = 1, and CBUF_INCR > 0 then this field forms the upper bits of the limit address for the circular buffer. Note: This register field is sticky.
15:4	RO	0xfff	**LIMIT_ADDR_HW**: Forms lower bits of the end address of the address region to be translated. The end address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so these bits are always all ones. When SINGLE_ADDR_LOC_TRANS_EN = 1, MSG_TYPE_MATCH_MODE = 1, and CBUF_INCR > 0 then this field forms lower bits of the limit address for the circular buffer. A write to this location is ignored by the PCIe controller. Note: This register field is sticky.

Bit	R/W	Reset	Description
3:0	RW	0x0	**CBUF_INCR**: Circular Buffer Increment. When CX_ATU_SLOC_CBUF = 0, then this field is Read-only and forms the lowest bits of the end address of the address region to be translated. When CX_ATU_SLOC_CBUF = 1, then this field is R/W and forms the upper bits of the Circular Buffer Increment size (CBUF_INCR) field for Single Location Address translation. The increment value (in bytes) is decoded as follows: - 0000b: 0 (Default; legacy Single Address Location mode) - 0001b: 4 - 0010b: 8 - 0011b: 16 - 0100b: 32 - 0101b: 64 - 0110b: 128 - 0111b: 256 - 1000b: 512 - 1001b: 1024 - 1010b: 2048 - 1011b: 4096 - 1100b: 8192 - 1101b: rsvd. - 1110b: rsvd. - 1111b: rsvd. Note: A write to any bit in the CBUF_INCR field resets the circular buffer pointer - that is, the next matched received Message will be translated to the start address of the Circular Buffer. This field must be written to AFTER the target and limit registers have been updated. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R (sticky) Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LWR_TARGET_ADDR_OFF_INBOUND_1_0

Description: This register holds the Lower Target part of the new address of the translated region.

Offset: 0x314
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:16	RW	0x0	**LWR_TARGET_RW**: Forms MSB's of the Lower Target part of the new address of the translated region. These bits are always '0'. - Field size depends on log2(CX_ATU_MIN_REGION_SIZE) in address match mode. - Field size depends on log2(BAR_MASK+1) in BAR match mode. Note: This register field is sticky.
15:0	RO	0x0	**LWR_TARGET_HW**: Forms the LSB's of the Lower Target part of the new address of the translated region. The start address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary (in address match mode); and to the Bar size boundary (in BAR match mode) so that these bits are always '0'. If the BAR is smaller than the iATU region size, then the iATU target address must align to the iATU region size; otherwise it must align to the BAR size. A write to this location is ignored by the PCIe controller. - Field size depends on log2(CX_ATU_MIN_REGION_SIZE) in address match mode. - Field size depends on log2(BAR_MASK+1) in BAR match mode.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPER_TARGET_ADDR_OFF_INBOUND_1_0

Description: This register holds the upper 32 bits of the start address (Upper Target part) of the new address of the translated region.

Offset: 0x318
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UPPER_TARGET_RW**: Forms bits [63:32] of the start address (Upper Target part) of the new address of the translated region. In systems with a 32-bit address space, this register is not used and therefore writing to this register has no effect. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPR_LIMIT_ADDR_OFF_INBOUND_1_0

Description: The CX_MAX_ATU_REGION_SIZE configuration parameter (Value Range: 0(4GB) to 32(16 EB)) specifies the maximum region size of an address translation region. This register is only used when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_INBOUND_i is '1'.

Offset: 0x320
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:3	RO	0x0	**UPPR_LIMIT_ADDR_HW**: Forms MSB's of the Upper Limit part of the region "end address" to be translated. Only applies to 64-bit systems and when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_INBOUND_i is '1'. These bits are always '0'.
2:0	RW	0x0	**UPPR_LIMIT_ADDR_RW**: Forms the LSB's of the Upper Limit part of the region "end address" to be translated. Only applies to 64-bit systems and when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_INBOUND_i is '1' Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_REGION_CTRL_1_OFF_OUTBOUND_2_0

Description: This register controls the iATU outbound region access based on the optional iATU outbound features enabled using iATU Region Control 2 Register.

Offset: 0x400
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,x000,xxxx,xx0x,x000,0000,0000)

Bit	Reset	Description
22:20	0x0	**CTRL_1_FUNC_NUM**: Function Number. - When the address of an outbound TLP is matched to this region and the FUNC_BYPASS field in the "iATU Region Control 2 Register" is '0', then the function number used in generating the function part of the requester ID (RID) field of the TLP is taken from this 5-bit register. The value in this register must be 0x0 unless multifunction operation in the controller is enabled (CX_NFUNC > 1). - When you are using the AXI Bridge, then this field is swapped before AXI decomposition occurs so that the correct "Max_Read_Request_Size" and "Max_Payload_Size" values are used. Note: This register field is sticky.
13	0x0	**INCREASE_REGION_SIZE**: Increase the maximum ATU Region size. When set, the maximum ATU Region size is determined by CX_ATU_MAX_REGION_SIZE. When clear, the maximum ATU Region size is 4 GB (default). Note: This register field is sticky.
10:9	0x0	**ATTR**: When the address of an outbound TLP is matched to this region, then the ATTR field of the TLP is changed to the value in this register. Note: This register field is sticky.
8	0x0	**TD**: This is a reserved field. Do not use. Note: This register field is sticky.
7:5	0x0	**TC**: When the address of an outbound TLP is matched to this region, then the TC field of the TLP is changed to the value in this register. Note: This register field is sticky.
4:0	0x0	**TYPE**: When the address of an outbound TLP is matched to this region, then the TYPE field of the TLP is changed to the value in this register. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_REGION_CTRL_2_OFF_OUTBOUND_2_0

Description: Using this register you can enable/disable the outbound iATU optional features.

Offset: 0x404
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0x00,0xxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**REGION_EN**: Region Enable. This bit must be set to '1' for address translation to take place. Note: This register field is sticky.
29	0x0	**INVERT_MODE**: Invert Mode. When set the address matching region is inverted. Therefore, an address match occurs when the untranslated address is in the region outside the defined range (Base Address to Limit Address). Note: This register field is sticky.
28	0x0	**CFG_SHIFT_MODE**: CFG Shift Mode. The iATU uses bits [27:12] of the untranslated address (on the XALI0/1/2 interface or AXI slave interface address) to form the BDF number of the outgoing CFG TLP. This supports the Enhanced Configuration Address Mapping (ECAM) mechanism (Section 7.2.2 of the PCI Express Base 3.1 Specification, revision 1.0) by allowing all outgoing I/O and MEM TLPs (that have been translated to CFG) to be mapped from memory space into any 256 MB region of the PCIe configuration space. Note: This register field is sticky.
27	0x0	**DMA_BYPASS**: DMA Bypass Mode. Allows request TLPs which are initiated by the DMA controller to pass through the iATU untranslated. Note: This field is reserved for the SW product. You must set it to '0'. Note: This register field is sticky.
23	0x0	**HEADER_SUBSTITUTE_EN**: Header Substitute Enable. When enabled and region address is matched, the iATU fully substitutes bytes 8-11 (for 3 DWORD header) or bytes 12-15 (for 4 DWORD header) of the outbound TLP header with the contents of the LWR_TARGET_RW field in IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i. Encodings are as above. Note: This register field is sticky.
22	0x0	**INHIBIT_PAYLOAD**: Inhibit TLP Payload Data for TLP's in Matched Region; assign iATU region to be TLP without data. When enabled and region address is matched, the iATU marks all TLPs as having no payload data by forcing the TLP header Fmt[1] bit =0, regardless of the application inputs such as slv_wstrb. Encoding are define as above. Note: This register field is sticky.
21	0x0	**TLP_HEADER_FIELDS_BYPASS**: TLP Header Fields Translation Bypass. In this mode header fields of the translated TLP is taken from your application transmit interface or, if AMBA is configured, from the AMBA sideband bus (slv_awmisc_info) and not from the corresponding fields of the IATU_REGION_CTRL_1_OFF_OUTBOUND_i or IATU_REGION_CTRL_2_OFF_OUTBOUND_i registers. The header fields are - TC - PH - TH - ST - AT - Attr (IDO, RO and NS). Note: This register field is sticky.
20	0x0	**SNP**: Serialize Non-Posted Requests. In this mode, when the AXI Bridge is populated, same AXI ID Non-Posted Read/Write Requests are transmitted on the wire if there are no other same ID Non-Posted Requests outstanding. Note: This register field is sticky.
19	0x0	**FUNC_BYPASS**: Function Number Translation Bypass. In this mode, the function number of the translated TLP is taken from your application transmit interface and not from the CTRL_1_FUNC_NUM field of the "iATU Region Control 1 Register" or the VF_NUMBER field of the "iATU Control 3 Register." Note: This register field is sticky.

Bit	Reset	Description
18:17	0x0	**MSB2BITS_TAG**: TAG. Two most significant bits of the substituted TAG field in the outgoing TLP header when TAG_SUBSTITUTE_EN is set. Note: This register field is sticky.
16	0x0	**TAG_SUBSTITUTE_EN**: TAG Substitute Enable. When enabled and region address is matched, the iATU substitutes the TAG field of the outbound TLP header with the contents of the TAG field in this register. The expected usage scenario is translation from AXI MWr to Vendor Defined Msg/MsgD. Note (CX_TPH_ENABLE=1): TAG substitution for MWr will not occur because this field (byte 6) in the TLP header is the ST field. ST substitution can still take place using the MSG_CODE field in IATU_REGION_CTRL_2_OFF_OUTBOUND_i. Note: This register field is sticky.
15:8	0x0	**TAG**: TAG. The substituted TAG field (byte 6) in the outgoing TLP header when TAG_SUBSTITUTE_EN is set. Note: This register field is sticky.
7:0	0x0	**MSG_CODE**: MSG TLPs (Message Code). When the address of an outbound TLP is matched to this region, and the translated TLP TYPE field is Msg or MsgD; then the message field of the TLP is changed to the value in this register. Memory TLPs: (ST: Steering Tag). When the ST field of an outbound TLP is matched to this region, and the translated TLP TYPE field targets memory space; then the ST field of the TLP is changed to the value in this register. Only Valid when the CX_TPH_ENABLE configuration parameter is 1. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LWR_BASE_ADDR_OFF_OUTBOUND_2_0

Description: The CX_ATU_MIN_REGION_SIZE configuration parameter (Value Range: 4 kB, 8 kB, 16 kB, 32 kB, 64 kB defaults to 64 kB) specifies the minimum size of an address translation region. For example, if set to 64 kB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned on 64 kB boundaries. More precisely, the lower log2(CX_ATU_MIN_REGION_SIZE) bits are zero.

Offset: 0x408
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:16	RW	0x0	**LWR_BASE_RW**: Forms bits [31:n] of the start address of the address region to be translated. n is log2(CX_ATU_MIN_REGION_SIZE) Note: This register field is sticky.

Bit	R/W	Reset	Description
15:0	RO	0x0	**LWR_BASE_HW**: Forms bits [n-1:0] of the start address of the address region to be translated. The start address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. n is log2(CX_ATU_MIN_REGION_SIZE)

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_2_0

Description: This register holds the upper 32-bits of the start (and end) address of the address region to be translated.

Offset: 0x40c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UPPER_BASE_RW**: Forms bits [63:32] of the start (and end) address of the address region to be translated. In systems with a 32-bit address space, this register is not used and therefore writing to this register has no effect. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LIMIT_ADDR_OFF_OUTBOUND_2_0

Description: This register holds the end address of the address region to be translated.

Offset: 0x410
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	R/W	Reset	Description
31:16	RW	0x0	**LIMIT_ADDR_RW**: Forms upper bits of the end address of the address region to be translated. Note: This register field is sticky.

Bit	R/W	Reset	Description
15:4	RO	0xfff	**LIMIT_ADDR_HW**: Forms lower bits of the end address of the address region to be translated. The end address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so these bits are always all ones. A write to this location is ignored by the PCIe controller. Note: This register field is sticky.
3:0	RO	0xf	**CBUF_INCR**: Circular Buffer. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_2_0

Description: This register holds the Lower Target part of the new address of the translated region, or the outbound TLP header information, depending on the setting of HEADER_SUBSTITUTE_EN field of IATU_REGION_CTRL_2_VIEWPORT_OFF_OUTBOUND_i register.

Offset: 0x414
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LWR_TARGET_RW_OUTBOUND**: When HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_ is '0' (normal operation): - LWR_TARGET_RW[31:n] forms MSB's of the Lower Target part of the new address of the translated region; - LWR_TARGET_RW[n-1:0] are not used. (The start address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so the lower bits of the start address of the new address of the translated region (bits n-1:0) are always '0'). - n is log2(CX_ATU_MIN_REGION_SIZE). When HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_i is '1': - LWR_TARGET_RW[31:0] forms bytes 8-11 (for 3 dword header) or bytes 12-15 (for 4 dword header) of the outbound TLP header. Usage scenarios include the transmission of Vendor Defined Messages where the controller determines the content of bytes 12 to 15 of the TLP header. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_2_0

Description: This register holds the upper 32 bits of the start address (Upper Target part) of the new address of the translated region.

Offset: 0x418
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UPPER_TARGET_RW**: Forms bits [63:32] of the start address (Upper Target part) of the new address of the translated region. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPR_LIMIT_ADDR_OFF_OUTBOUND_2_0

Description: The CX_MAX_ATU_REGION_SIZE configuration parameter (Value Range: 0->32, 8=1TB) specifies the maximum region size of an address translation region. This register is only used when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_OUTBOUND_i is '1'.

Offset: 0x420
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:3	RO	0x0	**UPPR_LIMIT_ADDR_HW**: Forms MSB's of the Upper Limit part of the region "end address" to be translated. Only applies to 64-bit systems and when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_OUTBOUND_i is '1'. These bits are always '0'.
2:0	RW	0x0	**UPPR_LIMIT_ADDR_RW**: Forms the LSB's of the Upper Limit part of the region "end address" to be translated. Only applies to 64-bit systems and when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_OUTBOUND_i is '1' Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_REGION_CTRL_1_OFF_OUTBOUND_3_0

Description: This register controls the iATU outbound region access based on the optional iATU outbound features enabled using iATU Region Control 2 Register.

Offset: 0x600
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,x000,xxxx,xx0x,x000,0000,0000)

Bit	Reset	Description
22:20	0x0	**CTRL_1_FUNC_NUM**: Function Number. - When the address of an outbound TLP is matched to this region and the FUNC_BYPASS field in the "iATU Region Control 2 Register" is '0', then the function number used in generating the function part of the requester ID (RID) field of the TLP is taken from this 5-bit register. The value in this register must be 0x0 unless multifunction operation in the controller is enabled (CX_NFUNC > 1). - When you are using the AXI Bridge, then this field is swapped before AXI decomposition occurs so that the correct "Max_Read_Request_Size" and "Max_Payload_Size" values are used. Note: This register field is sticky.
13	0x0	**INCREASE_REGION_SIZE**: Increase the maximum ATU Region size. When set, the maximum ATU Region size is determined by CX_ATU_MAX_REGION_SIZE. When clear, the maximum ATU Region size is 4 GB (default). Note: This register field is sticky.
10:9	0x0	**ATTR**: When the address of an outbound TLP is matched to this region, then the ATTR field of the TLP is changed to the value in this register. Note: This register field is sticky.
8	0x0	**TD**: This is a reserved field. Do not use. Note: This register field is sticky.
7:5	0x0	**TC**: When the address of an outbound TLP is matched to this region, then the TC field of the TLP is changed to the value in this register. Note: This register field is sticky.
4:0	0x0	**TYPE**: When the address of an outbound TLP is matched to this region, then the TYPE field of the TLP is changed to the value in this register. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_REGION_CTRL_2_OFF_OUTBOUND_3_0

Description: Using this register you can enable/disable the outbound iATU optional features.

Offset: 0x604
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0x00,0xxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**REGION_EN**: Region Enable. This bit must be set to '1' for address translation to take place. Note: This register field is sticky.

Bit	Reset	Description
29	0x0	**INVERT_MODE**: Invert Mode. When set the address matching region is inverted. Therefore, an address match occurs when the untranslated address is in the region outside the defined range (Base Address to Limit Address). Note: This register field is sticky.
28	0x0	**CFG_SHIFT_MODE**: CFG Shift Mode. The iATU uses bits [27:12] of the untranslated address (on the XALI0/1/2 interface or AXI slave interface address) to form the BDF number of the outgoing CFG TLP. This supports the Enhanced Configuration Address Mapping (ECAM) mechanism (Section 7.2.2 of the PCI Express Base 3.1 Specification, revision 1.0) by allowing all outgoing I/O and MEM TLPs (that have been translated to CFG) to be mapped from memory space into any 256 MB region of the PCIe configuration space. Note: This register field is sticky.
27	0x0	**DMA_BYPASS**: DMA Bypass Mode. Allows request TLPs which are initiated by the DMA controller to pass through the iATU untranslated. Note: This field is reserved for the SW product. You must set it to '0'. Note: This register field is sticky.
23	0x0	**HEADER_SUBSTITUTE_EN**: Header Substitute Enable. When enabled and region address is matched, the iATU fully substitutes bytes 8-11 (for 3 DWORD header) or bytes 12-15 (for 4 DWORD header) of the outbound TLP header with the contents of the LWR_TARGET_RW field in IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i. Encodings are as above. Note: This register field is sticky.
22	0x0	**INHIBIT_PAYLOAD**: Inhibit TLP Payload Data for TLP's in Matched Region; assign iATU region to be TLP without data. When enabled and region address is matched, the iATU marks all TLPs as having no payload data by forcing the TLP header Fmt[1] bit =0, regardless of the application inputs such as slv_wstrb. Encoding are define as above. Note: This register field is sticky.
21	0x0	**TLP_HEADER_FIELDS_BYPASS**: TLP Header Fields Translation Bypass. In this mode header fields of the translated TLP is taken from your application transmit interface or, if AMBA is configured, from the AMBA sideband bus (slv_awmisc_info) and not from the corresponding fields of the IATU_REGION_CTRL_1_OFF_OUTBOUND_i or IATU_REGION_CTRL_2_OFF_OUTBOUND_i registers. The header fields are - TC - PH - TH - ST - AT - Attr (IDO, RO and NS). Note: This register field is sticky.
20	0x0	**SNP**: Serialize Non-Posted Requests. In this mode, when the AXI Bridge is populated, same AXI ID Non-Posted Read/Write Requests are transmitted on the wire if there are no other same ID Non-Posted Requests outstanding. Note: This register field is sticky.
19	0x0	**FUNC_BYPASS**: Function Number Translation Bypass. In this mode, the function number of the translated TLP is taken from your application transmit interface and not from the CTRL_1_FUNC_NUM field of the "iATU Region Control 1 Register" or the VF_NUMBER field of the "iATU Control 3 Register." Note: This register field is sticky.
18:17	0x0	**MSB2BITS_TAG**: TAG. Two most significant bits of the substituted TAG field in the outgoing TLP header when TAG_SUBSTITUTE_EN is set. Note: This register field is sticky.

Bit	Reset	Description
16	0x0	**TAG_SUBSTITUTE_EN**: TAG Substitute Enable. When enabled and region address is matched, the iATU substitutes the TAG field of the outbound TLP header with the contents of the TAG field in this register. The expected usage scenario is translation from AXI MWr to Vendor Defined Msg/MsgD. Note (CX_TPH_ENABLE=1): TAG substitution for MWr will not occur because this field (byte 6) in the TLP header is the ST field. ST substitution can still take place using the MSG_CODE field in IATU_REGION_CTRL_2_OFF_OUTBOUND_i. Note: This register field is sticky.
15:8	0x0	**TAG**: TAG. The substituted TAG field (byte 6) in the outgoing TLP header when TAG_SUBSTITUTE_EN is set. Note: This register field is sticky.
7:0	0x0	**MSG_CODE**: MSG TLPs (Message Code). When the address of an outbound TLP is matched to this region, and the translated TLP TYPE field is Msg or MsgD; then the message field of the TLP is changed to the value in this register. Memory TLPs: (ST: Steering Tag). When the ST field of an outbound TLP is matched to this region, and the translated TLP TYPE field targets memory space; then the ST field of the TLP is changed to the value in this register. Only Valid when the CX_TPH_ENABLE configuration parameter is 1. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LWR_BASE_ADDR_OFF_OUTBOUND_3_0

Description: The CX_ATU_MIN_REGION_SIZE configuration parameter (Value Range: 4 kB, 8 kB, 16 kB, 32 kB, 64 kB defaults to 64 kB) specifies the minimum size of an address translation region. For example, if set to 64 kB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned on 64 kB boundaries. More precisely, the lower log2(CX_ATU_MIN_REGION_SIZE) bits are zero.

Offset: 0x608
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:16	RW	0x0	**LWR_BASE_RW**: Forms bits [31:n] of the start address of the address region to be translated. n is log2(CX_ATU_MIN_REGION_SIZE) Note: This register field is sticky.
15:0	RO	0x0	**LWR_BASE_HW**: Forms bits [n-1:0] of the start address of the address region to be translated. The start address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. n is log2(CX_ATU_MIN_REGION_SIZE)

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_3_0

Description: This register holds the upper 32-bits of the start (and end) address of the address region to be translated.

Offset: 0x60c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UPPER_BASE_RW**: Forms bits [63:32] of the start (and end) address of the address region to be translated. In systems with a 32-bit address space, this register is not used and therefore writing to this register has no effect. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LIMIT_ADDR_OFF_OUTBOUND_3_0

Description: This register holds the end address of the address region to be translated.

Offset: 0x610
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	R/W	Reset	Description
31:16	RW	0x0	**LIMIT_ADDR_RW**: Forms upper bits of the end address of the address region to be translated. Note: This register field is sticky.
15:4	RO	0xfff	**LIMIT_ADDR_HW**: Forms lower bits of the end address of the address region to be translated. The end address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so these bits are always all ones. A write to this location is ignored by the PCIe controller. Note: This register field is sticky.
3:0	RO	0xf	**CBUF_INCR**: Circular Buffer. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_3_0

Description: This register holds the Lower Target part of the new address of the translated region, or the outbound TLP header information, depending on the setting of HEADER_SUBSTITUTE_EN field of IATU_REGION_CTRL_2_VIEWPORT_OFF_OUTBOUND_i register.

Offset: 0x614
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LWR_TARGET_RW_OUTBOUND**: When HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_ is '0' (normal operation): - LWR_TARGET_RW[31:n] forms MSB's of the Lower Target part of the new address of the translated region; - LWR_TARGET_RW[n-1:0] are not used. (The start address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so the lower bits of the start address of the new address of the translated region (bits n-1:0) are always '0'). - n is log2(CX_ATU_MIN_REGION_SIZE). When HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_i is '1': - LWR_TARGET_RW[31:0] forms bytes 8-11 (for 3 dword header) or bytes 12-15 (for 4 dword header) of the outbound TLP header. Usage scenarios include the transmission of Vendor Defined Messages where the controller determines the content of bytes 12 to 15 of the TLP header. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_3_0

Description: This register holds the upper 32 bits of the start address (Upper Target part) of the new address of the translated region.

Offset: 0x618
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UPPER_TARGET_RW**: Forms bits [63:32] of the start address (Upper Target part) of the new address of the translated region. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPR_LIMIT_ADDR_OFF_OUTBOUND_3_0

Description: The CX_MAX_ATU_REGION_SIZE configuration parameter (Value Range: 0->32, 8=1TB) specifies the maximum region size of an address translation region. This register is only used when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_OUTBOUND_i is '1'.

Offset: 0x620
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:3	RO	0x0	**UPPR_LIMIT_ADDR_HW**: Forms MSB's of the Upper Limit part of the region "end address" to be translated. Only applies to 64-bit systems and when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_OUTBOUND_i is '1'. These bits are always '0'.
2:0	RW	0x0	**UPPR_LIMIT_ADDR_RW**: Forms the LSB's of the Upper Limit part of the region "end address" to be translated. Only applies to 64-bit systems and when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_OUTBOUND_i is '1' Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_REGION_CTRL_1_OFF_OUTBOUND_4_0

Description: This register controls the iATU outbound region access based on the optional iATU outbound features enabled using iATU Region Control 2 Register.

Offset: 0x800
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,x000,xxxx,xx0x,x000,0000,0000)

Bit	Reset	Description
22:20	0x0	**CTRL_1_FUNC_NUM**: Function Number. - When the address of an outbound TLP is matched to this region and the FUNC_BYPASS field in the "iATU Region Control 2 Register" is '0', then the function number used in generating the function part of the requester ID (RID) field of the TLP is taken from this 5-bit register. The value in this register must be 0x0 unless multifunction operation in the controller is enabled (CX_NFUNC > 1). - When you are using the AXI Bridge, then this field is swapped before AXI decomposition occurs so that the correct "Max_Read_Request_Size" and "Max_Payload_Size" values are used. Note: This register field is sticky.
13	0x0	**INCREASE_REGION_SIZE**: Increase the maximum ATU Region size. When set, the maximum ATU Region size is determined by CX_ATU_MAX_REGION_SIZE. When clear, the maximum ATU Region size is 4 GB (default). Note: This register field is sticky.
10:9	0x0	**ATTR**: When the address of an outbound TLP is matched to this region, then the ATTR field of the TLP is changed to the value in this register. Note: This register field is sticky.
8	0x0	**TD**: This is a reserved field. Do not use. Note: This register field is sticky.
7:5	0x0	**TC**: When the address of an outbound TLP is matched to this region, then the TC field of the TLP is changed to the value in this register. Note: This register field is sticky.
4:0	0x0	**TYPE**: When the address of an outbound TLP is matched to this region, then the TYPE field of the TLP is changed to the value in this register. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_REGION_CTRL_2_OFF_OUTBOUND_4_0

Description: Using this register you can enable/disable the outbound iATU optional features.

Offset: 0x804
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0x00,0xxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**REGION_EN**: Region Enable. This bit must be set to '1' for address translation to take place. Note: This register field is sticky.

Bit	Reset	Description
29	0x0	**INVERT_MODE**: Invert Mode. When set the address matching region is inverted. Therefore, an address match occurs when the untranslated address is in the region outside the defined range (Base Address to Limit Address). Note: This register field is sticky.
28	0x0	**CFG_SHIFT_MODE**: CFG Shift Mode. The iATU uses bits [27:12] of the untranslated address (on the XALI0/1/2 interface or AXI slave interface address) to form the BDF number of the outgoing CFG TLP. This supports the Enhanced Configuration Address Mapping (ECAM) mechanism (Section 7.2.2 of the PCI Express Base 3.1 Specification, revision 1.0) by allowing all outgoing I/O and MEM TLPs (that have been translated to CFG) to be mapped from memory space into any 256 MB region of the PCIe configuration space. Note: This register field is sticky.
27	0x0	**DMA_BYPASS**: DMA Bypass Mode. Allows request TLPs which are initiated by the DMA controller to pass through the iATU untranslated. Note: This field is reserved for the SW product. You must set it to '0'. Note: This register field is sticky.
23	0x0	**HEADER_SUBSTITUTE_EN**: Header Substitute Enable. When enabled and region address is matched, the iATU fully substitutes bytes 8-11 (for 3 DWORD header) or bytes 12-15 (for 4 DWORD header) of the outbound TLP header with the contents of the LWR_TARGET_RW field in IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i. Encodings are as above. Note: This register field is sticky.
22	0x0	**INHIBIT_PAYLOAD**: Inhibit TLP Payload Data for TLP's in Matched Region; assign iATU region to be TLP without data. When enabled and region address is matched, the iATU marks all TLPs as having no payload data by forcing the TLP header Fmt[1] bit =0, regardless of the application inputs such as slv_wstrb. Encoding are define as above. Note: This register field is sticky.
21	0x0	**TLP_HEADER_FIELDS_BYPASS**: TLP Header Fields Translation Bypass. In this mode header fields of the translated TLP is taken from your application transmit interface or, if AMBA is configured, from the AMBA sideband bus (slv_awmisc_info) and not from the corresponding fields of the IATU_REGION_CTRL_1_OFF_OUTBOUND_i or IATU_REGION_CTRL_2_OFF_OUTBOUND_i registers. The header fields are - TC - PH - TH - ST - AT - Attr (IDO, RO and NS). Note: This register field is sticky.
20	0x0	**SNP**: Serialize Non-Posted Requests. In this mode, when the AXI Bridge is populated, same AXI ID Non-Posted Read/Write Requests are transmitted on the wire if there are no other same ID Non-Posted Requests outstanding. Note: This register field is sticky.
19	0x0	**FUNC_BYPASS**: Function Number Translation Bypass. In this mode, the function number of the translated TLP is taken from your application transmit interface and not from the CTRL_1_FUNC_NUM field of the "iATU Region Control 1 Register" or the VF_NUMBER field of the "iATU Control 3 Register." Note: This register field is sticky.
18:17	0x0	**MSB2BITS_TAG**: TAG. Two most significant bits of the substituted TAG field in the outgoing TLP header when TAG_SUBSTITUTE_EN is set. Note: This register field is sticky.

Bit	Reset	Description
16	0x0	**TAG_SUBSTITUTE_EN**: TAG Substitute Enable. When enabled and region address is matched, the iATU substitutes the TAG field of the outbound TLP header with the contents of the TAG field in this register. The expected usage scenario is translation from AXI MWr to Vendor Defined Msg/MsgD. Note (CX_TPH_ENABLE=1): TAG substitution for MWr will not occur because this field (byte 6) in the TLP header is the ST field. ST substitution can still take place using the MSG_CODE field in IATU_REGION_CTRL_2_OFF_OUTBOUND_i. Note: This register field is sticky.
15:8	0x0	**TAG**: TAG. The substituted TAG field (byte 6) in the outgoing TLP header when TAG_SUBSTITUTE_EN is set. Note: This register field is sticky.
7:0	0x0	**MSG_CODE**: MSG TLPs (Message Code). When the address of an outbound TLP is matched to this region, and the translated TLP TYPE field is Msg or MsgD; then the message field of the TLP is changed to the value in this register. Memory TLPs: (ST: Steering Tag). When the ST field of an outbound TLP is matched to this region, and the translated TLP TYPE field targets memory space; then the ST field of the TLP is changed to the value in this register. Only Valid when the CX_TPH_ENABLE configuration parameter is 1. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LWR_BASE_ADDR_OFF_OUTBOUND_4_0

Description: The CX_ATU_MIN_REGION_SIZE configuration parameter (Value Range: 4 kB, 8 kB, 16 kB, 32 kB, 64 kB defaults to 64 kB) specifies the minimum size of an address translation region. For example, if set to 64 kB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned on 64 kB boundaries. More precisely, the lower log2(CX_ATU_MIN_REGION_SIZE) bits are zero.

Offset: 0x808
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:16	RW	0x0	**LWR_BASE_RW**: Forms bits [31:n] of the start address of the address region to be translated. n is log2(CX_ATU_MIN_REGION_SIZE) Note: This register field is sticky.
15:0	RO	0x0	**LWR_BASE_HW**: Forms bits [n-1:0] of the start address of the address region to be translated. The start address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. n is log2(CX_ATU_MIN_REGION_SIZE)

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_4_0

Description: This register holds the upper 32-bits of the start (and end) address of the address region to be translated.

Offset: 0x80c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UPPER_BASE_RW**: Forms bits [63:32] of the start (and end) address of the address region to be translated. In systems with a 32-bit address space, this register is not used and therefore writing to this register has no effect. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LIMIT_ADDR_OFF_OUTBOUND_4_0

Description: This register holds the end address of the address region to be translated.

Offset: 0x810
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	R/W	Reset	Description
31:16	RW	0x0	**LIMIT_ADDR_RW**: Forms upper bits of the end address of the address region to be translated. Note: This register field is sticky.
15:4	RO	0xfff	**LIMIT_ADDR_HW**: Forms lower bits of the end address of the address region to be translated. The end address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so these bits are always all ones. A write to this location is ignored by the PCIe controller. Note: This register field is sticky.
3:0	RO	0xf	**CBUF_INCR**: Circular Buffer. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_4_0

Description: This register holds the Lower Target part of the new address of the translated region, or the outbound TLP header information, depending on the setting of HEADER_SUBSTITUTE_EN field of IATU_REGION_CTRL_2_VIEWPORT_OFF_OUTBOUND_i register.

Offset: 0x814
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LWR_TARGET_RW_OUTBOUND**: When HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_ is '0' (normal operation): - LWR_TARGET_RW[31:n] forms MSB's of the Lower Target part of the new address of the translated region; - LWR_TARGET_RW[n-1:0] are not used. (The start address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so the lower bits of the start address of the new address of the translated region (bits n-1:0) are always '0'). - n is log2(CX_ATU_MIN_REGION_SIZE). When HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_i is '1': - LWR_TARGET_RW[31:0] forms bytes 8-11 (for 3 dword header) or bytes 12-15 (for 4 dword header) of the outbound TLP header. Usage scenarios include the transmission of Vendor Defined Messages where the controller determines the content of bytes 12 to 15 of the TLP header. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_4_0

Description: This register holds the upper 32 bits of the start address (Upper Target part) of the new address of the translated region.

Offset: 0x818
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UPPER_TARGET_RW**: Forms bits [63:32] of the start address (Upper Target part) of the new address of the translated region. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPR_LIMIT_ADDR_OFF_OUTBOUND_4_0

Description: The CX_MAX_ATU_REGION_SIZE configuration parameter (Value Range: 0->32, 8=1TB) specifies the maximum region size of an address translation region. This register is only used when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_OUTBOUND_i is '1'.

Offset: 0x820
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:3	RO	0x0	**UPPR_LIMIT_ADDR_HW**: Forms MSB's of the Upper Limit part of the region "end address" to be translated. Only applies to 64-bit systems and when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_OUTBOUND_i is '1'. These bits are always '0'.
2:0	RW	0x0	**UPPR_LIMIT_ADDR_RW**: Forms the LSB's of the Upper Limit part of the region "end address" to be translated. Only applies to 64-bit systems and when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_OUTBOUND_i is '1' Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_REGION_CTRL_1_OFF_OUTBOUND_5_0

Description: This register controls the iATU outbound region access based on the optional iATU outbound features enabled using iATU Region Control 2 Register.

Offset: 0xa00
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,x000,xxxx,xx0x,x000,0000,0000)

Bit	Reset	Description
22:20	0x0	**CTRL_1_FUNC_NUM**: Function Number. - When the address of an outbound TLP is matched to this region and the FUNC_BYPASS field in the "iATU Region Control 2 Register" is '0', then the function number used in generating the function part of the requester ID (RID) field of the TLP is taken from this 5-bit register. The value in this register must be 0x0 unless multifunction operation in the controller is enabled (CX_NFUNC > 1). - When you are using the AXI Bridge, then this field is swapped before AXI decomposition occurs so that the correct "Max_Read_Request_Size" and "Max_Payload_Size" values are used. Note: This register field is sticky.
13	0x0	**INCREASE_REGION_SIZE**: Increase the maximum ATU Region size. When set, the maximum ATU Region size is determined by CX_ATU_MAX_REGION_SIZE. When clear, the maximum ATU Region size is 4 GB (default). Note: This register field is sticky.
10:9	0x0	**ATTR**: When the address of an outbound TLP is matched to this region, then the ATTR field of the TLP is changed to the value in this register. Note: This register field is sticky.
8	0x0	**TD**: This is a reserved field. Do not use. Note: This register field is sticky.
7:5	0x0	**TC**: When the address of an outbound TLP is matched to this region, then the TC field of the TLP is changed to the value in this register. Note: This register field is sticky.
4:0	0x0	**TYPE**: When the address of an outbound TLP is matched to this region, then the TYPE field of the TLP is changed to the value in this register. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_REGION_CTRL_2_OFF_OUTBOUND_5_0

Description: Using this register you can enable/disable the outbound iATU optional features.

Offset: 0xa04
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0x00,0xxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**REGION_EN**: Region Enable. This bit must be set to '1' for address translation to take place. Note: This register field is sticky.

Bit	Reset	Description
29	0x0	**INVERT_MODE**: Invert Mode. When set the address matching region is inverted. Therefore, an address match occurs when the untranslated address is in the region outside the defined range (Base Address to Limit Address). Note: This register field is sticky.
28	0x0	**CFG_SHIFT_MODE**: CFG Shift Mode. The iATU uses bits [27:12] of the untranslated address (on the XALI0/1/2 interface or AXI slave interface address) to form the BDF number of the outgoing CFG TLP. This supports the Enhanced Configuration Address Mapping (ECAM) mechanism (Section 7.2.2 of the PCI Express Base 3.1 Specification, revision 1.0) by allowing all outgoing I/O and MEM TLPs (that have been translated to CFG) to be mapped from memory space into any 256 MB region of the PCIe configuration space. Note: This register field is sticky.
27	0x0	**DMA_BYPASS**: DMA Bypass Mode. Allows request TLPs which are initiated by the DMA controller to pass through the iATU untranslated. Note: This field is reserved for the SW product. You must set it to '0'. Note: This register field is sticky.
23	0x0	**HEADER_SUBSTITUTE_EN**: Header Substitute Enable. When enabled and region address is matched, the iATU fully substitutes bytes 8-11 (for 3 DWORD header) or bytes 12-15 (for 4 DWORD header) of the outbound TLP header with the contents of the LWR_TARGET_RW field in IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i. Encodings are as above. Note: This register field is sticky.
22	0x0	**INHIBIT_PAYLOAD**: Inhibit TLP Payload Data for TLP's in Matched Region; assign iATU region to be TLP without data. When enabled and region address is matched, the iATU marks all TLPs as having no payload data by forcing the TLP header Fmt[1] bit =0, regardless of the application inputs such as slv_wstrb. Encoding are define as above. Note: This register field is sticky.
21	0x0	**TLP_HEADER_FIELDS_BYPASS**: TLP Header Fields Translation Bypass. In this mode header fields of the translated TLP is taken from your application transmit interface or, if AMBA is configured, from the AMBA sideband bus (slv_awmisc_info) and not from the corresponding fields of the IATU_REGION_CTRL_1_OFF_OUTBOUND_i or IATU_REGION_CTRL_2_OFF_OUTBOUND_i registers. The header fields are - TC - PH - TH - ST - AT - Attr (IDO, RO and NS). Note: This register field is sticky.
20	0x0	**SNP**: Serialize Non-Posted Requests. In this mode, when the AXI Bridge is populated, same AXI ID Non-Posted Read/Write Requests are transmitted on the wire if there are no other same ID Non-Posted Requests outstanding. Note: This register field is sticky.
19	0x0	**FUNC_BYPASS**: Function Number Translation Bypass. In this mode, the function number of the translated TLP is taken from your application transmit interface and not from the CTRL_1_FUNC_NUM field of the "iATU Region Control 1 Register" or the VF_NUMBER field of the "iATU Control 3 Register." Note: This register field is sticky.
18:17	0x0	**MSB2BITS_TAG**: TAG. Two most significant bits of the substituted TAG field in the outgoing TLP header when TAG_SUBSTITUTE_EN is set. Note: This register field is sticky.

Bit	Reset	Description
16	0x0	**TAG_SUBSTITUTE_EN**: TAG Substitute Enable. When enabled and region address is matched, the iATU substitutes the TAG field of the outbound TLP header with the contents of the TAG field in this register. The expected usage scenario is translation from AXI MWr to Vendor Defined Msg/MsgD. Note (CX_TPH_ENABLE=1): TAG substitution for MWr will not occur because this field (byte 6) in the TLP header is the ST field. ST substitution can still take place using the MSG_CODE field in IATU_REGION_CTRL_2_OFF_OUTBOUND_i. Note: This register field is sticky.
15:8	0x0	**TAG**: TAG. The substituted TAG field (byte 6) in the outgoing TLP header when TAG_SUBSTITUTE_EN is set. Note: This register field is sticky.
7:0	0x0	**MSG_CODE**: MSG TLPs (Message Code). When the address of an outbound TLP is matched to this region, and the translated TLP TYPE field is Msg or MsgD; then the message field of the TLP is changed to the value in this register. Memory TLPs: (ST: Steering Tag). When the ST field of an outbound TLP is matched to this region, and the translated TLP TYPE field targets memory space; then the ST field of the TLP is changed to the value in this register. Only Valid when the CX_TPH_ENABLE configuration parameter is 1. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LWR_BASE_ADDR_OFF_OUTBOUND_5_0

Description: The CX_ATU_MIN_REGION_SIZE configuration parameter (Value Range: 4 kB, 8 kB, 16 kB, 32 kB, 64 kB defaults to 64 kB) specifies the minimum size of an address translation region. For example, if set to 64 kB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned on 64 kB boundaries. More precisely, the lower log2(CX_ATU_MIN_REGION_SIZE) bits are zero.

Offset: 0xa08
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:16	RW	0x0	**LWR_BASE_RW**: Forms bits [31:n] of the start address of the address region to be translated. n is log2(CX_ATU_MIN_REGION_SIZE) Note: This register field is sticky.
15:0	RO	0x0	**LWR_BASE_HW**: Forms bits [n-1:0] of the start address of the address region to be translated. The start address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. n is log2(CX_ATU_MIN_REGION_SIZE)

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_5_0

Description: This register holds the upper 32-bits of the start (and end) address of the address region to be translated.

Offset: 0xa0c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UPPER_BASE_RW**: Forms bits [63:32] of the start (and end) address of the address region to be translated. In systems with a 32-bit address space, this register is not used and therefore writing to this register has no effect. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LIMIT_ADDR_OFF_OUTBOUND_5_0

Description: This register holds the end address of the address region to be translated.

Offset: 0xa10
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	R/W	Reset	Description
31:16	RW	0x0	**LIMIT_ADDR_RW**: Forms upper bits of the end address of the address region to be translated. Note: This register field is sticky.
15:4	RO	0xfff	**LIMIT_ADDR_HW**: Forms lower bits of the end address of the address region to be translated. The end address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so these bits are always all ones. A write to this location is ignored by the PCIe controller. Note: This register field is sticky.
3:0	RO	0xf	**CBUF_INCR**: Circular Buffer. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_5_0

Description: This register holds the Lower Target part of the new address of the translated region, or the outbound TLP header information, depending on the setting of HEADER_SUBSTITUTE_EN field of IATU_REGION_CTRL_2_VIEWPORT_OFF_OUTBOUND_i register.

Offset: 0xa14
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LWR_TARGET_RW_OUTBOUND**: When HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_ is '0' (normal operation): - LWR_TARGET_RW[31:n] forms MSB's of the Lower Target part of the new address of the translated region; - LWR_TARGET_RW[n-1:0] are not used. (The start address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so the lower bits of the start address of the new address of the translated region (bits n-1:0) are always '0'). - n is log2(CX_ATU_MIN_REGION_SIZE). When HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_i is '1': - LWR_TARGET_RW[31:0] forms bytes 8-11 (for 3 dword header) or bytes 12-15 (for 4 dword header) of the outbound TLP header. Usage scenarios include the transmission of Vendor Defined Messages where the controller determines the content of bytes 12 to 15 of the TLP header. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_5_0

Description: This register holds the upper 32 bits of the start address (Upper Target part) of the new address of the translated region.

Offset: 0xa18
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UPPER_TARGET_RW**: Forms bits [63:32] of the start address (Upper Target part) of the new address of the translated region. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPR_LIMIT_ADDR_OFF_OUTBOUND_5_0

Description: The CX_MAX_ATU_REGION_SIZE configuration parameter (Value Range: 0->32, 8=1TB) specifies the maximum region size of an address translation region. This register is only used when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_OUTBOUND_i is '1'.

Offset: 0xa20
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:3	RO	0x0	**UPPR_LIMIT_ADDR_HW**: Forms MSB's of the Upper Limit part of the region "end address" to be translated. Only applies to 64-bit systems and when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_OUTBOUND_i is '1'. These bits are always '0'.
2:0	RW	0x0	**UPPR_LIMIT_ADDR_RW**: Forms the LSB's of the Upper Limit part of the region "end address" to be translated. Only applies to 64-bit systems and when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_OUTBOUND_i is '1' Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_REGION_CTRL_1_OFF_OUTBOUND_6_0

Description: This register controls the iATU outbound region access based on the optional iATU outbound features enabled using iATU Region Control 2 Register.

Offset: 0xc00
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,x000,xxxx,xx0x,x000,0000,0000)

Bit	Reset	Description
22:20	0x0	**CTRL_1_FUNC_NUM**: Function Number. - When the address of an outbound TLP is matched to this region and the FUNC_BYPASS field in the "iATU Region Control 2 Register" is '0', then the function number used in generating the function part of the requester ID (RID) field of the TLP is taken from this 5-bit register. The value in this register must be 0x0 unless multifunction operation in the controller is enabled (CX_NFUNC > 1). - When you are using the AXI Bridge, then this field is swapped before AXI decomposition occurs so that the correct "Max_Read_Request_Size" and "Max_Payload_Size" values are used. Note: This register field is sticky.
13	0x0	**INCREASE_REGION_SIZE**: Increase the maximum ATU Region size. When set, the maximum ATU Region size is determined by CX_ATU_MAX_REGION_SIZE. When clear, the maximum ATU Region size is 4 GB (default). Note: This register field is sticky.
10:9	0x0	**ATTR**: When the address of an outbound TLP is matched to this region, then the ATTR field of the TLP is changed to the value in this register. Note: This register field is sticky.
8	0x0	**TD**: This is a reserved field. Do not use. Note: This register field is sticky.
7:5	0x0	**TC**: When the address of an outbound TLP is matched to this region, then the TC field of the TLP is changed to the value in this register. Note: This register field is sticky.
4:0	0x0	**TYPE**: When the address of an outbound TLP is matched to this region, then the TYPE field of the TLP is changed to the value in this register. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_REGION_CTRL_2_OFF_OUTBOUND_6_0

Description: Using this register you can enable/disable the outbound iATU optional features.

Offset: 0xc04
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0x00,0xxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**REGION_EN**: Region Enable. This bit must be set to '1' for address translation to take place. Note: This register field is sticky.

Bit	Reset	Description
29	0x0	**INVERT_MODE**: Invert Mode. When set the address matching region is inverted. Therefore, an address match occurs when the untranslated address is in the region outside the defined range (Base Address to Limit Address). Note: This register field is sticky.
28	0x0	**CFG_SHIFT_MODE**: CFG Shift Mode. The iATU uses bits [27:12] of the untranslated address (on the XALI0/1/2 interface or AXI slave interface address) to form the BDF number of the outgoing CFG TLP. This supports the Enhanced Configuration Address Mapping (ECAM) mechanism (Section 7.2.2 of the PCI Express Base 3.1 Specification, revision 1.0) by allowing all outgoing I/O and MEM TLPs (that have been translated to CFG) to be mapped from memory space into any 256 MB region of the PCIe configuration space. Note: This register field is sticky.
27	0x0	**DMA_BYPASS**: DMA Bypass Mode. Allows request TLPs which are initiated by the DMA controller to pass through the iATU untranslated. Note: This field is reserved for the SW product. You must set it to '0'. Note: This register field is sticky.
23	0x0	**HEADER_SUBSTITUTE_EN**: Header Substitute Enable. When enabled and region address is matched, the iATU fully substitutes bytes 8-11 (for 3 DWORD header) or bytes 12-15 (for 4 DWORD header) of the outbound TLP header with the contents of the LWR_TARGET_RW field in IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i. Encodings are as above. Note: This register field is sticky.
22	0x0	**INHIBIT_PAYLOAD**: Inhibit TLP Payload Data for TLP's in Matched Region; assign iATU region to be TLP without data. When enabled and region address is matched, the iATU marks all TLPs as having no payload data by forcing the TLP header Fmt[1] bit =0, regardless of the application inputs such as slv_wstrb. Encoding are define as above. Note: This register field is sticky.
21	0x0	**TLP_HEADER_FIELDS_BYPASS**: TLP Header Fields Translation Bypass. In this mode header fields of the translated TLP is taken from your application transmit interface or, if AMBA is configured, from the AMBA sideband bus (slv_awmisc_info) and not from the corresponding fields of the IATU_REGION_CTRL_1_OFF_OUTBOUND_i or IATU_REGION_CTRL_2_OFF_OUTBOUND_i registers. The header fields are - TC - PH - TH - ST - AT - Attr (IDO, RO and NS). Note: This register field is sticky.
20	0x0	**SNP**: Serialize Non-Posted Requests. In this mode, when the AXI Bridge is populated, same AXI ID Non-Posted Read/Write Requests are transmitted on the wire if there are no other same ID Non-Posted Requests outstanding. Note: This register field is sticky.
19	0x0	**FUNC_BYPASS**: Function Number Translation Bypass. In this mode, the function number of the translated TLP is taken from your application transmit interface and not from the CTRL_1_FUNC_NUM field of the "iATU Region Control 1 Register" or the VF_NUMBER field of the "iATU Control 3 Register." Note: This register field is sticky.
18:17	0x0	**MSB2BITS_TAG**: TAG. Two most significant bits of the substituted TAG field in the outgoing TLP header when TAG_SUBSTITUTE_EN is set. Note: This register field is sticky.

Bit	Reset	Description
16	0x0	**TAG_SUBSTITUTE_EN**: TAG Substitute Enable. When enabled and region address is matched, the iATU substitutes the TAG field of the outbound TLP header with the contents of the TAG field in this register. The expected usage scenario is translation from AXI MWr to Vendor Defined Msg/MsgD. Note (CX_TPH_ENABLE=1): TAG substitution for MWr will not occur because this field (byte 6) in the TLP header is the ST field. ST substitution can still take place using the MSG_CODE field in IATU_REGION_CTRL_2_OFF_OUTBOUND_i. Note: This register field is sticky.
15:8	0x0	**TAG**: TAG. The substituted TAG field (byte 6) in the outgoing TLP header when TAG_SUBSTITUTE_EN is set. Note: This register field is sticky.
7:0	0x0	**MSG_CODE**: MSG TLPs (Message Code). When the address of an outbound TLP is matched to this region, and the translated TLP TYPE field is Msg or MsgD; then the message field of the TLP is changed to the value in this register. Memory TLPs: (ST: Steering Tag). When the ST field of an outbound TLP is matched to this region, and the translated TLP TYPE field targets memory space; then the ST field of the TLP is changed to the value in this register. Only Valid when the CX_TPH_ENABLE configuration parameter is 1. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LWR_BASE_ADDR_OFF_OUTBOUND_6_0

Description: The CX_ATU_MIN_REGION_SIZE configuration parameter (Value Range: 4 kB, 8 kB, 16 kB, 32 kB, 64 kB defaults to 64 kB) specifies the minimum size of an address translation region. For example, if set to 64 kB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned on 64 kB boundaries. More precisely, the lower log2(CX_ATU_MIN_REGION_SIZE) bits are zero.

Offset: 0xc08
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:16	RW	0x0	**LWR_BASE_RW**: Forms bits [31:n] of the start address of the address region to be translated. n is log2(CX_ATU_MIN_REGION_SIZE) Note: This register field is sticky.
15:0	RO	0x0	**LWR_BASE_HW**: Forms bits [n-1:0] of the start address of the address region to be translated. The start address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. n is log2(CX_ATU_MIN_REGION_SIZE)

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_6_0

Description: This register holds the upper 32-bits of the start (and end) address of the address region to be translated.

Offset: 0xc0c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UPPER_BASE_RW**: Forms bits [63:32] of the start (and end) address of the address region to be translated. In systems with a 32-bit address space, this register is not used and therefore writing to this register has no effect. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LIMIT_ADDR_OFF_OUTBOUND_6_0

Description: This register holds the end address of the address region to be translated.

Offset: 0xc10
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	R/W	Reset	Description
31:16	RW	0x0	**LIMIT_ADDR_RW**: Forms upper bits of the end address of the address region to be translated. Note: This register field is sticky.
15:4	RO	0xfff	**LIMIT_ADDR_HW**: Forms lower bits of the end address of the address region to be translated. The end address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so these bits are always all ones. A write to this location is ignored by the PCIe controller. Note: This register field is sticky.
3:0	RO	0xf	**CBUF_INCR**: Circular Buffer. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_6_0

Description: This register holds the Lower Target part of the new address of the translated region, or the outbound TLP header information, depending on the setting of HEADER_SUBSTITUTE_EN field of IATU_REGION_CTRL_2_VIEWPORT_OFF_OUTBOUND_i register.

Offset: 0xc14
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LWR_TARGET_RW_OUTBOUND**: When HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_ is '0' (normal operation): - LWR_TARGET_RW[31:n] forms MSB's of the Lower Target part of the new address of the translated region; - LWR_TARGET_RW[n-1:0] are not used. (The start address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so the lower bits of the start address of the new address of the translated region (bits n-1:0) are always '0'). - n is log2(CX_ATU_MIN_REGION_SIZE). When HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_i is '1': - LWR_TARGET_RW[31:0] forms bytes 8-11 (for 3 dword header) or bytes 12-15 (for 4 dword header) of the outbound TLP header. Usage scenarios include the transmission of Vendor Defined Messages where the controller determines the content of bytes 12 to 15 of the TLP header. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_6_0

Description: This register holds the upper 32 bits of the start address (Upper Target part) of the new address of the translated region.

Offset: 0xc18
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UPPER_TARGET_RW**: Forms bits [63:32] of the start address (Upper Target part) of the new address of the translated region. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPR_LIMIT_ADDR_OFF_OUTBOUND_6_0

Description: The CX_MAX_ATU_REGION_SIZE configuration parameter (Value Range: 0->32, 8=1TB) specifies the maximum region size of an address translation region. This register is only used when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_OUTBOUND_i is '1'.

Offset: 0xc20
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:3	RO	0x0	**UPPR_LIMIT_ADDR_HW**: Forms MSB's of the Upper Limit part of the region "end address" to be translated. Only applies to 64-bit systems and when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_OUTBOUND_i is '1'. These bits are always '0'.
2:0	RW	0x0	**UPPR_LIMIT_ADDR_RW**: Forms the LSB's of the Upper Limit part of the region "end address" to be translated. Only applies to 64-bit systems and when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_OUTBOUND_i is '1' Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_REGION_CTRL_1_OFF_OUTBOUND_7_0

Description: This register controls the iATU outbound region access based on the optional iATU outbound features enabled using iATU Region Control 2 Register.

Offset: 0xe00
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,x000,xxxx,xx0x,x000,0000,0000)

Bit	Reset	Description
22:20	0x0	**CTRL_1_FUNC_NUM**: Function Number. - When the address of an outbound TLP is matched to this region and the FUNC_BYPASS field in the "iATU Region Control 2 Register" is '0', then the function number used in generating the function part of the requester ID (RID) field of the TLP is taken from this 5-bit register. The value in this register must be 0x0 unless multifunction operation in the controller is enabled (CX_NFUNC > 1). - When you are using the AXI Bridge, then this field is swapped before AXI decomposition occurs so that the correct "Max_Read_Request_Size" and "Max_Payload_Size" values are used. Note: This register field is sticky.
13	0x0	**INCREASE_REGION_SIZE**: Increase the maximum ATU Region size. When set, the maximum ATU Region size is determined by CX_ATU_MAX_REGION_SIZE. When clear, the maximum ATU Region size is 4 GB (default). Note: This register field is sticky.
10:9	0x0	**ATTR**: When the address of an outbound TLP is matched to this region, then the ATTR field of the TLP is changed to the value in this register. Note: This register field is sticky.
8	0x0	**TD**: This is a reserved field. Do not use. Note: This register field is sticky.
7:5	0x0	**TC**: When the address of an outbound TLP is matched to this region, then the TC field of the TLP is changed to the value in this register. Note: This register field is sticky.
4:0	0x0	**TYPE**: When the address of an outbound TLP is matched to this region, then the TYPE field of the TLP is changed to the value in this register. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_REGION_CTRL_2_OFF_OUTBOUND_7_0

Description: Using this register you can enable/disable the outbound iATU optional features.

Offset: 0xe04
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0x00,0xxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**REGION_EN**: Region Enable. This bit must be set to '1' for address translation to take place. Note: This register field is sticky.

Bit	Reset	Description
29	0x0	**INVERT_MODE**: Invert Mode. When set the address matching region is inverted. Therefore, an address match occurs when the untranslated address is in the region outside the defined range (Base Address to Limit Address). Note: This register field is sticky.
28	0x0	**CFG_SHIFT_MODE**: CFG Shift Mode. The iATU uses bits [27:12] of the untranslated address (on the XALI0/1/2 interface or AXI slave interface address) to form the BDF number of the outgoing CFG TLP. This supports the Enhanced Configuration Address Mapping (ECAM) mechanism (Section 7.2.2 of the PCI Express Base 3.1 Specification, revision 1.0) by allowing all outgoing I/O and MEM TLPs (that have been translated to CFG) to be mapped from memory space into any 256 MB region of the PCIe configuration space. Note: This register field is sticky.
27	0x0	**DMA_BYPASS**: DMA Bypass Mode. Allows request TLPs which are initiated by the DMA controller to pass through the iATU untranslated. Note: This field is reserved for the SW product. You must set it to '0'. Note: This register field is sticky.
23	0x0	**HEADER_SUBSTITUTE_EN**: Header Substitute Enable. When enabled and region address is matched, the iATU fully substitutes bytes 8-11 (for 3 DWORD header) or bytes 12-15 (for 4 DWORD header) of the outbound TLP header with the contents of the LWR_TARGET_RW field in IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i. Encodings are as above. Note: This register field is sticky.
22	0x0	**INHIBIT_PAYLOAD**: Inhibit TLP Payload Data for TLP's in Matched Region; assign iATU region to be TLP without data. When enabled and region address is matched, the iATU marks all TLPs as having no payload data by forcing the TLP header Fmt[1] bit =0, regardless of the application inputs such as slv_wstrb. Encoding are define as above. Note: This register field is sticky.
21	0x0	**TLP_HEADER_FIELDS_BYPASS**: TLP Header Fields Translation Bypass. In this mode header fields of the translated TLP is taken from your application transmit interface or, if AMBA is configured, from the AMBA sideband bus (slv_awmisc_info) and not from the corresponding fields of the IATU_REGION_CTRL_1_OFF_OUTBOUND_i or IATU_REGION_CTRL_2_OFF_OUTBOUND_i registers. The header fields are - TC - PH - TH - ST - AT - Attr (IDO, RO and NS). Note: This register field is sticky.
20	0x0	**SNP**: Serialize Non-Posted Requests. In this mode, when the AXI Bridge is populated, same AXI ID Non-Posted Read/Write Requests are transmitted on the wire if there are no other same ID Non-Posted Requests outstanding. Note: This register field is sticky.
19	0x0	**FUNC_BYPASS**: Function Number Translation Bypass. In this mode, the function number of the translated TLP is taken from your application transmit interface and not from the CTRL_1_FUNC_NUM field of the "iATU Region Control 1 Register" or the VF_NUMBER field of the "iATU Control 3 Register." Note: This register field is sticky.
18:17	0x0	**MSB2BITS_TAG**: TAG. Two most significant bits of the substituted TAG field in the outgoing TLP header when TAG_SUBSTITUTE_EN is set. Note: This register field is sticky.

Bit	Reset	Description
16	0x0	**TAG_SUBSTITUTE_EN**: TAG Substitute Enable. When enabled and region address is matched, the iATU substitutes the TAG field of the outbound TLP header with the contents of the TAG field in this register. The expected usage scenario is translation from AXI MWr to Vendor Defined Msg/MsgD. Note (CX_TPH_ENABLE=1): TAG substitution for MWr will not occur because this field (byte 6) in the TLP header is the ST field. ST substitution can still take place using the MSG_CODE field in IATU_REGION_CTRL_2_OFF_OUTBOUND_i. Note: This register field is sticky.
15:8	0x0	**TAG**: TAG. The substituted TAG field (byte 6) in the outgoing TLP header when TAG_SUBSTITUTE_EN is set. Note: This register field is sticky.
7:0	0x0	**MSG_CODE**: MSG TLPs (Message Code). When the address of an outbound TLP is matched to this region, and the translated TLP TYPE field is Msg or MsgD; then the message field of the TLP is changed to the value in this register. Memory TLPs: (ST: Steering Tag). When the ST field of an outbound TLP is matched to this region, and the translated TLP TYPE field targets memory space; then the ST field of the TLP is changed to the value in this register. Only Valid when the CX_TPH_ENABLE configuration parameter is 1. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LWR_BASE_ADDR_OFF_OUTBOUND_7_0

Description: The CX_ATU_MIN_REGION_SIZE configuration parameter (Value Range: 4 kB, 8 kB, 16 kB, 32 kB, 64 kB defaults to 64 kB) specifies the minimum size of an address translation region. For example, if set to 64 kB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned on 64 kB boundaries. More precisely, the lower log2(CX_ATU_MIN_REGION_SIZE) bits are zero.

Offset: 0xe08
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:16	RW	0x0	**LWR_BASE_RW**: Forms bits [31:n] of the start address of the address region to be translated. n is log2(CX_ATU_MIN_REGION_SIZE) Note: This register field is sticky.
15:0	RO	0x0	**LWR_BASE_HW**: Forms bits [n-1:0] of the start address of the address region to be translated. The start address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. n is log2(CX_ATU_MIN_REGION_SIZE)

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_7_0

Description: This register holds the upper 32-bits of the start (and end) address of the address region to be translated.

Offset: 0xe0c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UPPER_BASE_RW**: Forms bits [63:32] of the start (and end) address of the address region to be translated. In systems with a 32-bit address space, this register is not used and therefore writing to this register has no effect. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LIMIT_ADDR_OFF_OUTBOUND_7_0

Description: This register holds the end address of the address region to be translated.

Offset: 0xe10
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000ffff (0b0000,0000,0000,0000,1111,1111,1111,1111)

Bit	R/W	Reset	Description
31:16	RW	0x0	**LIMIT_ADDR_RW**: Forms upper bits of the end address of the address region to be translated. Note: This register field is sticky.
15:4	RO	0xfff	**LIMIT_ADDR_HW**: Forms lower bits of the end address of the address region to be translated. The end address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so these bits are always all ones. A write to this location is ignored by the PCIe controller. Note: This register field is sticky.
3:0	RO	0xf	**CBUF_INCR**: Circular Buffer. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_7_0

Description: This register holds the Lower Target part of the new address of the translated region, or the outbound TLP header information, depending on the setting of HEADER_SUBSTITUTE_EN field of IATU_REGION_CTRL_2_VIEWPORT_OFF_OUTBOUND_i register.

Offset: 0xe14
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LWR_TARGET_RW_OUTBOUND**: When HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_ is '0' (normal operation): - LWR_TARGET_RW[31:n] forms MSB's of the Lower Target part of the new address of the translated region; - LWR_TARGET_RW[n-1:0] are not used. (The start address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so the lower bits of the start address of the new address of the translated region (bits n-1:0) are always '0'). - n is log2(CX_ATU_MIN_REGION_SIZE). When HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_i is '1': - LWR_TARGET_RW[31:0] forms bytes 8-11 (for 3 dword header) or bytes 12-15 (for 4 dword header) of the outbound TLP header. Usage scenarios include the transmission of Vendor Defined Messages where the controller determines the content of bytes 12 to 15 of the TLP header. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_7_0

Description: This register holds the upper 32 bits of the start address (Upper Target part) of the new address of the translated region.

Offset: 0xe18
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UPPER_TARGET_RW**: Forms bits [63:32] of the start address (Upper Target part) of the new address of the translated region. Note: This register field is sticky.

PCIE_IATU_CAP_PF0_ATU_CAP_IATU_UPPR_LIMIT_ADDR_OFF_OUTBOUND_7_0

Description: The CX_MAX_ATU_REGION_SIZE configuration parameter (Value Range: 0->32, 8=1TB) specifies the maximum region size of an address translation region. This register is only used when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_OUTBOUND_i is '1'.

Offset: 0xe20
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:3	RO	0x0	**UPPR_LIMIT_ADDR_HW**: Forms MSB's of the Upper Limit part of the region "end address" to be translated. Only applies to 64-bit systems and when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_OUTBOUND_i is '1'. These bits are always '0'.
2:0	RW	0x0	**UPPR_LIMIT_ADDR_RW**: Forms the LSB's of the Upper Limit part of the region "end address" to be translated. Only applies to 64-bit systems and when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_OUTBOUND_i is '1' Note: This register field is sticky.

reserve[130167] incr4;

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_CTRL_DATA_ARB_PRIOR_OFF_0

Description: This register is used to control traffic priorities among various sources that are delivered to your application through TRGT1 where 0x0 represents the highest priority. - Non-DMA Rx Requests - DMA Write Channel MRd Requests (DMA data requests and LL element/descriptor access) - DMA Read Channel MRd Requests (LL element/descriptor access) - DMA Read Channel MWr Requests Concurrent traffic from channels with same priority are sorted according to Round-Robin arbitration rules. The arbitration priority defaults to Non-DMA requests (highest), Write Channel MRd, Read Channel MRd, Read Channel MWr. For more details, see the "Internal Architecture" section in the DMA chapter of the Databook.

Offset: 0x20000
Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000688 (0b0000,0000,0000,0000,0000,0110,1000,1000)

Bit	R/W	Reset	Description
31:17	RO	0x0	**RSVDP_17**: Reserved for future use.
16:12	RO	0x0	**VERSION**: Reports the version of Register Map of eDMA.
11:9	RW	0x3	**RDBUFF_TRGT_WEIGHT**: DMA Read Channel MWr Requests. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
8:6	RW	0x2	**RD_CTRL_TRGT_WEIGHT**: DMA Read Channel MRd Requests (for LL element/descriptor access). Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
5:3	RW	0x1	**WR_CTRL_TRGT_WEIGHT**: DMA Write Channel MRd Requests (for DMA data requests and LL element/descriptor access). Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
2:0	RW	0x0	**RTRGT1_WEIGHT**: Non-DMA Rx Requests. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_CTRL_OFF_0

Description: This register provides information regarding the number of configured DMA read and write channels. Note: When this register does not exist, value is fixed to 32'hFFFF_FFFF.

Offset: 0x20008
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00020004 (0b0000,0000,0000,0010,0000,0000,0000,0100)

Bit	R/W	Reset	Description
31:26	RO	0x0	**RSVDP_26**: Reserved for future use.
25	RW	0x0	**DIS_C2W_CACHE_RD**: Disable DMA Read Channel's "completion to memory write" context cache pre-fetch function. Note: For internal debugging only. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

Bit	R/W	Reset	Description
24	RW	0x0	**DIS_C2W_CACHE_WR**: Disable DMA Write Channel's "completion to memory write" context cache pre-fetch function. Note: For internal debugging only. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
23:20	RO	0x0	**RSVDP_20**: Reserved for future use.
19:16	RO	0x2	**NUM_DMA_RD_CHAN**: Number of Read Channels. You can read this register to determine the number of read channels configured.
15:4	RO	0x0	**RSVDP_4**: Reserved for future use.
3:0	RO	0x4	**NUM_DMA_WR_CHAN**: Number of Write Channels. You can read this register to determine the number of write channels configured.

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_WRITE_ENGINE_EN_OFF_0

Description: This register indicates the status of DMA write engine and the status of DMA write engine handshake feature (per-channel).

Offset: 0x2000c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,xxxx,xxxx,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**RSVDP_24**: Reserved for future use.
15:1	RO	0x0	**RSVDP_1**: Reserved for future use.

Bit	R/W	Reset	Description
0	RW	0x0	**DMA_WRITE_ENGINE**: DMA Write Engine Enable. For normal operation, you must initially set this bit to '1', before any other software setup action. You do not need to toggle or rewrite to this bit during normal operation. You should set this bit to '0' when you want to "Soft Reset" the DMA controller write logic. There are three possible reasons for resetting the DMA controller write logic: - The "Abort Interrupt Status" bit is set in the DMA Write Interrupt Status Register (DMA_WRITE_INT_STATUS_OFF), and any of the bits in the DMA Write Error Status Register (DMA_WRITE_ERR_STATUS_OFF) are set. Resetting the DMA controller write logic re-initializes the control logic, ensuring that the next DMA write transfer is executed successfully. - You have executed the procedure outlined in "Stop Bit", after which, the "Abort Interrupt Status" bit is set and the Channel Status(CS) field of the DMA Write Channel Control 1 Register (DMA_CH_CONTROL1_OFF_WRCH_i) is set to "Stopped". Resetting the DMA controller write logic re-initializes the control logic ensuring that the next DMA write transfer is executed successfully. - During software development, when you incorrectly program the DMA write engine. To "Soft Reset" the DMA controller write logic, you must: - De-assert the DMA write engine enable bit. - Wait for the DMA to complete any in-progress TLP transfer, by waiting until a read on the DMA write engine enable bit returns a '0'. - Assert the DMA write engine enable bit. This "Soft Reset" does not clear the DMA configuration registers. The DMA write transfer does not start until you write to the DMA Write Doorbell Register (DMA_WRITE_DOORBELL_OFF). Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_WRITE_DOORBELL_OFF_0

Description: This register controls the DMA write channel doorbell.

Offset: 0x20010
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**WR_STOP**: Stop. Set in conjunction with the Doorbell Number field. The DMA write channel stops issuing requests, sets the channel status to "Stopped", and asserts the "Abort" interrupt if it is enabled. Before setting the Stop bit, you must read the channel Status field (CS) of the DMA Write Channel Control 1 Register (DMA_CH_CONTROL1_OFF_WRCH_0) to ensure that the write channel is "Running" (transferring data). For more information, see "Stopping the DMA Transfer (Software Stop)". Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
30:3	RO	0x0	**RSVDP_3**: Reserved for future use.

Bit	R/W	Reset	Description
2:0	RW	0x0	**WR_DOORBELL_NUM**: Doorbell Number. You must write the channel number to this register to start the DMA write transfer for that channel. The DMA detects a write to this register field even if the value of this field does not change. You do not need to toggle or write any other value to this register to start a new transfer. The range of this field is 0x0 to 0x7, where 0x0 corresponds to channel 0. Note: A write to this field triggers the controller to exit L1 substates. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_WRITE_CHANNEL_ARB_WEIGHT_LOW_OFF_0

Description: The 5-bit channel weight (for write channels 0-3) specifies the maximum number of TLP requests that the DMA can issue for that channel before it must return to the arbitration routine. When the channel weight count is reached or DMA channel request transfer size reaches zero, the WWR arbiter selects the next channel to be processed. Your software must initialize this register before ringing the doorbell. For more details, see "Multichannel Arbitration". Value range is (0-0x1F) corresponding to (1-32) transaction requests.

Offset: 0x20018
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00008421 (0b0000,0000,0000,0000,1000,0100,0010,0001)

Bit	R/W	Reset	Description
31:20	RO	0x0	**RSVDP_20**: Reserved for future use.
19:15	RW	0x1	**WRITE_CHANNEL3_WEIGHT**: Channel 3 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. A value of '0' means that one TLP is issued before moving to the next channel. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
14:10	RW	0x1	**WRITE_CHANNEL2_WEIGHT**: Channel 2 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. A value of '0' means that one TLP is issued before moving to the next channel. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

Bit	R/W	Reset	Description
9:5	RW	0x1	**WRITE_CHANNEL1_WEIGHT**: Channel 1 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. A value of '0' means that one TLP is issued before moving to the next channel. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
4:0	RW	0x1	**WRITE_CHANNEL0_WEIGHT**: Channel 0 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. A value of '0' means that one TLP is issued before moving to the next channel. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_WRITE_CHANNEL_ARB_WEIGHT_HIGH_OFF_0

Description: The 5-bit channel weight (for write channels 4-7) specifies the maximum number of TLP requests that the DMA can issue for that channel before it must return to the arbitration routine. When the channel weight count is reached or DMA channel request transfer size reaches zero, the WWR arbiter selects the next channel to be processed. Your software must initialize this register before ringing the doorbell. For more details, see "Multichannel Arbitration". Value range is (0-0x1F) corresponding to (1-32) transaction requests.

Offset: 0x2001c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00008421 (0b0000,0000,0000,0000,1000,0100,0010,0001)

Bit	R/W	Reset	Description
31:20	RO	0x0	**RSVDP_20**: Reserved for future use.
19:15	RW	0x1	**WRITE_CHANNEL7_WEIGHT**: Channel 7 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. A value of '0' means that one TLP is issued before moving to the next channel. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
14:10	RW	0x1	**WRITE_CHANNEL6_WEIGHT**: Channel 6 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. A value of '0' means that one TLP is issued before moving to the next channel. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

Bit	R/W	Reset	Description
9:5	RW	0x1	**WRITE_CHANNEL5_WEIGHT**: Channel 5 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. A value of '0' means that one TLP is issued before moving to the next channel. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
4:0	RW	0x1	**WRITE_CHANNEL4_WEIGHT**: Channel 4 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. A value of '0' means that one TLP is issued before moving to the next channel. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_READ_ENGINE_EN_OFF_0

Description: This register indicates the status of DMA read engine and the status of DMA read engine handshake feature (per-channel).

Offset: 0x2002c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,xxxx,xxxx,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**RSVDP_24**: Reserved for future use.
15:1	RO	0x0	**RSVDP_1**: Reserved for future use.

Bit	R/W	Reset	Description
0	RW	0x0	**DMA_READ_ENGINE**: DMA Read Engine Enable. For normal operation, you must initially set this bit to '1', before any other software setup action. You do not need to toggle or rewrite to this bit during normal operation. You should set this field to '0' when you want to "Soft Reset" the DMA controller read logic. There are three possible reasons for resetting the DMA controller read logic: - The "Abort Interrupt Status" bit is set in the DMA Read Interrupt Status Register (DMA_READ_INT_STATUS_OFF), and any of the bits in the DMA Read Error Status Low Register (DMA_READ_ERR_STATUS_LOW_OFF) is set. Resetting the DMA controller read logic re-initializes the control logic, ensuring that the next DMA read transfer is executed successfully. - You have executed the procedure outlined in "Stop Bit", after which, the "Abort Interrupt Status" bit is set and the channel Status field (CS) of the DMA Read Channel Control 1 Register (DMA_CH_CONTROL1_OFF_WRCH_i) is set to "Stopped". Resetting the DMA controller read logic re-initializes the control logic ensuring that the next DMA read transfer is executed successfully. - During software development, when you incorrectly program the DMA read engine. To "Soft Reset" the DMA controller read logic, you must: - De-assert the DMA read engine enable bit. - Wait for the DMA to complete any in-progress TLP transfer, by waiting until a read on the DMA read engine enable bit returns a '0'. - Assert the DMA read engine enable bit. This "Soft Reset" does not clear the DMA configuration registers. The DMA read transfer does not start until you write to the DMA Read Doorbell Register (DMA_READ_DOORBELL_OFF). Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_READ_DOORBELL_OFF_0

Description: This register controls the DMA read channel doorbell.

Offset: 0x20030
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**RD_STOP**: Stop. Set in conjunction with the Doorbell Number field. The DMA read channel stops issuing requests, sets the channel status to "Stopped", and asserts the "Abort" interrupt if it is enabled. Before setting the Stop bit, you must read the channel Status field (CS) of the DMA Channel Control 1 Register (DMA_CH_CONTROL1_OFF_RDCH_i) to ensure that the read channel is "Running" (transferring data). For more information, see "Stopping the DMA Transfer (Software Stop)". Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
30:3	RO	0x0	**RSVDP_3**: Reserved for future use.

Bit	R/W	Reset	Description
2:0	RW	0x0	**RD_DOORBELL_NUM**: Doorbell Number. You must write 0x0 to this register to start the DMA read transfer for that channel. The DMA detects a write to this register field even if the value of this field does not change. The range of this field is 0x0 to 0x7, where 0x0 corresponds to channel 0. Note: A write to this field triggers the controller to exit L1 substates. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_READ_CHANNEL_ARB_WEIGHT_LOW_OFF_0

Description: The 5-bit channel weight (for read channels 0-3) specifies the maximum number of TLP requests that the DMA can issue for that channel before it must return to the arbitration routine. When the channel weight count is reached or DMA channel request transfer size reaches zero, the WWR arbiter selects the next channel to be processed. Your software must initialize this register before ringing the doorbell. For more details, see "Multichannel Arbitration". Value range is (0-0x1F) corresponding to (1-32) transaction requests.

Offset: 0x20038
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00008421 (0b0000,0000,0000,0000,1000,0100,0010,0001)

Bit	R/W	Reset	Description
31:20	RO	0x0	**RSVDP_20**: Reserved for future use.
19:15	RW	0x1	**READ_CHANNEL3_WEIGHT**: Channel 3 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
14:10	RW	0x1	**READ_CHANNEL2_WEIGHT**: Channel 2 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
9:5	RW	0x1	**READ_CHANNEL1_WEIGHT**: Channel 1 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

Bit	R/W	Reset	Description
4:0	RW	0x1	**READ_CHANNEL0_WEIGHT**: Channel 0 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_READ_CHANNEL_ARB_WEIGHT_HIGH_OFF_0

Description: The 5-bit channel weight (for read channels 4-7) specifies the maximum number of TLP requests that the DMA can issue for that channel before it must return to the arbitration routine. When the channel weight count is reached or DMA channel request transfer size reaches zero, the WWR arbiter selects the next channel to be processed. Your software must initialize this register before ringing the doorbell. For more details, see "Multichannel Arbitration". Value range is (0-0x1F) corresponding to (1-32) transaction requests.

Offset: 0x2003c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00008421 (0b0000,0000,0000,0000,1000,0100,0010,0001)

Bit	R/W	Reset	Description
31:20	RO	0x0	**RSVDP_20**: Reserved for future use.
19:15	RW	0x1	**READ_CHANNEL7_WEIGHT**: Channel 7 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
14:10	RW	0x1	**READ_CHANNEL6_WEIGHT**: Channel 6 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
9:5	RW	0x1	**READ_CHANNEL5_WEIGHT**: Channel 5 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

Bit	R/W	Reset	Description
4:0	RW	0x1	**READ_CHANNEL4_WEIGHT**: Channel 4 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_WRITE_INT_STATUS_OFF_0

Description: This register indicates the status of Abort and Done interrupts for the DMA write channels.

Offset: 0x2004c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**RSVDP_24**: Reserved for future use.
23:16	RW	0x0	**WR_ABORT_INT_STATUS**: Abort Interrupt Status. The DMA write channel has detected an error, or you manually stopped the transfer as described in "Error Handling Assistance by Remote Software". Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For more details, see "Interrupts and Error Handling". - Masking: The DMA write interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the DMA write interrupt Clear register to clear this interrupt bit. Note: You can write to this register to emulate interrupt generation, during software or hardware testing. A write to the address triggers an interrupt, but the DMA does not set the Done or Abort bits in this register. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
15:8	RO	0x0	**RSVDP_8**: Reserved for future use.
7:0	RW	0x0	**WR_DONE_INT_STATUS**: Done Interrupt Status. The DMA write channel has successfully completed the DMA transfer. For more details, see "Interrupts and Error Handling". Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For more details, see "Interrupts and Error Handling". - Masking: The DMA write interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the DMA write interrupt Clear register to clear this interrupt bit. Note: You can write to this register to emulate interrupt generation, during software or hardware testing. A write to the address triggers an interrupt, but the DMA does not set the Done or Abort bits in this register. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_WRITE_INT_MASK_OFF_0

Description: This register holds the Abort and Done DMA write interrupt mask.

Offset: 0x20054
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000f000f (0b0000,0000,xxxx,1111,0000,0000,xxxx,1111)

Bit	R/W	Reset	Description
31:24	RO	0x0	**RSVDP_24**: Reserved for future use.
19:16	RW	0xf	**WR_ABORT_INT_MASK**: Abort Interrupt Mask. Prevents the Abort interrupt status field in the DMA write interrupt status register from asserting the edma_int output. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
15:8	RO	0x0	**RSVDP_8**: Reserved for future use.
3:0	RW	0xf	**WR_DONE_INT_MASK**: Done Interrupt Mask. Prevents the Done interrupt status field in the DMA write interrupt status register from asserting the edma_int output. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_WRITE_INT_CLEAR_OFF_0

Description: You can write to this register to clear the Abort and Done write interrupts.

Offset: 0x20058
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,xxxx,0000,0000,0000,xxxx,0000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**RSVDP_24**: Reserved for future use.

Bit	R/W	Reset	Description
19:16	WO	0x0	**WR_ABORT_INT_CLEAR**: Abort Interrupt Clear. You must write a 1'b1 to clear the corresponding bit in the Abort interrupt status field of the DMA write interrupt status register. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Note: Reading from this self-clearing register field always returns a '0'.
15:8	RO	0x0	**RSVDP_8**: Reserved for future use.
3:0	WO	0x0	**WR_DONE_INT_CLEAR**: Done Interrupt Clear. You must write a 1'b1 to clear the corresponding bit in the Done interrupt status field of the DMA write interrupt status register. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Note: Reading from this self-clearing register field always returns a '0'.

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_WRITE_ERR_STATUS_OFF_0

Description: This register provides information pertaining to errors incurred during write channel operation.

Offset: 0x2005c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**RSVDP_24**: Reserved for future use.
23:16	0x0	**LINKLIST_ELEMENT_FETCH_ERR_DETECT**: Linked List Element Fetch Error Detected. The DMA write channel has received an error response from the AXI bus (or TRGT1 interface when the AXI Bridge is not used) while reading a linked list element from local memory. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For more details, see "Interrupts and Error Handling". - Masking: The DMA write interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the Abort interrupt field of the DMA Write Interrupt Clear Register (DMA_WRITE_INT_CLEAR_OFF) to clear this error bit.
15:8	0x0	**RSVDP_8**: Reserved for future use.

Bit	Reset	Description
7:0	0x0	**APP_READ_ERR_DETECT**: Application Read Error Detected. The DMA write channel has received an error response from the AXI bus (or TRGT1 interface when the AXI Bridge is not used) while reading data from it. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For more details, see "Interrupts and Error Handling". - Masking: The DMA write interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the Abort interrupt field of the DMA Write Interrupt Clear Register (DMA_WRITE_INT_CLEAR_OFF) to clear this error bit.

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_WRITE_DONE_IMWR_LOW_OFF_0

Description: This register holds the lower 32 bits of the Done IMWr TLP address.

Offset: 0x20060
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_WRITE_DONE_LOW_REG**: The DMA uses this field to generate bits [31:0] of the address field for the Done IMWr TLP. Bits [1:0] must be '00' as this address must be dword aligned. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_WRITE_DONE_IMWR_HIGH_OFF_0

Description: This register holds the higher 32 bits of the Done IMWr TLP address.

Offset: 0x20064
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_WRITE_DONE_HIGH_REG**: The DMA uses this field to generate bits [63:32] of the address field for the Done IMWr TLP. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_WRITE_ABORT_IMWR_LOW_OFF_0

Description: This register holds the lower 32 bits of Abort IMWr TLP.

Offset: 0x20068
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_WRITE_ABORT_LOW_REG**: The DMA uses this field to generate bits [31:0] of the address field for the Abort IMWr TLP it generates. Bits [1:0] must be '00' as this address must be dword aligned. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_WRITE_ABORT_IMWR_HIGH_OFF_0

Description: This register holds the higher 32 bits of Abort IMWr TLP.

Offset: 0x2006c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_WRITE_ABORT_HIGH_REG**: The DMA uses this field to generate bits [63:32] of the address field for the Abort IMWr TLP. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_WRITE_CH01_IMWR_DATA_OFF_0

Description: This register holds the Channel 1 and 0 IMWr Done or Abort TLP Data.

Offset: 0x20070
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**WR_CHANNEL_1_DATA**: The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for write channel 1. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
15:0	0x0	**WR_CHANNEL_0_DATA**: The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for write channel 0. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_WRITE_CH23_IMWR_DATA_OFF_0

Description: This register holds the Channel 3 and 2 IMWr Done or Abort TLP Data.

Offset: 0x20074
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**WR_CHANNEL_3_DATA**: The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for write channel 3. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
15:0	0x0	**WR_CHANNEL_2_DATA**: The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for write channel 2. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_WRITE_CH45_IMWR_DATA_OFF_0

Description: This register holds the Channel 5 and 4 IMWr Done or Abort TLP Data.

Offset: 0x20078
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**WR_CHANNEL_5_DATA**: The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for write channel 5. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
15:0	0x0	**WR_CHANNEL_4_DATA**: The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for write channel 4. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_WRITE_CH67_IMWR_DATA_OFF_0

Description: This register holds the Channel 7 and 6 IMWr Done or Abort TLP Data.

Offset: 0x2007c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**WR_CHANNEL_7_DATA**: The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for write channel 7. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
15:0	0x0	**WR_CHANNEL_6_DATA**: The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for write channel 6. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_WRITE_LINKED_LIST_ERR_EN_OFF_0

Description: The LIE and RIE bits in the LL element enable the channel "done" interrupts (local and remote). The LLLAIE and LLRAIE bits enable the channel "abort" interrupts (local and remote).

Offset: 0x20090
Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,xxxx,0000,0000,0000,xxxx,0000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**RSVDP_24**: Reserved for future use.
19:16	RW	0x0	**WR_CHANNEL_LLLAIE**: Write Channel LL Local Abort Interrupt Enable (LLLAIE). You enable the write channel local abort interrupt through this bit. The LIE and RIE bits in the LL element enable the write channel done interrupts. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Used in linked list mode only. For more details, see "Interrupt Handling". Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
15:8	RO	0x0	**RSVDP_8**: Reserved for future use.
3:0	RW	0x0	**WR_CHANNEL_LLRAIE**: Write Channel LL Remote Abort Interrupt Enable (LLRAIE). You enable the write channel remote abort interrupt through this bit. The LIE and RIE bits in the LL element enable the write channel done interrupts. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Used in linked list mode only. For more details, see "Interrupt Handling". Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_READ_INT_STATUS_OFF_0

Description: This register indicates the status of Abort and Done interrupts for the DMA read channels.

Offset: 0x200a0
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**RSVDP_24**: Reserved for future use.

Bit	R/W	Reset	Description
23:16	RW	0x0	**RD_ABORT_INT_STATUS**: Abort Interrupt Status. The DMA read channel has detected an error, or you manually stopped the transfer as described in "Stopping the DMA Transfer (Software Stop)". Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. You can read the DMA Read Error Status Low Register (DMA_READ_ERR_STATUS_LOW_OFF) and DMA Read Error Status High Register (DMA_READ_ERR_STATUS_HIGH_OFF) to determine the source of the error. - Enabling: For more details, see "Interrupts and Error Handling". - Masking: The DMA read interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the DMA read interrupt Clear register to clear this interrupt bit. Note: You can write to this register to emulate interrupt generation, during software or hardware testing. A write to the address triggers an interrupt, but the DMA does not set the Done or Abort bits in this register. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
15:8	RO	0x0	**RSVDP_8**: Reserved for future use.
7:0	RW	0x0	**RD_DONE_INT_STATUS**: Done Interrupt Status. The DMA read channel has successfully completed the DMA read transfer. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For more details, see "Interrupts and Error Handling". - Masking: The DMA read interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the DMA read interrupt Clear register to clear this interrupt bit. Note: You can write to this register to emulate interrupt generation, during software or hardware testing. A write to the address triggers an interrupt, but the DMA does not set the Done or Abort bits in this register. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_READ_INT_MASK_OFF_0

Description: This register holds the Abort and Done DMA read interrupt mask.

Offset: 0x200a8
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00030003 (0b0000,0000,xxxx,xx11,0000,0000,xxxx,xx11)

Bit	R/W	Reset	Description
31:24	RO	0x0	**RSVDP_24**: Reserved for future use.
17:16	RW	0x3	**RD_ABORT_INT_MASK**: Abort Interrupt Mask. Prevents the Abort interrupt status field in the DMA read interrupt status register from asserting the edma_int output. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

Bit	R/W	Reset	Description
15:8	RO	0x0	**RSVDP_8**: Reserved for future use.
1:0	RW	0x3	**RD_DONE_INT_MASK**: Done Interrupt Mask. Prevents the Done interrupt status field in the DMA read interrupt status register from asserting the edma_int output. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_READ_INT_CLEAR_OFF_0

Description: You can write to this register to clear the Abort and Done read interrupts.

Offset: 0x200ac
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**RSVDP_24**: Reserved for future use.
23:16	WO	0x0	**RD_ABORT_INT_CLEAR**: Abort Interrupt Clear. You must write a 1'b1 to clear the corresponding bit in the Abort interrupt status field of the DMA read interrupt status register. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Note: Reading from this self-clearing register field always returns a '0'.
15:8	RO	0x0	**RSVDP_8**: Reserved for future use.
7:0	WO	0x0	**RD_DONE_INT_CLEAR**: Done Interrupt Clear. You must write a 1'b1 to clear the corresponding bit in the Done interrupt status field of the DMA read interrupt status register. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Note: Reading from this self-clearing register field always returns a '0'.

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_READ_ERR_STATUS_LOW_OFF_0

Description: This register provides information pertaining to linked list element fetch error and application write error.

Offset: 0x200b4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**RSVDP_24**: Reserved for future use.
23:16	0x0	**LINK_LIST_ELEMENT_FETCH_ERR_DETECT**: Linked List Element Fetch Error Detected. - The DMA read channel has received an error response from the AXI bus while reading a linked list element from local memory. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For details, see "Interrupts and Error Handling". - Masking: The DMA read interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the Abort interrupt field of the DMA Read Interrupt Clear Register (DMA_READ_INT_CLEAR_OFF) to clear this error bit. Note, this clears all bits in this register, and also the DMA Read Error Status High register (DMA_READ_ERR_STATUS_HIGH_OFF).
15:8	0x0	**RSVDP_8**: Reserved for future use.
7:0	0x0	**APP_WR_ERR_DETECT**: Application Write Error Detected. The DMA read channel has received an error response from the AXI bus (or TRGT1 interface when the AXI Bridge is not used) while writing data to it. This error is fatal. You must restart the transfer from the beginning, as the channel context is corrupted, and the transfer is not rolled back. For more details, see "Linked List Mode". Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For more details, see "Interrupts and Error Handling". - Masking: The DMA read interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the Abort interrupt field of the "DMA Read Interrupt Clear Register" (DMA_READ_INT_CLEAR_OFF) to clear this error bit. Note, this clears all bits in this register, and also the DMA Read Error Status High register (DMA_READ_ERR_STATUS_HIGH_OFF).

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_READ_ERR_STATUS_HIGH_OFF_0

Description: This register provides information pertaining to completion errors.

Offset: 0x200b8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**DATA_POISIONING**: Data Poisoning. The DMA read channel has detected data poisoning in the completion from the remote device (in response to the MRd request). The DMA read channel will drop the completion and then be halted. The CX_FLT_MASK_UR_POIS filter rule does not affect this behavior. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For more details, see "Interrupts and Error Handling". - Masking: The DMA read interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the Abort interrupt field of the DMA Read Interrupt Clear Register (DMA_READ_INT_CLEAR_OFF) to clear this error bit. Note, this also clears the other error bits for the same channel in this register and in the DMA Read Error Status Low register.
23:16	0x0	**CPL_TIMEOUT**: Completion Time Out. The DMA read channel has timed-out while waiting for the remote device to respond to the MRd request, or a malformed CplD has been received. For more details, see "Linked List Mode". Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For details, see "Interrupts and Error Handling" . - Masking: The DMA read interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the Abort interrupt field of the DMA Read Interrupt Clear Register (DMA_READ_INT_CLEAR_OFF) to clear this error bit. Note, this also clears the other error bits for the same channel in this register and in the DMA Read Error Status Low register.
15:8	0x0	**CPL_ABORT**: Completer Abort. The DMA read channel has received a PCIe completer abort completion status from the remote device in response to the MRd request. For more details, see "Linked List Mode". Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For more details, see "Interrupts and Error Handling". - Masking: The DMA read interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the Abort interrupt field of the DMA Read Interrupt Clear Register (DMA_READ_INT_CLEAR_OFF) to clear this error bit. Note, this also clears the other error bits for the same channel in this register and in the DMA Read Error Status Low register.
7:0	0x0	**UNSUPPORTED_REQ**: Unsupported Request. The DMA read channel has received a PCIe unsupported request completion status from the remote device in response to the MRd request. For more details, see "Linked List Mode". Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For more details, see "Interrupts and Error Handling". - Masking: The DMA read interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the Abort interrupt field of the DMA Read Interrupt Clear Register (DMA_READ_INT_CLEAR_OFF) to clear this error bit. Note, this also clears the other error bits for the same channel in this register and in the DMA Read Error Status Low register.

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_READ_LINKED_LIST_ERR_EN_OFF_0

Description: The LIE and RIE bits in the LL element enable the channel "done" interrupts (local and remote). The LLLAIE and LLRAIE bits enable the channel "abort" interrupts (local and remote).

Offset: 0x200c4
Read/Write: See table below
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,xxxx,xx00,0000,0000,xxxx,xx00)

Bit	R/W	Reset	Description
31:24	RO	0x0	**RSVDP_24**: Reserved for future use.
17:16	RW	0x0	**RD_CHANNEL_LLLAIE**: Read Channel LL Local Abort Interrupt Enable (LLLAIE). You enable the read channel Local Abort interrupt through this bit. The LIE and RIE bits in the LL element enable the read channel done interrupts. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Used in linked list mode only. For more details, see "Interrupt Handling". Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
15:8	RO	0x0	**RSVDP_8**: Reserved for future use.
1:0	RW	0x0	**RD_CHANNEL_LLRAIE**: Read Channel LL Remote Abort Interrupt Enable (LLRAIE). You enable the read channel Remote Abort interrupt through this bit. The LIE and RIE bits in the LL element enable the read channel done interrupts. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Used in linked list mode only. For more details, see "Interrupt Handling". Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_READ_DONE_IMWR_LOW_OFF_0

Description: This register holds the lower 32 bits of the Done IMWr TLP address.

Offset: 0x200cc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_READ_DONE_LOW_REG**: The DMA uses this field to generate bits [31:0] of the address field for the Done IMWr TLP. Bits [1:0] must be '00' as this address must be dword aligned. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_READ_DONE_IMWR_HIGH_OFF_0

Description: This register holds the higher 32 bits of the Done IMWr TLP address.

Offset: 0x200d0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_READ_DONE_HIGH_REG**: The DMA uses this field to generate bits [63:32] of the address field for the Done IMWr TLP. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_READ_ABORT_IMWR_LOW_OFF_0

Description: This register holds the lower 32 bits of the Abort IMWr TLP address.

Offset: 0x200d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_READ_ABORT_LOW_REG**: The DMA uses this field to generate bits [31:0] of the address field for the Abort IMWr TLP. Bits [1:0] must be '00' as this address must be dword aligned. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_READ_ABORT_IMWR_HIGH_OFF_0

Description: This register holds the higher 32 bits of the Abort IMWr TLP address.

Offset: 0x200d8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_READ_ABORT_HIGH_REG**: The DMA uses this field to generate bits [63:32] of the address field for the Abort IMWr TLP. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_READ_CH01_IMWR_DATA_OFF_0

Description: This register holds the Channel 1 and 0 IMWr Done or Abort TLP Data.

Offset: 0x200dc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RD_CHANNEL_1_DATA**: The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for read channel 1. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
15:0	0x0	**RD_CHANNEL_0_DATA**: The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for read channel 0. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_READ_CH23_IMWR_DATA_OFF_0

Description: This register holds the Channel 3 and 2 IMWr Done or Abort TLP Data.

Offset: 0x200e0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RD_CHANNEL_3_DATA**: The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for read channel 3. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

Bit	Reset	Description
15:0	0x0	**RD_CHANNEL_2_DATA**: The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for read channel 2. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_READ_CH45_IMWR_DATA_OFF_0

Description: This register holds the Channel 5 and 4 IMWr Done or Abort TLP Data.

Offset: 0x200e4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RD_CHANNEL_5_DATA**: The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for read channel 5. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
15:0	0x0	**RD_CHANNEL_4_DATA**: The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for read channel 4. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_READ_CH67_IMWR_DATA_OFF_0

Description: This register holds the Channel 7 and 6 IMWr Done or Abort TLP Data.

Offset: 0x200e8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RD_CHANNEL_7_DATA**: The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for read channel 7. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

Bit	Reset	Description
15:0	0x0	**RD_CHANNEL_6_DATA**: The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for read channel 6. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_CH_CONTROL1_OFF_WRCH_0_0

Description: This register controls the DMA write channel operation.

Offset: 0x20200
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,00xx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
31:30	0x0	**DMA_AT**: Address Translation TLP Header Bit (AT). The DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
29:27	0x0	**DMA_TC**: Traffic Class TLP Header Bit (TC). The DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
26	0x0	**DMA_RESERVED5**: Reserved. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
25	0x0	**DMA_RO**: Relaxed Ordering TLP Header Bit (RO). DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
24	0x0	**DMA_NS_SRC**: Source No Snoop TLP Header Bit. The DMA uses this TLP header field when generating MRd (SAR addressing space) (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
23	0x0	**DMA_NS_DST**: Destination No Snoop TLP Header Bit. The DMA uses this TLP header field when generating MWr (DAR addressing space) (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
22	0x0	**DMA_MEM_TYPE**: Master AXI ACE-Lite Cache Coherency Control. This field sets the DMA channel memory type of the address space of the data transfer as follows: For more details, see "ACE-Lite Features and Limitations" section of the Databook. Irrespective of the value of this field, Linked list descriptor requests have this attribute set to 0x1. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

Bit	Reset	Description
16:12	0x0	**DMA_FUNC_NUM**: Function Number (FN). The controller uses this field when generating the requester ID for the MRd/MWr DMA TLP. When you have enabled SR-IOV, then this field is ignored if you have set the VFE field in the "DMA Write Channel Control 2 Register" (DMA_CH_CONTROL2_OFF_WRCH_0). Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
11:10	0x0	**DMA_RESERVED1**: Reserved. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
9	0x0	**LLE**: Linked List Enable (LLE). Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
8	0x0	**CCS**: Consumer Cycle State (CCS). Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". You must initialize this bit. The DMA updates this bit during linked list operation. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
7	0x0	**DMA_RESERVED0**: Reserved. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
6:5	0x0	**CS**: Channel Status (CS). The channel status bits identify the current operational state of the DMA channel. The operation state encoding for each DMA channel is a s follows:
4	0x0	**RIE**: Remote Interrupt Enable (RIE). You must set this bit to enable the generation of the Done or Abort Remote interrupts. For more details, see "Interrupts and Error Handling". In LL mode, the DMA overwrites this with the RIE of the LL element. The RIE bit in a LL element only enables the Done interrupt. In non-LL mode, the RIE bit enables the Done and Abort interrupts. This field is not defined in a link LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
3	0x0	**LIE**: Local Interrupt Enable (LIE). You must set this bit to enable the generation of the Done or Abort Local interrupts. For more details, see "Interrupts and Error Handling". In LL mode, the DMA overwrites this with the LIE of the LL element. The LIE bit in a LL element only enables the Done interrupt. In non-LL mode, the LIE bit enables the Done and Abort interrupts. This field is not defined in a link LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
2	0x0	**LLP**: Load Link Pointer (LLP). Used in linked list mode only. Indicates that this linked list element is a link element, and its LL element pointer dwords are pointing to the next (non-contiguous) element. The DMA loads this field with the LLP of the linked list element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

Bit	Reset	Description
1	0x0	**TCB**: Toggle Cycle Bit (TCB). Indicates to the DMA to toggle its interpretation of the CB. Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". The DMA loads this field with the TCB of the linked list element. this field is not defined in a data LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
0	0x0	**CB**: Cycle Bit (CB). Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". The DMA loads this field with the CB of the linked list element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_TRANSFER_SIZE_OFF_WRCH_0_0

Description: This register holds the DMA write transfer size.

Offset: 0x20208
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_TRANSFER_SIZE**: DMA Transfer Size. You program this register with the size of the DMA transfer. The maximum DMA transfer size is 4Gbytes. The minimum transfer size is one byte (0x1). This field is automatically decremented by the DMA as the DMA write channel transfer progresses. This field indicates the number bytes remaining to be transferred. When all bytes are successfully transferred the current transfer size is zero. In LL mode, the DMA overwrites this register with the corresponding dword of the LL element. You can read this register to monitor the transfer progress, however in some scenarios this register is updated after a delay. For example, when less than 3 channels are doorbelled, this register is updated only after a descriptor finishes(linked list mode), or the transfer ends (non-linked list mode). Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_SAR_LOW_OFF_WRCH_0_0

Description: This register holds the lower 32 bits of the Source Address Register (SAR).

Offset: 0x2020c
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_ADDR_REG_LOW**: Source Address Register (lower 32 bits). Indicates the next address to be read from. The DMA increments the SAR as the DMA transfer progresses. In LL mode, the DMA overwrites this with the corresponding dword of the LL element. - DMA Read: The SAR is the address of the remote memory. - DMA Write: The SAR is the address of the local memory. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_SAR_HIGH_OFF_WRCH_0_0

Description: This register holds the higher 32 bits of the Source Address Register (SAR).

Offset: 0x20210
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_ADDR_REG_HIGH**: Source Address Register (higher 32 bits). In LL mode, the DMA overwrites this with the corresponding dword of the LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_DAR_LOW_OFF_WRCH_0_0

Description: This register holds the lower 32 bits of the Destination Address Register (DAR).

Offset: 0x20214
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_ADDR_REG_LOW**: Destination Address Register (lower 32 bits). Indicates the next address to be written to. The DMA increments the DAR as the DMA transfer progresses. In LL mode, the DMA overwrites this with the corresponding dword of the LL element. - DMA Read: The DAR is the address of the local memory. - DMA Write: The DAR is the address of the remote memory. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_DAR_HIGH_OFF_WRCH_0_0

Description: This register holds the higher 32 bits of the Destination Address Register (DAR).

Offset: 0x20218
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_ADDR_REG_HIGH**: Destination Address Register (Higher 32 bits). In LL mode, the DMA overwrites this with the corresponding dword of the LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_LLP_LOW_OFF_WRCH_0_0

Description: This register holds the lower 32 bits of the DMA write linked list pointer.

Offset: 0x2021c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LLP_LOW**: Lower bits of the address of the linked list transfer list in local memory. Used in linked list mode only. Updated by the DMA to point to the next element in the transfer list after the previous element is consumed. - When the current element is a data element; this field is incremented by 6 DWORDS. - When the current element is a link element; this field is overwritten by the LL Element Pointer of the element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_LLP_HIGH_OFF_WRCH_0_0

Description: This register holds the higher 32 bits of the DMA write linked list pointer.

Offset: 0x20220
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LLP_HIGH**: Higher 32 bits of the address of the linked list transfer list in local memory. Used in linked list mode only. Updated by the DMA to point to the next element in the transfer list as elements are consumed. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_CH_CONTROL1_OFF_RDCH_0_0

Description: This register controls the DMA read channel operation.

Offset: 0x20300
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,00xx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
31:30	0x0	**DMA_AT**: Address Translation TLP Header Bit (AT). The DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
29:27	0x0	**DMA_TC**: Traffic Class TLP Header Bit (TC). The DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
26	0x0	**DMA_RESERVED5**: Reserved. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
25	0x0	**DMA_RO**: Relaxed Ordering TLP Header Bit (RO) The DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

Bit	Reset	Description
24	0x0	**DMA_NS_SRC**: Source No Snoop TLP Header Bit. The DMA uses this TLP header field when generating MRd (SAR addressing space) (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
23	0x0	**DMA_NS_DST**: Destination No Snoop TLP Header Bit. The DMA uses this TLP header field when generating MWr (DAR addressing space) (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
22	0x0	**DMA_MEM_TYPE**: Master AXI ACE-Lite Cache Coherency Control. This field sets the DMA channel memory type of the address space of the data transfer as follows: For more details, see "ACE-Lite Features and Limitations" section of the Databook. Irrespective of the value of this field, Linked list descriptor requests have this attribute set to 0x1. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
16:12	0x0	**DMA_FUNC_NUM**: Function Number (FN). The controller uses this field when generating the requester ID for the MRd/MWr DMA TLP. When you have enabled SR-IOV, then this field is ignored if you have set the VFE field in the DMA Read Channel Control 2 Register (DMA_CH_CONTROL2_OFF_RDCH_0). Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
11:10	0x0	**DMA_RESERVED1**: Reserved. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
9	0x0	**LLE**: Linked List Enable (LLE). Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
8	0x0	**CCS**: Consumer Cycle State (CCS). Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". You must initialize this bit. The DMA updates this bit during linked list operation. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
7	0x0	**DMA_RESERVED0**: Reserved. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
6:5	0x0	**CS**: Channel Status (CS). The channel status bits identify the current operational state of the DMA channel. The operation state encoding for each DMA channel is a s follows:
4	0x0	**RIE**: Remote Interrupt Enable (RIE). You must set this bit to enable the generation of the Done or Abort Remote interrupts. For more details, see "Interrupts and Error Handling". In LL mode, the DMA overwrites this with the RIE of the LL element. The RIE bit in a LL element only enables the Done interrupt. In non-LL mode, the RIE bit enables the Done and Abort interrupts. This field is not defined in a link LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

Bit	Reset	Description
3	0x0	**LIE**: Local Interrupt Enable (LIE). You must set this bit to enable the generation of the Done or Abort Local interrupts. For more details, see "Interrupts and Error Handling". In LL mode, the DMA overwrites this with the LIE of the LL element. The LIE bit in a LL element only enables the Done interrupt. In non-LL mode, the LIE bit enables the Done and Abort interrupts. This field is not defined in a link LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
2	0x0	**LLP**: Load Link Pointer (LLP). Used in linked list mode only. Indicates that this linked list element is a link element, and its LL element pointer dwords are pointing to the next (non-contiguous) element. The DMA loads this field with the LLP of the linked list element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
1	0x0	**TCB**: Toggle Cycle Bit (TCB). Indicates to the DMA to toggle its interpretation of the CB. Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". The DMA loads this field with the TCB of the linked list element. This field is not defined in a data LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
0	0x0	**CB**: Cycle Bit (CB). Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". The DMA loads this field with the CB of the linked list element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_TRANSFER_SIZE_OFF_RDCH_0_0

Description: This register holds the DMA read transfer size.

Offset: 0x20308
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_TRANSFER_SIZE**: DMA Transfer Size. You program this register with the size of the DMA transfer. The maximum DMA transfer size is 4Gbytes. The minimum transfer size is one byte (0x1). This field is automatically decremented by the DMA as the DMA read channel transfer progresses. This field indicates the number bytes remaining to be transferred. When all bytes are successfully transferred the current transfer size is zero. In LL mode, the DMA overwrites this register with the corresponding dword of the LL element. You can read this register to monitor the transfer progress, however in some scenarios this register is updated after a delay. For example, when less than 3 channels are doorbelled, this register is updated only after a descriptor finishes(linked list mode), or the transfer ends (non-linked list mode). Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_SAR_LOW_OFF_RDCH_0_0

Description: This register holds the lower 32 bits of the Source Address Register (SAR).

Offset: 0x2030c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_ADDR_REG_LOW**: Source Address Register (Lower 32 bits). Indicates the next address to be read from. The DMA increments the SAR as the DMA transfer progresses. In LL mode, the DMA overwrites this with the corresponding dword of the LL element. - DMA Read: The SAR is the address of the remote memory. - DMA Read: The SAR is the address of the local memory. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_SAR_HIGH_OFF_RDCH_0_0

Description: This register holds the higher 32 bits of the Source Address Register (SAR).

Offset: 0x20310
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_ADDR_REG_HIGH**: Source Address Register (higher 32 bits). In LL mode, the DMA overwrites this with the corresponding dword of the LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_DAR_LOW_OFF_RDCH_0_0

Description: This register holds the lower 32 bits of the Destination Address Register (DAR).

Offset: 0x20314
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_ADDR_REG_LOW**: Destination Address Register (Lower 32 bits). Indicates the next address to be written to. The DMA increments the DAR as the DMA transfer progresses. In LL mode, the DMA overwrites this with the corresponding dword of the LL element. - DMA Read: The DAR is the address of the local memory. - DMA Read: The DAR is the address of the remote memory. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_DAR_HIGH_OFF_RDCH_0_0

Description: This register holds the higher 32 bits of the Destination Address Register (DAR).

Offset: 0x20318
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_ADDR_REG_HIGH**: Destination Address Register (higher 32 bits). In LL mode, the DMA overwrites this with the corresponding dword of the LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_LLP_LOW_OFF_RDCH_0_0

Description: This register holds the lower 32 bits of the DMA read linked list pointer.

Offset: 0x2031c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LLP_LOW**: Lower bits of the address of the linked list transfer list in local memory. Used in linked list mode only. Updated by the DMA to point to the next element in the transfer list after the previous element is consumed. - When the current element is a data element; this field is incremented by 6 DWORDS. - When the current element is a link element; this field is overwritten by the LL Element Pointer of the element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_LLP_HIGH_OFF_RDCH_0_0

Description: This register holds the higher 32 bits of the DMA read linked list pointer.

Offset: 0x20320
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LLP_HIGH**: Higher 32 bits of the address of the linked list transfer list in local memory. Used in linked list mode only. Updated by the DMA to point to the next element in the transfer list as elements are consumed. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_CH_CONTROL1_OFF_WRCH_1_0

Description: This register controls the DMA write channel operation.

Offset: 0x20400
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,00xx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
31:30	0x0	**DMA_AT**: Address Translation TLP Header Bit (AT). The DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
29:27	0x0	**DMA_TC**: Traffic Class TLP Header Bit (TC). The DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
26	0x0	**DMA_RESERVED5**: Reserved. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
25	0x0	**DMA_RO**: Relaxed Ordering TLP Header Bit (RO). DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
24	0x0	**DMA_NS_SRC**: Source No Snoop TLP Header Bit. The DMA uses this TLP header field when generating MRd (SAR addressing space) (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
23	0x0	**DMA_NS_DST**: Destination No Snoop TLP Header Bit. The DMA uses this TLP header field when generating MWr (DAR addressing space) (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
22	0x0	**DMA_MEM_TYPE**: Master AXI ACE-Lite Cache Coherency Control. This field sets the DMA channel memory type of the address space of the data transfer as follows: For more details, see "ACE-Lite Features and Limitations" section of the Databook. Irrespective of the value of this field, Linked list descriptor requests have this attribute set to 0x1. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
16:12	0x0	**DMA_FUNC_NUM**: Function Number (FN). The controller uses this field when generating the requester ID for the MRd/MWr DMA TLP. When you have enabled SR-IOV, then this field is ignored if you have set the VFE field in the "DMA Write Channel Control 2 Register" (DMA_CH_CONTROL2_OFF_WRCH_0). Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
11:10	0x0	**DMA_RESERVED1**: Reserved. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
9	0x0	**LLE**: Linked List Enable (LLE). Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

Bit	Reset	Description
8	0x0	**CCS**: Consumer Cycle State (CCS). Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". You must initialize this bit. The DMA updates this bit during linked list operation. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
7	0x0	**DMA_RESERVED0**: Reserved. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
6:5	0x0	**CS**: Channel Status (CS). The channel status bits identify the current operational state of the DMA channel. The operation state encoding for each DMA channel is a s follows:
4	0x0	**RIE**: Remote Interrupt Enable (RIE). You must set this bit to enable the generation of the Done or Abort Remote interrupts. For more details, see "Interrupts and Error Handling". In LL mode, the DMA overwrites this with the RIE of the LL element. The RIE bit in a LL element only enables the Done interrupt. In non-LL mode, the RIE bit enables the Done and Abort interrupts. This field is not defined in a link LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
3	0x0	**LIE**: Local Interrupt Enable (LIE). You must set this bit to enable the generation of the Done or Abort Local interrupts. For more details, see "Interrupts and Error Handling". In LL mode, the DMA overwrites this with the LIE of the LL element. The LIE bit in a LL element only enables the Done interrupt. In non-LL mode, the LIE bit enables the Done and Abort interrupts. This field is not defined in a link LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
2	0x0	**LLP**: Load Link Pointer (LLP). Used in linked list mode only. Indicates that this linked list element is a link element, and its LL element pointer dwords are pointing to the next (non-contiguous) element. The DMA loads this field with the LLP of the linked list element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
1	0x0	**TCB**: Toggle Cycle Bit (TCB). Indicates to the DMA to toggle its interpretation of the CB. Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". The DMA loads this field with the TCB of the linked list element. this field is not defined in a data LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
0	0x0	**CB**: Cycle Bit (CB). Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". The DMA loads this field with the CB of the linked list element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_TRANSFER_SIZE_OFF_WRCH_1_0

Description: This register holds the DMA write transfer size.

Offset: 0x20408
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_TRANSFER_SIZE**: DMA Transfer Size. You program this register with the size of the DMA transfer. The maximum DMA transfer size is 4Gbytes. The minimum transfer size is one byte (0x1). This field is automatically decremented by the DMA as the DMA write channel transfer progresses. This field indicates the number bytes remaining to be transferred. When all bytes are successfully transferred the current transfer size is zero. In LL mode, the DMA overwrites this register with the corresponding dword of the LL element. You can read this register to monitor the transfer progress, however in some scenarios this register is updated after a delay. For example, when less than 3 channels are doorbelled, this register is updated only after a descriptor finishes(linked list mode), or the transfer ends (non-linked list mode). Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_SAR_LOW_OFF_WRCH_1_0

Description: This register holds the lower 32 bits of the Source Address Register (SAR).

Offset: 0x2040c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_ADDR_REG_LOW**: Source Address Register (lower 32 bits). Indicates the next address to be read from. The DMA increments the SAR as the DMA transfer progresses. In LL mode, the DMA overwrites this with the corresponding dword of the LL element. - DMA Read: The SAR is the address of the remote memory. - DMA Write: The SAR is the address of the local memory. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_SAR_HIGH_OFF_WRCH_1_0

Description: This register holds the higher 32 bits of the Source Address Register (SAR).

Offset: 0x20410
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_ADDR_REG_HIGH**: Source Address Register (higher 32 bits). In LL mode, the DMA overwrites this with the corresponding dword of the LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_DAR_LOW_OFF_WRCH_1_0

Description: This register holds the lower 32 bits of the Destination Address Register (DAR).

Offset: 0x20414
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_ADDR_REG_LOW**: Destination Address Register (lower 32 bits). Indicates the next address to be written to. The DMA increments the DAR as the DMA transfer progresses. In LL mode, the DMA overwrites this with the corresponding dword of the LL element. - DMA Read: The DAR is the address of the local memory. - DMA Write: The DAR is the address of the remote memory. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_DAR_HIGH_OFF_WRCH_1_0

Description: This register holds the higher 32 bits of the Destination Address Register (DAR).

Offset: 0x20418
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_ADDR_REG_HIGH**: Destination Address Register (Higher 32 bits). In LL mode, the DMA overwrites this with the corresponding dword of the LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_LLP_LOW_OFF_WRCH_1_0

Description: This register holds the lower 32 bits of the DMA write linked list pointer.

Offset: 0x2041c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LLP_LOW**: Lower bits of the address of the linked list transfer list in local memory. Used in linked list mode only. Updated by the DMA to point to the next element in the transfer list after the previous element is consumed. - When the current element is a data element; this field is incremented by 6 DWORDS. - When the current element is a link element; this field is overwritten by the LL Element Pointer of the element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_LLP_HIGH_OFF_WRCH_1_0

Description: This register holds the higher 32 bits of the DMA write linked list pointer.

Offset: 0x20420
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LLP_HIGH**: Higher 32 bits of the address of the linked list transfer list in local memory. Used in linked list mode only. Updated by the DMA to point to the next element in the transfer list as elements are consumed. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_CH_CONTROL1_OFF_RDCH_1_0

Description: This register controls the DMA read channel operation.

Offset: 0x20500
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,00xx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
31:30	0x0	**DMA_AT**: Address Translation TLP Header Bit (AT). The DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
29:27	0x0	**DMA_TC**: Traffic Class TLP Header Bit (TC). The DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
26	0x0	**DMA_RESERVED5**: Reserved. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
25	0x0	**DMA_RO**: Relaxed Ordering TLP Header Bit (RO) The DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
24	0x0	**DMA_NS_SRC**: Source No Snoop TLP Header Bit. The DMA uses this TLP header field when generating MRd (SAR addressing space) (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
23	0x0	**DMA_NS_DST**: Destination No Snoop TLP Header Bit. The DMA uses this TLP header field when generating MWr (DAR addressing space) (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
22	0x0	**DMA_MEM_TYPE**: Master AXI ACE-Lite Cache Coherency Control. This field sets the DMA channel memory type of the address space of the data transfer as follows: For more details, see "ACE-Lite Features and Limitations" section of the Databook. Irrespective of the value of this field, Linked list descriptor requests have this attribute set to 0x1. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
16:12	0x0	**DMA_FUNC_NUM**: Function Number (FN). The controller uses this field when generating the requester ID for the MRd/MWr DMA TLP. When you have enabled SR-IOV, then this field is ignored if you have set the VFE field in the DMA Read Channel Control 2 Register (DMA_CH_CONTROL2_OFF_RDCH_0). Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

Bit	Reset	Description
11:10	0x0	**DMA_RESERVED1**: Reserved. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
9	0x0	**LLE**: Linked List Enable (LLE). Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
8	0x0	**CCS**: Consumer Cycle State (CCS). Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". You must initialize this bit. The DMA updates this bit during linked list operation. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
7	0x0	**DMA_RESERVED0**: Reserved. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
6:5	0x0	**CS**: Channel Status (CS). The channel status bits identify the current operational state of the DMA channel. The operation state encoding for each DMA channel is a s follows:
4	0x0	**RIE**: Remote Interrupt Enable (RIE). You must set this bit to enable the generation of the Done or Abort Remote interrupts. For more details, see "Interrupts and Error Handling". In LL mode, the DMA overwrites this with the RIE of the LL element. The RIE bit in a LL element only enables the Done interrupt. In non-LL mode, the RIE bit enables the Done and Abort interrupts. This field is not defined in a link LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
3	0x0	**LIE**: Local Interrupt Enable (LIE). You must set this bit to enable the generation of the Done or Abort Local interrupts. For more details, see "Interrupts and Error Handling". In LL mode, the DMA overwrites this with the LIE of the LL element. The LIE bit in a LL element only enables the Done interrupt. In non-LL mode, the LIE bit enables the Done and Abort interrupts. This field is not defined in a link LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
2	0x0	**LLP**: Load Link Pointer (LLP). Used in linked list mode only. Indicates that this linked list element is a link element, and its LL element pointer dwords are pointing to the next (non-contiguous) element. The DMA loads this field with the LLP of the linked list element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
1	0x0	**TCB**: Toggle Cycle Bit (TCB). Indicates to the DMA to toggle its interpretation of the CB. Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". The DMA loads this field with the TCB of the linked list element. This field is not defined in a data LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

Bit	Reset	Description
0	0x0	**CB**: Cycle Bit (CB). Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". The DMA loads this field with the CB of the linked list element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_TRANSFER_SIZE_OFF_RDCH_1_0

Description: This register holds the DMA read transfer size.

Offset: 0x20508
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_TRANSFER_SIZE**: DMA Transfer Size. You program this register with the size of the DMA transfer. The maximum DMA transfer size is 4Gbytes. The minimum transfer size is one byte (0x1). This field is automatically decremented by the DMA as the DMA read channel transfer progresses. This field indicates the number bytes remaining to be transferred. When all bytes are successfully transferred the current transfer size is zero. In LL mode, the DMA overwrites this register with the corresponding dword of the LL element. You can read this register to monitor the transfer progress, however in some scenarios this register is updated after a delay. For example, when less than 3 channels are doorbelled, this register is updated only after a descriptor finishes(linked list mode), or the transfer ends (non-linked list mode). Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_SAR_LOW_OFF_RDCH_1_0

Description: This register holds the lower 32 bits of the Source Address Register (SAR).

Offset: 0x2050c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_ADDR_REG_LOW**: Source Address Register (Lower 32 bits). Indicates the next address to be read from. The DMA increments the SAR as the DMA transfer progresses. In LL mode, the DMA overwrites this with the corresponding dword of the LL element. - DMA Read: The SAR is the address of the remote memory. - DMA Read: The SAR is the address of the local memory. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_SAR_HIGH_OFF_RDCH_1_0

Description: This register holds the higher 32 bits of the Source Address Register (SAR).

Offset: 0x20510
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_ADDR_REG_HIGH**: Source Address Register (higher 32 bits). In LL mode, the DMA overwrites this with the corresponding dword of the LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_DAR_LOW_OFF_RDCH_1_0

Description: This register holds the lower 32 bits of the Destination Address Register (DAR).

Offset: 0x20514
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_ADDR_REG_LOW**: Destination Address Register (Lower 32 bits). Indicates the next address to be written to. The DMA increments the DAR as the DMA transfer progresses. In LL mode, the DMA overwrites this with the corresponding dword of the LL element. - DMA Read: The DAR is the address of the local memory. - DMA Read: The DAR is the address of the remote memory. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_DAR_HIGH_OFF_RDCH_1_0

Description: This register holds the higher 32 bits of the Destination Address Register (DAR).

Offset: 0x20518
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_ADDR_REG_HIGH**: Destination Address Register (higher 32 bits). In LL mode, the DMA overwrites this with the corresponding dword of the LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_LLP_LOW_OFF_RDCH_1_0

Description: This register holds the lower 32 bits of the DMA read linked list pointer.

Offset: 0x2051c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LLP_LOW**: Lower bits of the address of the linked list transfer list in local memory. Used in linked list mode only. Updated by the DMA to point to the next element in the transfer list after the previous element is consumed. - When the current element is a data element; this field is incremented by 6 DWORDS. - When the current element is a link element; this field is overwritten by the LL Element Pointer of the element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_LLP_HIGH_OFF_RDCH_1_0

Description: This register holds the higher 32 bits of the DMA read linked list pointer.

Offset: 0x20520
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LLP_HIGH**: Higher 32 bits of the address of the linked list transfer list in local memory. Used in linked list mode only. Updated by the DMA to point to the next element in the transfer list as elements are consumed. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_CH_CONTROL1_OFF_WRCH_2_0

Description: This register controls the DMA write channel operation.

Offset: 0x20600
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,00xx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
31:30	0x0	**DMA_AT**: Address Translation TLP Header Bit (AT). The DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
29:27	0x0	**DMA_TC**: Traffic Class TLP Header Bit (TC). The DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
26	0x0	**DMA_RESERVED5**: Reserved. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
25	0x0	**DMA_RO**: Relaxed Ordering TLP Header Bit (RO). DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
24	0x0	**DMA_NS_SRC**: Source No Snoop TLP Header Bit. The DMA uses this TLP header field when generating MRd (SAR addressing space) (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

Bit	Reset	Description
23	0x0	**DMA_NS_DST**: Destination No Snoop TLP Header Bit. The DMA uses this TLP header field when generating MWr (DAR addressing space) (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
22	0x0	**DMA_MEM_TYPE**: Master AXI ACE-Lite Cache Coherency Control. This field sets the DMA channel memory type of the address space of the data transfer as follows: For more details, see "ACE-Lite Features and Limitations" section of the Databook. Irrespective of the value of this field, Linked list descriptor requests have this attribute set to 0x1. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
16:12	0x0	**DMA_FUNC_NUM**: Function Number (FN). The controller uses this field when generating the requester ID for the MRd/MWr DMA TLP. When you have enabled SR-IOV, then this field is ignored if you have set the VFE field in the "DMA Write Channel Control 2 Register" (DMA_CH_CONTROL2_OFF_WRCH_0). Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
11:10	0x0	**DMA_RESERVED1**: Reserved. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
9	0x0	**LLE**: Linked List Enable (LLE). Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
8	0x0	**CCS**: Consumer Cycle State (CCS). Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". You must initialize this bit. The DMA updates this bit during linked list operation. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
7	0x0	**DMA_RESERVED0**: Reserved. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
6:5	0x0	**CS**: Channel Status (CS). The channel status bits identify the current operational state of the DMA channel. The operation state encoding for each DMA channel is a s follows:
4	0x0	**RIE**: Remote Interrupt Enable (RIE). You must set this bit to enable the generation of the Done or Abort Remote interrupts. For more details, see "Interrupts and Error Handling". In LL mode, the DMA overwrites this with the RIE of the LL element. The RIE bit in a LL element only enables the Done interrupt. In non-LL mode, the RIE bit enables the Done and Abort interrupts. This field is not defined in a link LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
3	0x0	**LIE**: Local Interrupt Enable (LIE). You must set this bit to enable the generation of the Done or Abort Local interrupts. For more details, see "Interrupts and Error Handling". In LL mode, the DMA overwrites this with the LIE of the LL element. The LIE bit in a LL element only enables the Done interrupt. In non-LL mode, the LIE bit enables the Done and Abort interrupts. This field is not defined in a link LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

Bit	Reset	Description
2	0x0	**LLP**: Load Link Pointer (LLP). Used in linked list mode only. Indicates that this linked list element is a link element, and its LL element pointer dwords are pointing to the next (non-contiguous) element. The DMA loads this field with the LLP of the linked list element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
1	0x0	**TCB**: Toggle Cycle Bit (TCB). Indicates to the DMA to toggle its interpretation of the CB. Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". The DMA loads this field with the TCB of the linked list element. this field is not defined in a data LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
0	0x0	**CB**: Cycle Bit (CB). Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". The DMA loads this field with the CB of the linked list element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_TRANSFER_SIZE_OFF_WRCH_2_0

Description: This register holds the DMA write transfer size.

Offset: 0x20608
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_TRANSFER_SIZE**: DMA Transfer Size. You program this register with the size of the DMA transfer. The maximum DMA transfer size is 4Gbytes. The minimum transfer size is one byte (0x1). This field is automatically decremented by the DMA as the DMA write channel transfer progresses. This field indicates the number bytes remaining to be transferred. When all bytes are successfully transferred the current transfer size is zero. In LL mode, the DMA overwrites this register with the corresponding dword of the LL element. You can read this register to monitor the transfer progress, however in some scenarios this register is updated after a delay. For example, when less than 3 channels are doorbelled, this register is updated only after a descriptor finishes(linked list mode), or the transfer ends (non-linked list mode). Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_SAR_LOW_OFF_WRCH_2_0

Description: This register holds the lower 32 bits of the Source Address Register (SAR).

Offset: 0x2060c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_ADDR_REG_LOW**: Source Address Register (lower 32 bits). Indicates the next address to be read from. The DMA increments the SAR as the DMA transfer progresses. In LL mode, the DMA overwrites this with the corresponding dword of the LL element. - DMA Read: The SAR is the address of the remote memory. - DMA Write: The SAR is the address of the local memory. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_SAR_HIGH_OFF_WRCH_2_0

Description: This register holds the higher 32 bits of the Source Address Register (SAR).

Offset: 0x20610
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_ADDR_REG_HIGH**: Source Address Register (higher 32 bits). In LL mode, the DMA overwrites this with the corresponding dword of the LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_DAR_LOW_OFF_WRCH_2_0

Description: This register holds the lower 32 bits of the Destination Address Register (DAR).

Offset: 0x20614
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_ADDR_REG_LOW**: Destination Address Register (lower 32 bits). Indicates the next address to be written to. The DMA increments the DAR as the DMA transfer progresses. In LL mode, the DMA overwrites this with the corresponding dword of the LL element. - DMA Read: The DAR is the address of the local memory. - DMA Write: The DAR is the address of the remote memory. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_DAR_HIGH_OFF_WRCH_2_0

Description: This register holds the higher 32 bits of the Destination Address Register (DAR).

Offset: 0x20618
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_ADDR_REG_HIGH**: Destination Address Register (Higher 32 bits). In LL mode, the DMA overwrites this with the corresponding dword of the LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_LLP_LOW_OFF_WRCH_2_0

Description: This register holds the lower 32 bits of the DMA write linked list pointer.

Offset: 0x2061c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LLP_LOW**: Lower bits of the address of the linked list transfer list in local memory. Used in linked list mode only. Updated by the DMA to point to the next element in the transfer list after the previous element is consumed. - When the current element is a data element; this field is incremented by 6 DWORDS. - When the current element is a link element; this field is overwritten by the LL Element Pointer of the element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_LLP_HIGH_OFF_WRCH_2_0

Description: This register holds the higher 32 bits of the DMA write linked list pointer.

Offset: 0x20620
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LLP_HIGH**: Higher 32 bits of the address of the linked list transfer list in local memory. Used in linked list mode only. Updated by the DMA to point to the next element in the transfer list as elements are consumed. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_CH_CONTROL1_OFF_WRCH_3_0

Description: This register controls the DMA write channel operation.

Offset: 0x20800
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,00xx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
31:30	0x0	**DMA_AT**: Address Translation TLP Header Bit (AT). The DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

Bit	Reset	Description
29:27	0x0	**DMA_TC**: Traffic Class TLP Header Bit (TC). The DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
26	0x0	**DMA_RESERVED5**: Reserved. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
25	0x0	**DMA_RO**: Relaxed Ordering TLP Header Bit (RO). DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
24	0x0	**DMA_NS_SRC**: Source No Snoop TLP Header Bit. The DMA uses this TLP header field when generating MRd (SAR addressing space) (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
23	0x0	**DMA_NS_DST**: Destination No Snoop TLP Header Bit. The DMA uses this TLP header field when generating MWr (DAR addressing space) (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
22	0x0	**DMA_MEM_TYPE**: Master AXI ACE-Lite Cache Coherency Control. This field sets the DMA channel memory type of the address space of the data transfer as follows: For more details, see "ACE-Lite Features and Limitations" section of the Databook. Irrespective of the value of this field, Linked list descriptor requests have this attribute set to 0x1. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
16:12	0x0	**DMA_FUNC_NUM**: Function Number (FN). The controller uses this field when generating the requester ID for the MRd/MWr DMA TLP. When you have enabled SR-IOV, then this field is ignored if you have set the VFE field in the "DMA Write Channel Control 2 Register" (DMA_CH_CONTROL2_OFF_WRCH_0). Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
11:10	0x0	**DMA_RESERVED1**: Reserved. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
9	0x0	**LLE**: Linked List Enable (LLE). Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
8	0x0	**CCS**: Consumer Cycle State (CCS). Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". You must initialize this bit. The DMA updates this bit during linked list operation. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
7	0x0	**DMA_RESERVED0**: Reserved. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

Bit	Reset	Description
6:5	0x0	**CS**: Channel Status (CS). The channel status bits identify the current operational state of the DMA channel. The operation state encoding for each DMA channel is a s follows:
4	0x0	**RIE**: Remote Interrupt Enable (RIE). You must set this bit to enable the generation of the Done or Abort Remote interrupts. For more details, see "Interrupts and Error Handling". In LL mode, the DMA overwrites this with the RIE of the LL element. The RIE bit in a LL element only enables the Done interrupt. In non-LL mode, the RIE bit enables the Done and Abort interrupts. This field is not defined in a link LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
3	0x0	**LIE**: Local Interrupt Enable (LIE). You must set this bit to enable the generation of the Done or Abort Local interrupts. For more details, see "Interrupts and Error Handling". In LL mode, the DMA overwrites this with the LIE of the LL element. The LIE bit in a LL element only enables the Done interrupt. In non-LL mode, the LIE bit enables the Done and Abort interrupts. This field is not defined in a link LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
2	0x0	**LLP**: Load Link Pointer (LLP). Used in linked list mode only. Indicates that this linked list element is a link element, and its LL element pointer dwords are pointing to the next (non-contiguous) element. The DMA loads this field with the LLP of the linked list element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
1	0x0	**TCB**: Toggle Cycle Bit (TCB). Indicates to the DMA to toggle its interpretation of the CB. Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". The DMA loads this field with the TCB of the linked list element. this field is not defined in a data LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
0	0x0	**CB**: Cycle Bit (CB). Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". The DMA loads this field with the CB of the linked list element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_TRANSFER_SIZE_OFF_WRCH_3_0

Description: This register holds the DMA write transfer size.

Offset: 0x20808
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DMA_TRANSFER_SIZE**: DMA Transfer Size. You program this register with the size of the DMA transfer. The maximum DMA transfer size is 4Gbytes. The minimum transfer size is one byte (0x1). This field is automatically decremented by the DMA as the DMA write channel transfer progresses. This field indicates the number bytes remaining to be transferred. When all bytes are successfully transferred the current transfer size is zero. In LL mode, the DMA overwrites this register with the corresponding dword of the LL element. You can read this register to monitor the transfer progress, however in some scenarios this register is updated after a delay. For example, when less than 3 channels are doorbelled, this register is updated only after a descriptor finishes(linked list mode), or the transfer ends (non-linked list mode). Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_SAR_LOW_OFF_WRCH_3_0

Description: This register holds the lower 32 bits of the Source Address Register (SAR).

Offset: 0x2080c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_ADDR_REG_LOW**: Source Address Register (lower 32 bits). Indicates the next address to be read from. The DMA increments the SAR as the DMA transfer progresses. In LL mode, the DMA overwrites this with the corresponding dword of the LL element. - DMA Read: The SAR is the address of the remote memory. - DMA Write: The SAR is the address of the local memory. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_SAR_HIGH_OFF_WRCH_3_0

Description: This register holds the higher 32 bits of the Source Address Register (SAR).

Offset: 0x20810
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SRC_ADDR_REG_HIGH**: Source Address Register (higher 32 bits). In LL mode, the DMA overwrites this with the corresponding dword of the LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_DAR_LOW_OFF_WRCH_3_0

Description: This register holds the lower 32 bits of the Destination Address Register (DAR).

Offset: 0x20814
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_ADDR_REG_LOW**: Destination Address Register (lower 32 bits). Indicates the next address to be written to. The DMA increments the DAR as the DMA transfer progresses. In LL mode, the DMA overwrites this with the corresponding dword of the LL element. - DMA Read: The DAR is the address of the local memory. - DMA Write: The DAR is the address of the remote memory. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_DAR_HIGH_OFF_WRCH_3_0

Description: This register holds the higher 32 bits of the Destination Address Register (DAR).

Offset: 0x20818
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DST_ADDR_REG_HIGH**: Destination Address Register (Higher 32 bits). In LL mode, the DMA overwrites this with the corresponding dword of the LL element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_LLP_LOW_OFF_WRCH_3_0

Description: This register holds the lower 32 bits of the DMA write linked list pointer.

Offset: 0x2081c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LLP_LOW**: Lower bits of the address of the linked list transfer list in local memory. Used in linked list mode only. Updated by the DMA to point to the next element in the transfer list after the previous element is consumed. - When the current element is a data element; this field is incremented by 6 DWORDS. - When the current element is a link element; this field is overwritten by the LL Element Pointer of the element. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_DMA_CAP_PF0_DMA_CAP_DMA_LLP_HIGH_OFF_WRCH_3_0

Description: This register holds the higher 32 bits of the DMA write linked list pointer.

Offset: 0x20820
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LLP_HIGH**: Higher 32 bits of the address of the linked list transfer list in local memory. Used in linked list mode only. Updated by the DMA to point to the next element in the transfer list as elements are consumed. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

9.3.5.2 PCIe Root Port

NOTE:

There are 11 instances for each of the registers as listed, one for each of the PCIe module instances, namely C0, C1, C2, C3, C4, C5, C6, C7, C8, C9, and C10.
For the base addresses of these different register instances, please refer to the System Address Map.

NOTE:

The binary “x” is turned to “0” for the Reset and PROD hex value. Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as “x” or “Undefined” actually means that the bit could be either “0” or “1” after Reset, whereas a bit PROD value as “x” or “Don’t-care” means that either “0” or “1” can be written to the bit by software Initialization.

PCIE_RP_APPL_PINMUX_0

Offset: 0x0
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00003880 (0bxxxx,xxxx,xxxx,xxxx,xx11,1000,1x00,0000)

Bit	R/W	Reset	Description
10	RW	0x0	**CLKREQ_OUT_OVERRIDE**: If CLKREQ_OUT_OVERRIDE_EN=1 This value will be used to override PEX_CLKREQ_O_N output to PAD
9	RW	0x0	**CLKREQ_OUT_OVERRIDE_EN**: Override Enable for PEX_CLKREQ_O_N output to PAD
8	RO	0x0	**CORE_CLK_REQ_N**: HW Generated CLKREQ_N. This value will be driven on PEX_CLKREQ_O_N if CLKREQ_OUT_OVERRIDE_EN=0.
7	RO	0x1	**PEX_RST_I_N**: PEX_RST input from PAD. To be used in EP mode to reset the controller by SW based interrupt APPL_INTR_EN_L0.PEX_RST.
5	RW	0x0	**CLK_OUTPUT_IN_OVERRIDE**: If CLK_OUTPUT_IN_OVERRIDE_EN=1, This value will be used to override PEX_REFCLK_OE_N output to PAD
4	RW	0x0	**CLK_OUTPUT_IN_OVERRIDE_EN**: Override Enable for PEX_REFCLK_OE_N output enable to PAD. This bit can be used for Always enable/disable REFCLK output.
3	RW	0x0	**CLKREQ_OVERRIDE**: If CLKREQ_OVERRIDE_EN=1, This value will be used to override PEX_CLKREQ_I_N input from PAD to controller.
2	RW	0x0	**CLKREQ_OVERRIDE_EN**: Override Enable for PEX_CLKREQ_I_N input from PAD

Bit	R/W	Reset	Description
1	RO	0x0	**PEX_CLKREQ_I_N**: PEX_CLKREQ_N input from PAD. CLKREQ_OVERRIDE_EN has no effect on this bit.
0	RW	0x0	**PEX_RST_O_N**: PEX_RST output to PAD. To be used in RC mode to control the reset of link partner.

PCIE_RP_APPL_CTRL_0

Offset: 0x4
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00449000 (0bxxxx,x000,0100,x100,1001,00x0,000x,xxxx)
PROD: 0x00000040 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x1xx,xxxx)

Bit	R/W	Reset	PROD	Description
23:22	RW	0x1	_NONE_	**HW_HOT_RST_MODE**: Effectlive only when HW_HOT_RST_EN is enabled. If DelayedReset(DLY_RST), The hot-reset of the controller is delayed by deasserting app_ltssm_enable bit immediately, allowing SW to read write PCIe Core registers before Reset is performed. If Immediate_Reset(IMDT_RST), Hot reset is asserted to Controller and app_ltssm_enable is deasserted, SW is expected to reprogram app_ltssm_enable for link up. If IMDT_RST_LTSSM_EN, just like Immediate_Reset except app_ltsmm_enable is not deasserted, Link proceeds for training. There are limitation of using IMDT_RST_LTSSM_EN as SW does not get a chance of update link parameter before the link is re-established. **0** = DLY_RST **1** = IMDT_RST **2** = IMDT_RST_LTSSM_EN
21	RW	0x0	_NONE_	**HW_UPHY_HOT_RST_EN**: If enabled HW will generate the reset to UPHY when Hot-Reset condition occurs. **0** = DIS **1** = EN
20	RW	0x0	_NONE_	**HW_HOT_RST_EN**: If enabled HW will generate the reset to Controller and P2U when Hot-Reset condition occurs. **0** = DIS **1** = EN

Bit	R/W	Reset	PROD	Description
18	RW	0x1	_NONE_	**SYS_AUX_PWR_DET**: Auxiliary Power Detected **0** = OFF **1** = ON
17	RW	0x0	_NONE_	**TX_LANE_FLIP_EN**: Performs manual lane reversal for transmit lanes **0** = OFF **1** = ON
16	RW	0x0	_NONE_	**RX_LANE_FLIP_EN**: Performs manual lane reversal for receive lanes **0** = OFF **1** = ON
15	RW	0x1	_NONE_	**APP_CLK_PM_EN**: Clock PM feature enabled by application. **0** = OFF **1** = ON
14	RW	0x0	_NONE_	**APP_XFER_PENDING**: This bit can be set to indicate that AXI transfers pending and prevents the core from entering L1. **0** = OFF **1** = ON
13	RW	0x0	_NONE_	**APP_REQ_EXIT_L1**: SELF-CLEAR : Application request to Exit L1. **0** = OFF **1** = ON
12	RW	0x1	_NONE_	**APP_READY_ENTR_L23**: Application Logic Ready to Enter L23. This bit should be Cleared to prevert entry to L2. **0** = OFF **1** = ON
11	RW	0x0	_NONE_	**APP_REQ_ENTR_L1**: SELF-CLEAR : Application request to Enter L1 ASPM state. This bit can be used to control L1 entry instead of using the L1 entry timer as defined in the PCI Express 3.0 Specification. It is only effective when L1 is enabled. The core latches this request when in L0 or L0s; to be acted upon later. **0** = OFF **1** = ON

Bit	R/W	Reset	PROD	Description
7	RW	0x0	_NONE_	**APP_LTSSM_ENABLE**: HW-CLEAR : Driven low by application after cold, warm or hot reset to hold the LTSSM in the Detect state until our application is ready for the link training to begin. When SW has finished reprogramming the core configuration registers, it should set this bit to allow the LTSSM to continue link establishment. Cleared by HW based on HW_HOT_RST_EN & HW_HOT_RST_MODE. **0** = OFF **1** = ON
6	RW	0x0	0x1	**SYS_PRE_DET_STATE**: Presence Detect State. Indicates whether or not a card is present in the slot **0** = OFF **1** = ON
5	RO	0x0	_NONE_	**SYS_PRE_DET_STATUS**: Status bit to indicate presence detect state (combined status of register SYS_PRE_DET_STATE and HW signal)

PCIE_RP_APPL_INTR_EN_L0_0

NOTE:

All the level 0 (_L0) status registers are read only.
All the level 1 registers are write-one-clear registers.
(Software needs to write to the corresponding register bit to clear the interrupt/fault once it services the interrupt/fault.)

Offset: 0x8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xc0000000 (0b11xx,xxxx,xxx0,0000,0000,0000,00x0,0000)

Bit	Reset	Description
31	0x1	**SYS_MSI_INTR_EN**: Global bit to enable/disable MSI interrupt generation. **0** = DIS **1** = EN
30	0x1	**SYS_INTR_EN**: Global bit to enable/disable Message interrupt generation. **0** = DIS **1** = EN

Bit	Reset	Description
20	0x0	**SAFETY_UNCORR_INT_EN**: Enable for fault generation for the corresponding bit in APPL_INTR_STATUS_L1_0 **0** = DIS **1** = EN
19	0x0	**SAFETY_CORR_INT_EN**: Enable for fault generation for the corresponding bit in APPL_INTR_STATUS_L1_0 **0** = DIS **1** = EN
18	0x0	**CDM_REG_CHK_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L0 **0** = DIS **1** = EN
17	0x0	**AXI_APB_ERR_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L0 **0** = DIS **1** = EN
15	0x0	**PCI_CMD_EN_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L0 **0** = DIS **1** = EN
14	0x0	**PARITY_ERR_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L0 **0** = DIS **1** = EN
13	0x0	**CPL_TIMEOUT_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L0 **0** = DIS **1** = EN
12	0x0	**RASDP_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L0 **0** = DIS **1** = EN
11	0x0	**TLP_ERR_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L0 **0** = DIS **1** = EN

Bit	Reset	Description
10	0x0	**UNLOCK_MSG_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L0 **0** = DIS **1** = EN
9	0x0	**PM_MSG_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L0 **0** = DIS **1** = EN
8	0x0	**INT_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L0 **0** = DIS **1** = EN
7	0x0	**VEN_MSG_SENT_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L0 **0** = DIS **1** = EN
6	0x0	**VEN_MSG_RCV_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L0 **0** = DIS **1** = EN
4	0x0	**MSI_RCV_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L0 **0** = DIS **1** = EN
3	0x0	**LTR_SENT_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L0 **0** = DIS **1** = EN
2	0x0	**LTR_RCV_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L0 **0** = DIS **1** = EN
1	0x0	**ERROR_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L0 **0** = DIS **1** = EN

Bit	Reset	Description
0	0x0	**LINK_STATE_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L0 **0** = DIS **1** = EN

PCIE_RP_APPL_INTR_STATUS_L0_0

Offset: 0xc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxx0,0000,0000,0000,00x0,0000)

Bit	Reset	Description
20	0x0	**SAFETY_UNCORR_INT**: Refer APPL_INTR_STATUS_L1_20. This interrupt status bit is set when any of the internal safety mechanisms reports an uncorrectable error;
19	0x0	**SAFETY_CORR_INT**: Refer APPL_INTR_STATUS_L1_19. This interrupt status bit is set to indicate when a PCIe protocol correctable error or a Data Path protection correctable error has occured
18	0x0	**CDM_REG_CHK_INT**: Refer APPL_INTR_STATUS_L1_18. This interrupt status bit is set to indicate CDM register check has error or completed
17	0x0	**AXI_APB_ERR_INT**: Refer APPL_INTR_STATUS_L1_17. This interrupt status bit is set to indicate AXI/APB error response status bits are set
15	0x0	**PCI_CMD_EN_INT**: Refer APPL_INTR_STATUS_L1_15. This interrupt status bit is set to indicate enable bits in pci command register is changed
14	0x0	**PARITY_ERR_INT**: Refer APPL_INTR_STATUS_L1_14. This interrupt status bit is set to indicate core detects transmit and receive datapath parity err
13	0x0	**CPL_TIMEOUT_INT**: Refer APPL_INTR_STATUS_L1_13. This interrupt status bit is set to indicate completion timeout is signaled
12	0x0	**RASDP_INT**: Refer APPL_INTR_STATUS_L1_12. This interrupt status bit is set to indicate error mode signals are set
11	0x0	**TLP_ERR_INT**: Refer APPL_INTR_STATUS_L1_11. This interrupt status bit is set to indicate error message send signals are set

Bit	Reset	Description
10	0x0	**UNLOCK_MSG_INT**: Refer APPL_INTR_STATUS_L1_10. This interrupt status bit is set to indicate unlock message signals are set
9	0x0	**PM_MSG_INT**: Refer APPL_INTR_STATUS_L1_9. This interrupt status bit is set to indicate power management message signals are set
8	0x0	**INT_INT**: Refer APPL_INTR_STATUS_L1_8. This interrupt status bit is set to indicate sii interrupt signals are set
7	0x0	**VEN_MSG_SENT_INT**: Refer APPL_INTR_STATUS_L1_7. This interrupt status bit is set to indicate vendor defined message is sent
6	0x0	**VEN_MSG_RCV_INT**: Refer APPL_INTR_STATUS_L1_6. This interrupt status bit is set to indicate vendor defined message is received
4	0x0	**MSI_RCV_INT**: Refer APPL_INTR_STATUS_L1_4. This interrupt status bit is set to indicate msi is received
3	0x0	**LTR_SENT_INT**: Refer APPL_INTR_STATUS_L1_3. This interrupt status bit is set to indicate LTR is sent
2	0x0	**LTR_RCV_INT**: Refer APPL_INTR_STATUS_L1_2. This interrupt status bit is set to indicate LTR is received
1	0x0	**ERROR_INT**: Refer APPL_INTR_STATUS_L1_1. This interrupt status bit is set to indicate error signals are set
0	0x0	**LINK_STATE_INT**: Refer APPL_INTR_STATUS_L1_0. This interrupt status bit is set to indicate link reset/status signals are set

PCIE_RP_APPL_FAULT_EN_L0_0

Offset: 0x10
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x80000000 (0b1xxx,xxxx,xxx0,000x,x000,0xx0,xxxx,xx00)

Bit	Reset	Description
31	0x1	**SYS_FAULT_EN**: Global bit to enable/disable fault generation **0** = DIS **1** = EN
20	0x0	**SAFETY_UNCORR_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_FAULT_STATUS_L0 **0** = DIS **1** = EN
19	0x0	**SAFETY_CORR_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_FAULT_STATUS_L0 **0** = DIS **1** = EN
18	0x0	**CDM_REG_CHK_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_FAULT_STATUS_L0 **0** = DIS **1** = EN
17	0x0	**AXI_APB_ERR_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_FAULT_STATUS_L0 **0** = DIS **1** = EN
14	0x0	**PARITY_ERR_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_FAULT_STATUS_L0 **0** = DIS **1** = EN
13	0x0	**CPL_TIMEOUT_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_FAULT_STATUS_L0 **0** = DIS **1** = EN
12	0x0	**RASDP_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_FAULT_STATUS_L0 **0** = DIS **1** = EN
11	0x0	**TLP_ERR_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_FAULT_STATUS_L0 **0** = DIS **1** = EN

Bit	Reset	Description
8	0x0	**INT_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_FAULT_STATUS_L0 **0** = DIS **1** = EN
1	0x0	**ERROR_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_FAULT_STATUS_L0 **0** = DIS **1** = EN
0	0x0	**LINK_STATE_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_FAULT_STATUS_L0 **0** = DIS **1** = EN

PCIE_RP_APPL_FAULT_STATUS_L0_0

Offset: 0x14
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxx0,000x,x000,0xx0,xxxx,xx00)

Bit	Reset	Description
20	0x0	**SAFETY_UNCORR_FAULT**: Refer APPL_INTR_STATUS_L1_20. This fault status bit is set when any of the internal safety mechanisms reports an uncorrectable error;
19	0x0	**SAFETY_CORR_FAULT**: Refer APPL_INTR_STATUS_L1_19. This fault status bit is set to indicate when a PCIe protocol correctable error or a Data Path protection correctable error has occured
18	0x0	**CDM_REG_CHK_FAULT**: Refer APPL_INTR_STATUS_L1_18. This fault status bit is set to indicate CDM register check has error or completed
17	0x0	**AXI_APB_ERR_FAULT**: Refer APPL_INTR_STATUS_L1_17. This fault status bit is set to indicate AXI/APB error response status bits are set
14	0x0	**PARITY_ERR_FAULT**: Refer APPL_INTR_STATUS_L1_14. This fault status bit is set to indicate core detects transmit and receive datapath parity err

Bit	Reset	Description
13	0x0	**CPL_TIMEOUT_FAULT**: Refer APPL_INTR_STATUS_L1_13. This fault status bit is set to indicate completion timeout is signaled
12	0x0	**RASDP_FAULT**: Refer APPL_INTR_STATUS_L1_12. This fault status bit is set to indicate error mode signals are set
11	0x0	**TLP_ERR_FAULT**: Refer APPL_INTR_STATUS_L1_11. This fault status bit is set to indicate error message send signals are set
8	0x0	**INT_FAULT**: Refer APPL_INTR_STATUS_L1_8. This fault status bit is set to indicate sii interrupt signals are set
1	0x0	**ERROR_FAULT**: Refer APPL_INTR_STATUS_L1_1. This fault status bit is set to indicate error signals are set
0	0x0	**LINK_STATE_FAULT**: Refer APPL_INTR_STATUS_L1_0. This fault status bit is set to indicate link reset/status signals are set

PCIE_RP_APPL_FAULT_EN_L1_0_0

Offset: 0x18
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**SURPRISE_DOWN_ERR_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_INTR_STATUS_L1_0 **0** = DIS **1** = EN
3	0x0	**RDLH_LINK_UP_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_INTR_STATUS_L1_0 **0** = DIS **1** = EN
2	0x0	**SMLH_REQ_RST_NOT_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_INTR_STATUS_L1_0 **0** = DIS **1** = EN

Bit	Reset	Description
1	0x0	**LINK_REQ_RST_NOT_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_INTR_STATUS_L1_0 **0** = DIS **1** = EN
0	0x0	**SMLH_LINK_UP_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_INTR_STATUS_L1_0 **0** = DIS **1** = EN

PCIE_RP_APPL_INTR_EN_L1_0_0

Offset: 0x1c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b00xx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
30	0x0	**HOT_RESET_DONE_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_0 **0** = DIS **1** = EN
5	0x0	**PM_DSTATE_CHGED_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_0 **0** = DIS **1** = EN
4	0x0	**SURPRISE_DOWN_ERR_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_0 **0** = DIS **1** = EN
3	0x0	**RDLH_LINK_UP_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_0 **0** = DIS **1** = EN

Bit	Reset	Description
2	0x0	**SMLH_REQ_RST_NOT_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_0 **0** = DIS **1** = EN
1	0x0	**LINK_REQ_RST_NOT_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_0 **0** = DIS **1** = EN
0	0x0	**SMLH_LINK_UP_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_0 **0** = DIS **1** = EN

PCIE_RP_APPL_INTR_STATUS_L1_0_0

Offset: 0x20
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b00xx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
30	0x0	**HOT_RESET_DONE**: WRITE-ONE-CLEAR : Status bit to indicate HW hot reset sequence has finished.
5	0x0	**PM_DSTATE_CHGED**: WRITE-ONE-CLEAR : Status bit to indicate that pm dstate has changed from either non-D0 to D0 or vice-versa.
4	0x0	**SURPRISE_DOWN_ERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate that surprise link down is detected
3	0x0	**RDLH_LINK_UP_CHGED**: WRITE-ONE-CLEAR : Status bit to indicate change in Data link layer link status.
2	0x0	**SMLH_REQ_RST_NOT_CHGED**: WRITE-ONE-CLEAR : Status bit to indicate a reset is requested because the link has gone down or the core received a hot-reset request. Early indication that Link down is imminent.

Bit	Reset	Description
1	0x0	**LINK_REQ_RST_NOT_CHGED**: WRITE-ONE-CLEAR : Status bit to indicate a reset is requested because the link has gone down or the core received a hot-reset request. Asserted after the Controller Datapath is flushed.
0	0x0	**SMLH_LINK_UP_CHGED**: WRITE-ONE-CLEAR : Status bit to indicate change in link status

PCIE_RP_APPL_FAULT_EN_L1_1_0

Offset: 0x24
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**RADM_QOVERFLOW_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_INTR_STATUS_L1_1 **0** = DIS **1** = EN
3	0x0	**RADM_FATAL_ERR_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_INTR_STATUS_L1_1 **0** = DIS **1** = EN
2	0x0	**RADM_NONFATAL_ERR_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_INTR_STATUS_L1_1 **0** = DIS **1** = EN
1	0x0	**RADM_CORRECTABLE_ERR_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_INTR_STATUS_L1_1 **0** = DIS **1** = EN
0	0x0	**CFG_SYS_ERR_RC_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_INTR_STATUS_L1_1 **0** = DIS **1** = EN

PCIE_RP_APPL_INTR_EN_L1_1_0

Offset: 0x28
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**RADM_QOVERFLOW_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_1 **0** = DIS **1** = EN
3	0x0	**RADM_FATAL_ERR_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_1 **0** = DIS **1** = EN
2	0x0	**RADM_NONFATAL_ERR_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_1 **0** = DIS **1** = EN
1	0x0	**RADM_CORRECTABLE_ERR_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_1 **0** = DIS **1** = EN
0	0x0	**CFG_SYS_ERR_RC_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_1 **0** = DIS **1** = EN

PCIE_RP_APPL_INTR_STATUS_L1_1_0

Offset: 0x2c
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**RADM_QOVERFLOW_INT**: WRITE-ONE-CLEAR : Status bit to indicate Receive Datapath (RADM) Queue has overflown.
3	0x0	**RADM_FATAL_ERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate the core received an ERR_FATAL message. Message header is available in APPL_RADM_MSG_PAYLOAD1 & APPL_RADM_MSG_PAYLOAD0 Registers.
2	0x0	**RADM_NONFATAL_ERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate the core received an ERR_NONFATAL message. Message header is available in APPL_RADM_MSG_PAYLOAD1 & APPL_RADM_MSG_PAYLOAD0 Registers.
1	0x0	**RADM_CORRECTABLE_ERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate the core received an ERR_COR message. Message header is available in APPL_RADM_MSG_PAYLOAD1 & APPL_RADM_MSG_PAYLOAD0 Registers.
0	0x0	**CFG_SYS_ERR_RC_STATE**: WRITE-ONE-CLEAR : Status bit to indicate a system error is detected. It indicates if any device in the hierarchy reports any of these errors: ERR_COR, ERR_FATAL, ERR_NONFATAL. Also asserted when an internal error is detected.

PCIE_RP_APPL_INTR_STATUS_L1_2_0

Offset: 0x30
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**RADM_MSG_LTR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate the core receives an LTR message. Message header is available in APPL_RADM_MSG_PAYLOAD1 & APPL_RADM_MSG_PAYLOAD0 Registers.

PCIE_RP_APPL_INTR_STATUS_L1_3_0

Offset: 0x34
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**APP_LTR_MSG_GRANT_STATE**: WRITE-ONE-CLEAR : Status bit to indicate request have been accepted to send an LTR message. Refer : APPL_LTR_MSG_1/APPL_LTR_MSG_2

PCIE_RP_APPL_INTR_STATUS_L1_4_0

Offset: 0x38
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**MSI_CTRL_INT_VEC_STATE**: Status bit to indicate that MSI/MSI-X message has received from the corresponding endpoint. One bit per Endpoint. Refer to PF0_PORT_LOGIC_MSI_CTRL_INT_<EP_NUM>_STATUS_OFF to find the corresponding MSI Vector.

PCIE_RP_APPL_INTR_STATUS_L1_6_0

Offset: 0x3c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**RADM_VENDOR_MSG_STATE**: WRITE-ONE-CLEAR : Status bit to indicate that the core received a vendor-defined message. Corresponding Payload can be accessed from APPL_VENDOR_MSG_PAYLOAD1 & APPL_VENDOR_MSG_PAYLOAD0.

PCIE_RP_APPL_INTR_STATUS_L1_7_0

Offset: 0x40
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**VEN_MSG_GRANT_STATE**: WRITE-ONE-CLEAR : Status bit to indicate request have been accepted to send the Vendor Defined Message requested using APPL_VDM_MSG_1/ APPL_VDM_MSG_0.

PCIE_RP_APPL_INTR_EN_L1_8_0

Offset: 0x44
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,x000,0xxx,0xxx,x000,0000)

Bit	Reset	Description
18	0x0	**PCIE_ELPG_READY_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_8 **0** = DIS **1** = EN
17	0x0	**HP_APPL_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_8 **0** = DIS **1** = EN
16	0x0	**HP_MSI_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_8 **0** = DIS **1** = EN
15	0x0	**CFG_AER_RC_ERR_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_8 **0** = DIS **1** = EN

Bit	Reset	Description
11	0x0	**INTX_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_8 **0** = DIS **1** = EN
6	0x0	**EDMA_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_8 **0** = DIS **1** = EN
5	0x0	**CFG_PME_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_8 **0** = DIS **1** = EN
4	0x0	**CFG_LINK_EQ_REQ_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_8 **0** = DIS **1** = EN
3	0x0	**CFG_LINK_AUTO_BW_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_8 **0** = DIS **1** = EN
2	0x0	**CFG_BW_MGT_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_8 **0** = DIS **1** = EN
1	0x0	**HP_PME_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_8 **0** = DIS **1** = EN
0	0x0	**HP_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_8 **0** = DIS **1** = EN

PCIE_RP_APPL_FAULT_EN_L1_8_0

Offset: 0x48
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0xxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
15	0x0	**CFG_AER_RC_ERR_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_INTR_STATUS_L1_8 **0** = DIS **1** = EN

PCIE_RP_APPL_INTR_STATUS_L1_8_0

Offset: 0x4c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
19	RW	0x0	**PCIE_ELPG_READY_STATE**: WRITE-ONE-CLEAR : Status bit to indicate PPC disconnect completed with PCIe requests flushes and UPHY in P4 to ELPG the partition
18	RW	0x0	**HP_APPL_STATE_CHGED**: WRITE-ONE-CLEAR : Status bit to indicate presence detect state change status
17	RW	0x0	**HP_MSI_STATE**: WRITE-ONE-CLEAR : Status bit to indicate all of the following conditions are true:
16	RW	0x0	**CFG_AER_RC_ERR_INT**: WRITE-ONE-CLEAR : Status bit to indicate a reported error condition causes a bit to be set in the Root Error Status register and the associated error message reporting enable bit is set in the Root Error Command register. cfg_aer_rc_err_int is set when the RC internally generates an error or when an error message is received by the RC. Since the RC itself generates it, this needs to be propagated up to the system software which would then need to read the error registers to see which error occurred.
15	RO	0x0	**INTD_STATE**: Status bit to indicate the core received an Assert_INTD Message from the downstream device.

Bit	R/W	Reset	Description
14	RO	0x0	**INTC_STATE**: Status bit to indicate the core received an Assert_INTC Message from the downstream device.
13	RO	0x0	**INTB_STATE**: Status bit to indicate the core received an Assert_INTB Message from the downstream device.
12	RO	0x0	**INTA_STATE**: Status bit to indicate the core received an Assert_INTA Message from the downstream device.
11:6	RO	0x0	**EDMA_INT**: Status bit to indicate the DMA transfer has completed or that an error has occurred
5	RW	0x0	**CFG_PME_INT**: WRITE-ONE-CLEAR : Status bit to indicate all of the following conditions are true: --> The INTx Assertion Disable bit in the Command register is 0. --> The PME Interrupt Enable bit in the Root Control register is set to 1. --> The PME Status bit in the Root Status register is set to 1.
4	RO	0x0	**CFG_LINK_EQ_REQ_INT_STATE**: Status bit to indicate the Link Equalization Request bit in the Link Status 2 Register has been set.
3	RW	0x0	**CFG_LINK_AUTO_BW_INT_STATE**: WRITE-ONE-CLEAR : Status bit to indicate the Link Autonomous Bandwidth Status register (Link Status register bit 15) is updated and the Link Autonomous Bandwidth Interrupt Enable (Link Control register bit 11) is set. For upstream port: Reserved.
2	RW	0x0	**CFG_BW_MGT_INT_STATE**: WRITE-ONE-CLEAR : Status bit to indicate the Link Bandwidth Management Status register (Link Status register bit 14) is updated and the Link Bandwidth Management Interrupt Enable (Link Control register bit 10) is set. This bit is not applicable to, and is reserved, for endpoint devices.
1	RW	0x0	**HP_PME_STATE**: WRITE-ONE-CLEAR : Status bit to indicate all of the following conditions are true: -->The PME Enable bit in the Power Management Control and Status register is set to 1. -->Any bit in the Slot Status register transitions from 0 to 1 and the associated event notification is enabled in the Slot Control register. The core does not check if the PM state is D1, D2, or D3hot. It is up to the application to check the value pm_dstate to make sure the device is in D1, D2, or D3hot.
0	RO	0x0	**HP_INT**: Status bit to indicate all of the following conditions are true: --> The INTx Assertion Disable bit in the Command register is 0. --> Hot-Plug interrupts are enabled in the Slot Control register. --> Any bit in the Slot Status register is equal to 1, and the associated event notification is enabled in the Slot Control register.

PCIE_RP_APPL_INTR_EN_L1_9_0

Offset: 0x50
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,000x)

Bit	Reset	Description
4	0x0	**RADM_PM_STATUS_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_9 **0** = DIS **1** = EN
3	0x0	**RADM_PM_TURNOFF_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_9 **0** = DIS **1** = EN
2	0x0	**RADM_PM_TO_ACK_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_9 **0** = DIS **1** = EN
1	0x0	**RADM_PM_PME_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_9 **0** = DIS **1** = EN

PCIE_RP_APPL_INTR_STATUS_L1_9_0

Offset: 0x54
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,000x)

Bit	R/W	Reset	Description
4	RO	0x0	**RADM_PM_STATUS_STATE**: READ-ONLY : PME Status bit from the PMCSR

Bit	R/W	Reset	Description
3	RW	0x0	**RADM_PM_TURNOFF_STATE**: WRITE-ONE-CLEAR : Status bit to indicate that the core received a PME Turnoff message. When RX_TLP > 1 and when two messages of the same type are received in the same clock cycle (back-to-back), then no indication is given for the second message. The core makes the message header available the radm_msg_payload output.
2	RW	0x0	**RADM_PM_TO_ACK_STATE**: WRITE-ONE-CLEAR : Status bit to indicate that the core received a PME_TO_Ack message. The core makes the message header available the radm_msg_payload output. Upstream port: Reserved.
1	RW	0x0	**RADM_PM_PME_STATE**: WRITE-ONE-CLEAR : Status bit to indicate that the core received a PM_PME message. The core makes the message header available the radm_msg_payload output

PCIE_RP_APPL_INTR_STATUS_L1_10_0

Offset: 0x58
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**RADM_MSG_UNLOCK_STATE**: WRITE-ONE-CLEAR : Status bit to indicate the core receives an UNLOCK message. Message header is available in APPL_RADM_MSG_PAYLOAD1 & APPL_RADM_MSG_PAYLOAD0 Registers.

PCIE_RP_APPL_FAULT_EN_L1_11_0

Offset: 0x5c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**CFG_SEND_NF_ERR_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_INTR_STATUS_L1_11 **0** = DIS **1** = EN
1	0x0	**CFG_SEND_F_ERR_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_INTR_STATUS_L1_11 **0** = DIS **1** = EN
0	0x0	**CFG_SEND_COR_ERR_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_INTR_STATUS_L1_11 **0** = DIS **1** = EN

PCIE_RP_APPL_INTR_EN_L1_11_0

Offset: 0x60
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**CFG_SEND_NF_ERR_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_11 **0** = DIS **1** = EN
1	0x0	**CFG_SEND_F_ERR_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_11 **0** = DIS **1** = EN
0	0x0	**CFG_SEND_COR_ERR_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_11 **0** = DIS **1** = EN

PCIE_RP_APPL_INTR_STATUS_L1_11_0

Offset: 0x64
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**CFG_SEND_NF_ERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate that Core has sent a message towards the Root Complex indicating that an Rx TLP that contained a fatal error, and that can not be corrected, was received by the EndPoint
1	0x0	**CFG_SEND_F_ERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate that Core has sent a message towards the Root Complex indicating that an Rx TLP that contained an non-fatal error, and that can not be corrected, was received by the EndPoint
0	0x0	**CFG_SEND_COR_ERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate that Core has sent a message towards the Root Complex indicating that an Rx TLP that contained an error, and that can be corrected, was received by the EndPoint

PCIE_RP_APPL_FAULT_EN_L1_12_0

Offset: 0x68
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**SLV_RASDP_ERR_MODE_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_INTR_STATUS_L1_12
0	0x0	**MSTR_RASDP_ERR_MODE_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_INTR_STATUS_L1_12

PCIE_RP_APPL_INTR_EN_L1_12_0

Offset: 0x6c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**SLV_RASDP_ERR_MODE_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_12
0	0x0	**MSTR_RASDP_ERR_MODE_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_12

PCIE_RP_APPL_INTR_STATUS_L1_12_0

Offset: 0x70
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**SLV_RASDP_ERR_MODE**: SLV_ACLK version: Status bit to indicate that the core that it has entered RASDP error mode.The core enters RASDP error mode (if the ERROR_MODE_EN register field =1) upon detection of the first uncorrectable error.
0	0x0	**MSTR_RASDP_ERR_MODE**: MSTR_ACLK version: Status bit to indicate that the core has entered RASDP error mode.The core enters RASDP error mode (if the ERROR_MODE_EN register field =1) upon detection of the first uncorrectable error.

PCIE_RP_APPL_INTR_STATUS_L1_13_0

Offset: 0x74
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**RADM_CPL_TIMEOUT_STATE**: WRITE-ONE-CLEAR : Status bit to indicate that the completion TLP for a request has not been received within the expected time window. Refer : APPL_RADM_CPL_TO

PCIE_RP_APPL_INTR_STATUS_L1_14_0

Offset: 0x78
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29	0x0	**IB_WREQ_C2A_CDC_RAM_ADDRB_PAR_ERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate address parity error on Inbound Write Req C2A CDC RAM
28	0x0	**IB_RREQ_ORDR_RAM_ADDRB_PAR_ERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate address parity error on Inbound Read Req Order RAM
27	0x0	**IB_RREQ_C2A_CDC_RAM_ADDRB_PAR_ERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate address parity error on Inbound Read Req C2A CDC RAM
26	0x0	**IB_MCPL_SB_RAM_ADDRB_PAR_ERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate address parity error on Inbound Master CPL SB RAM
25	0x0	**IB_MCPL_A2C_CDC_RAM_ADDRB_PAR_ERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate address parity error on Inbound Master CPL A2C CDC RAM
24	0x0	**XDLH_RETRYSOTRAM_RADDR_PAR_ERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate address parity error on XDLH Retry SOT RAM
23	0x0	**XDLH_RETRYRAM_ADDR_PAR_ERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate address parity error on XDLH Retry Buffer RAM
22	0x0	**SLV_NPW_SAB_RAM_ADDRB_PAR_ERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate address parity error on Slv Non Posted Write Sab RAM
21	0x0	**P_HDRQ_ADDRB_PAR_ERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate address parity error on Posted Hdr Queue RAM
20	0x0	**P_DATAQ_ADDRB_PAR_ERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate address parity error on Posted Data Queue RAM

Bit	Reset	Description
19	0x0	**OB_PDCMP_HDR_RAM_ADDRB_PAR_ERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate address parity error on Outbound Posted Decompose Hdr RAM
18	0x0	**OB_PDCMP_DATA_RAM_ADDRB_PAR_ERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate address parity error on Outbound Posted Decompose Data RAM
17	0x0	**OB_NPDCMP_RAM_ADDRB_PAR_ERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate address parity error on Outbound Non Posted Decompose RAM
16	0x0	**OB_CPL_C2A_CDC_RAM_ADDRB_PAR_ERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate address parity error on Outbound Completion C2A CDC RAM
15	0x0	**OB_CCMP_DATA_RAM_ADDRB_PAR_ERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate address parity error on Outbound Completion Compose Data RAM
14	0x0	**EDMARBUFF2RAM_ADDRB_PAR_ERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate address parity error on EDMA RBUFF RAM
13	0x0	**EDMA2RAM_ADDRB_PAR_ERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate address parity error on EDMA RAM
12:11	0x0	**MSTR_AW_ORDR_PARITY_ERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate parity error on Master Write Address Channelbefore Ordering Address translation: [0] Parity Error detected, [1] Latent Error detected
10:9	0x0	**SLV_R_PARITY_ERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate parity error on Slave read Data Channel: [0] Parity Error detected, [1] Latent Error detected
8:7	0x0	**MSTR_W_PARITY_ERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate parity error on Master Write Data Channel: [0] Parity Error detected, [1] Latent Error detected
6:5	0x0	**MSTR_AW_PARITY_ERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate parity error on Master Write Address Channel: [0] Parity Error detected, [1] Latent Error detected
4:3	0x0	**MSTR_AR_PARITY_ERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate parity error on Master Read Address Channel: [0] Parity Error detected, [1] Latent Error detected
2:0	0x0	**APP_PARITY_ERRS_STATE**: WRITE-ONE-CLEAR : Status bit to indicate that the core detected a datapath parity error, one bit for each of the following parity errors: --> app_parity_errs[0]: Parity error at front end of the transmit datapath. --> app_parity_errs[1]: Parity error at back end of the transmit datapath. --> app_parity_errs[2]: Parity error the receive datapath.

PCIE_RP_APPL_INTR_STATUS_L1_15_0

Offset: 0x7c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**CFG_BUS_MASTER_EN_CHGED**: WRITE-ONE-CLEAR : Status bit to indicate that the Host Software has updated Bus Master Enable (BME)
0	0x0	**CFG_MEM_SPACE_EN_CHGED**: WRITE-ONE-CLEAR : Status bit to indicate that the Host Software has updated Memory Space Enable (MSE)

PCIE_RP_APPL_FAULT_EN_L1_17_0

Offset: 0x80
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**APB_PSLVERR_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_INTR_STATUS_L1_17
3	0x0	**AXI_SLV_RRESP_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_INTR_STATUS_L1_17
2	0x0	**AXI_SLV_BRESP_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_INTR_STATUS_L1_17
1	0x0	**AXI_MSTR_RRESP_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_INTR_STATUS_L1_17
0	0x0	**AXI_MSTR_BRESP_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_INTR_STATUS_L1_17

PCIE_RP_APPL_INTR_EN_L1_17_0

Offset: 0x84
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**APB_PSLVERR_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_17
3	0x0	**AXI_SLV_RRESP_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_17
2	0x0	**AXI_SLV_BRESP_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_17
1	0x0	**AXI_MSTR_RRESP_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_17
0	0x0	**AXI_MSTR_BRESP_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_17

PCIE_RP_APPL_INTR_STATUS_L1_17_0

Offset: 0x88
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4	0x0	**APB_PSLVERR_STATE**: WRITE-ONE-CLEAR : Status bit to indicate that APB error has occured
3	0x0	**AXI_SLV_RRESP_STATE**: WRITE-ONE-CLEAR : Status bit to indicate that AXI slave read response error has occured
2	0x0	**AXI_SLV_BRESP_STATE**: WRITE-ONE-CLEAR : Status bit to indicate that AXI slave write response error has occured
1	0x0	**AXI_MSTR_RRESP_STATE**: WRITE-ONE-CLEAR : Status bit to indicate that AXI master read response error has occured

Bit	Reset	Description
0	0x0	**AXI_MSTR_BRESP_STATE**: WRITE-ONE-CLEAR : Status bit to indicate that AXI master write response error has occured

PCIE_RP_APPL_FAULT_EN_L1_18_0

Offset: 0x8c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**CDM_REG_CHK_CMP_ERR_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_INTR_STATUS_L1_18
0	0x0	**CDM_REG_CHK_LOGIC_ERR_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_INTR_STATUS_L1_18

PCIE_RP_APPL_INTR_EN_L1_18_0

Offset: 0x90
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**CDM_REG_CHK_CMPLT_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_18
1	0x0	**CDM_REG_CHK_CMP_ERR_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_18
0	0x0	**CDM_REG_CHK_LOGIC_ERR_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_18

PCIE_RP_APPL_INTR_STATUS_L1_18_0

Offset: 0x94
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**CDM_REG_CHK_CMPLT_STATE**: Status bit to indicate that Register Checking Sequence has Completed. Place holder Register. Needs to be cleared in CDM register
1	0x0	**CDM_REG_CHK_CMP_ERR_STATE**: Status bit to indicate that the register values read from both CDM's do not match. Place holder Register. Needs to be cleared in CDM register
0	0x0	**CDM_REG_CHK_LOGIC_ERR_STATE**: Status bit to indicate that there is an Error in the Register Checking Logic. Place holder Register. Needs to be cleared in CDM register

PCIE_RP_AUTOMOTIVE_DEBUG_REG_0

Offset: 0x98
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**CDM_REG_CHK_TEST_EN**: Signal to enter test mode in the Register Checking Logic. Shadow copy(u_cdm_b) of the controller registers will not be written to when this signal is high. Allows a value to be written to Original SET (u_cdm) to compare against the previous value in u_cdm_b. If this bit is set and a controller register is updated, would trigger APPL_INTR_STATUS_L1_18 when the CDM register check is enabled.

PCIE_RP_APPL_VDM_MSG_0_0

Register space mapped for different vendor message header fields.
The mapping is as follows:

- ven_msg_len --> ven_msg_hdr0[9:0]

- ven_msg_fmt --> ven_msg_hdr0[11:10]
- ven_msg_attr --> ven_msg_hdr0[13:12]
- ven_msg_ep --> ven_msg_hdr0[14]
- ven_msg_td --> ven_msg_hdr0[15]
- ven_msg_func_num --> ven_msg_hdr0[18:16]
- ven_msg_tc --> ven_msg_hdr0[22:20]
- ven_msg_type --> ven_msg_hdr0[28:24]
- ven_msg_code --> ven_msg_hdr1[7:0]
- ven_msg_tag --> ven_msg_hdr1[15:8]
- ven_msg_data --> {ven_msg_hdr3,ven_msg_hdr2}

Offset: 0x9c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**VEN_MSG_REQ_STATE**: HW-CLEAR : When set to 1, VDM message will be transmitted using the fields VEN_MSG_HDR0/1/2/3.
30:0	0x0	**VEN_MSG_HDR0**: {ven_msg_type[4:0],1'b0,ven_msg_tc[2:0],1'b0,'ven_msg_func_num[2:0],ven_msg_td,ven_msg_ep,ven_msg_attr[1:0],ven_msg_fmt[1:0],ven_msg_len[9:0]}

PCIE_RP_APPL_VDM_MSG_1_0

Offset: 0xa0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VEN_MSG_HDR1**: {ven_msg_tag[7:0],ven_msg_code[7:0]}

PCIE_RP_APPL_VDM_MSG_2_0

Offset: 0xa4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VEN_MSG_HDR2**: VDM Payload Lower DWORD

PCIE_RP_APPL_VDM_MSG_3_0

Offset: 0xa8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VEN_MSG_HDR3**: VDM Payload Upper DWORD

PCIE_RP_APPL_MSI_CTRL_1_0

Offset: 0xac
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**SEND_MSI_REQ**: SELF-CLEAR : When a bit is set, corresponding MSI Vector is sent out as a MSI msg, Is the vector is masked, corresponding bit is set in CFG_MSI_PENDING. Writing 0 has no effect.

PCIE_RP_APPL_MSI_CTRL_2_0

Offset: 0xb0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CFG_MSI_PENDING**: WRITE-ONE-CLEAR : Current status of the MSI Pending bits set by HW. Write 1 to clear the bit once remote RP SW has handled the interrupt. Writing 0 has no effect on the bits. This bit is set if a requested interrupt is masked or a unmasked interrupt is waiting for arbiteration.

PCIE_RP_APPL_MSI_CTRL_3_0

Offset: 0xb4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:3	0x0	**VEN_MSI_TC**: Traffic Class to be used all MSI message being transmitted. Refer : APPL_MSI_CTRL_1.SEND_MSI_REQ
2:0	0x0	**VEN_MSI_FUNC_NUM**: CYA : Must be set to 0. to be used all MSI message being transmitted. Refer : APPL_MSI_CTRL_1.SEND_MSI_REQ

PCIE_RP_APPL_LEGACY_INTX_0

Offset: 0xb8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SYS_INT**: sys_int is intended to generate a message that emulates the legacy PCI Interrupts. 0->1 will send ASSERT_INTA. 1->0 will send DEASSERT_INTA message. **0** = OFF **1** = ON

PCIE_RP_APPL_DIAG_CTRL_0

Diagnostic Control Bus

x01: Insert LCRC error by inverting the LSB of LCRC

x10: Insert ECRC error by inverting the LSB of ECRC

The rising edge of these two signals ([1:0]) enable the core to assert an LCRC or ECRC to the packet that it currently being transferred.
When LCRC or ECRC error packets are transmitted by the core, the core asserts diag_status_bus[lcrc_err_asserted] or diag_status_bus[ecrc_err_asserted] to report that the requested action has been completed.
This handshake between control and status allows your application to control a specific packet being injected with an CRC or ECRC error.
The LCRC and ECRC errors are generated by simply inverting the last bit of the LCRC or ECRC value.

1xx: Select Fast Link Mode.

- Sets all internal timers to Fast Mode for speeding up simulation purposes.
- Forces the LTSSM training (link initialization) to use shorter timeouts and to link up faster.

The scaling factor is 1024 for all internal timers.
Fast Link Mode can also be activated by setting the "Fast Link Mode" bit of the "Port Link Control Register".

Offset: 0xc0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	0x0	**DIAG_CTRL_BUS**: 'b1xx: Select Fast Link Mode 'bx01: Insert LCRC error by inverting the LSB of LCRC 'bx10: Insert ECRC error by inverting the LSB of ECRC

PCIE_RP_APPL_LTR_MSG_1_0

Offset: 0xc4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**APP_LTR_MSG_LATENCY**: Latency Values to be used by the LTR message to be transmitted. Refer : APPL_LTR_MSG_2.APP_LTR_MSG_REQ_STATE. Format is same as the defined in PCIe Spec.

PCIE_RP_APPL_LTR_MSG_2_0

Offset: 0xc8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	0x0	**APP_LTR_MSG_REQ_STATE**: HW-CLEAR : EP-MODE-ONLY. Request for LTR message to be transmitter. When set to 1 LTR message is transmitter with APPL_LTR_MSG_1.APP_LTR_MSG_LATENCY as payload.
2:0	0x0	**APP_LTR_MSG_FUNC_NUM**: Function Number to be used for LTR message.

PCIE_RP_APPL_LINK_STATUS_0

Offset: 0xcc
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000002 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10)

Bit	R/W	Reset	Description
1	RW	0x1	**APP_CLK_REQ_N**: Set to 0 if removal of ref_clk (by asserting clkreq_n) is not desired by application logic.
0	RO	0x0	**RDLH_LINK_UP**: Live status of Data Link Layer.

PCIE_RP_APPL_DEBUG_0

Offset: 0xd0
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xx00,0000,0000,0x00,0000,0000)

Bit	R/W	Reset	Description
21	RW	0x0	**PM_LINKST_IN_L2_LAT**: WRITE-ONE_CLEAR : Sticky version of PM_LINKST_IN_L2. Latches a value of 1 when PM_LINKST_IN_L2 goes high.
20	RO	0x0	**AUX_PM_EN**: DEBUG STATUS : Auxiliary Power Enable bit in the Device Control register.
19	RO	0x0	**PM_STATUS**: DEBUG STATUS : PME Status bit from the PMCSR.
18	RO	0x0	**PM_LINKST_IN_L1SUB**: DEBUG STATUS : Power management is in L1 substate. Indicates when the link has entered L1 substates. It is used in clk_rst.v (see 'Clock Generation and Gating Design Example') to ensure that the switching back of aux_clk from AUXCLK to PCLK occurs only after L1 substates have been exited. For L1.2 this signal is deasserted at the end of the L1.2.Exit state, after the t_power_on constraint has been satisfied. External logic can use the transition high to low on this signal to initiate REFCLK restore.
17	RO	0x0	**PM_LINKST_L2_EXIT**: DEBUG STATUS : Power management is exiting L2 state. Not applicable for downstream port.
16	RO	0x0	**PM_LINKST_IN_L2**: Power management is in L2 state . SW polls this register to enable partition power down sequence.
15	RO	0x0	**PM_LINKST_IN_L1**: DEBUG STATUS : Power management is in L1 state
14	RO	0x0	**PM_LINKST_IN_L0s**: DEBUG STATUS : Power management is in L0s state

Bit	R/W	Reset	Description
13:11	RO	0x0	**PM_DSTATE**: DEBUG STATUS : The current power management D-state of the function
9	RO	0x0	**SMLH_LTSSM_STATE_RCVRY_EQ**: DEBUG STATUS :

Bit	R/W	Reset	Description
8:3	RO	0x0	**SMLH_LTSSM_STATE**: DEBUG STATUS : Current state of the LTSSM. Encoding is defined as: ▪ 6'h00: S_DETECT_QUIET ▪ 6'h01: S_DETECT_ACT ▪ 6'h02: S_POLL_ACTIVE ▪ 6'h03: S _POLL_COMPLIANCE ▪ 6'h04: S_POLL_CONFIG ▪ 6'h05: S_PRE_DETECT_QUIET ▪ 6'h06: S_DETECT_WAIT ▪ 6'h07: S_CFG_LINKWD_START ▪ 6'h08: S_CFG_LINKWD_ACEPT ▪ 6'h09: S_CFG_LANENUM_WAI ▪ 6'h0A: S_CFG_LANENUM_ACEPT ▪ 6'h0B: S_CFG_COMPLETE ▪ 6'h0C: S_CFG_IDLE ▪ 6'h0D: S_RCVRY_LOCK ▪ 6'h0E: S_RCVRY_SPEED ▪ 6'h0F: S_RCVRY_RCVRCFG ▪ 6'h10: S_RCVRY_IDLE ▪ 6'h11: S_L0 ▪ 6'h12: S_L0S ▪ 6'h13: S_L123_SEND_EIDLE ▪ 6'h14: S_L 1_1DLE ▪ 6'h15: S_L2_1DLE ▪ 6'h16: S_L2_WAKE ▪ 6'h17: S_DISABLED_ENTRY ▪ 6'h18: S_DISABLED_IDLE ▪ 6'h19: S_DISABLED ▪ 6'h1A: S_LPBK_ENTRY ▪ 6'h1B: S_LPBK_ACTIVE ▪ 6'h1C: S_LPBK_EXIT ▪ 6'h1D: S_LPBK_EXIT_TIMEOUT ▪ 6'h1E: S_HOT_RESET_ENTRY ▪ 6'h1F: S_HOT_RESET ▪ 6'h20: S_RCVRY_EQ0 ▪ 6'h21: S_RCVRY_EQ1 ▪ 6'h22: S_RCVRY_EQ2 ▪ 6'h23: S_RCVRY_EQ3 **Exists:** Always **Synchronous To:** core_clk **Registered:** N/A **Power Domain:** PD_VMAIN_SW **Active State:** High **Validated by:** Not validated by another signal
2:0	RO	0x0	**PM_CURNT_STATE**: DEBUG STATUS : Indicates the current power state. The pm_curnt_state output is intended for debugging purposes, not for system operation.

PCIE_RP_APPL_RADM_CPL_TO_0

Offset: 0xd4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,x00x)

Bit	Reset	Description
29:27	0x0	**REQ_FUNC_NUM**: The function Number of the timed out completion. Function numbering starts at '0'
26:24	0x0	**REQ_CPL_TC**: The Traffic Class of the timed out completion.
23:16	0x0	**REQ_CPL_TAG**: The Tag field of the timed out completion.
15:4	0x0	**REQ_CPL_LEN**: Length (in bytes) of the timed out completion. For a split completion, it indicates the number of bytes remaining to be delivered when the completion timed out.
2:1	0x0	**REQ_CPL_ATTR**: The Attributes field of the timed out completion.

PCIE_RP_APPL_RAS_5_0

Offset: 0xd8
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,0000,0000,0xxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
27:26	RW	0x0	**APP_RAS_DES_TBA_CTRL**: SELF-CLEAR : Controls the start/end of time based analysis. 2'b01: Start , 2'b10: End. This setting is only used when the TIME_BASED_DURATION_SELECT field of TIME_BASED_ANALYSIS_CONTROL_REG is set to "manual control". 2'b11: Reserved

Bit	R/W	Reset	Description
25:19	RO	0x0	**CDM_RAS_DES_TBA_INFO_COMMON**: DEBUG :Common event signal status bus used in RAS D.E.S. time based analysis. Indicates the internal signals that are used in the time-based analysis . The results are in TIME_BASED_ANALYSIS_DATA_REG. Each bit indicates the state that the controller stays in. All signals are level sensitive unless otherwise indicated.[6]: smlh_link_in_training: Config/Recovery,[5]: pm_in_l12: L1.2,[4]: pm_in_l11: L1.1,[3]: smlh_in_l1: L1,[2]: smlh_in_l0: L0,[1]: smlh_in_rl0s: L0s, [0]: smlh_in_l0s: L0s

PCIE_RP_APPL_CFG_STATUS0_0

Offset: 0xdc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,00x0,000x,xxxx,xxxx)

Bit	Reset	Description
15	0x0	**CFG_PWR_CTRLER_CTRL**: Controls the system power controller (from bit 10 of the Slot Control register), per function: 0: Power On 1: Power Off
14	0x0	**CFG_PM_NO_SOFT_RST**: This is the value of the No Soft Reset bit in the Power Management Control and Status Register. When set, you should not reset any core registers when transitioning from D3hot to D0. Therefore, you should not assert the non_sticky_rst_n and sticky_rst_n inputs.
12	0x0	**CFG_LTR_M_EN**: The LTR Mechanism Enable field of the Device Control 2 register of function 0.
11	0x0	**CFG_L1SUB_EN**: Indicates that any of the L1 Substates are enabled in the L1 Substates Control 1 Register. Could be used by your application in a downstream port to determine when not to drive CLKREQ# such as when L1 Substates are not enabled.
10	0x0	**CFG_HW_AUTO_SP_DIS**:
9	0x0	**CFG_BUS_MASTER_EN**: The state of the bus master enable bit in the PCI-compatible Command register. There is 1 bit of cfg_bus_master_en assigned to each configured function.

PCIE_RP_APPL_CFG_STATUS1_0

Offset: 0xe0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,0000,0000,0000,0000)

Bit	Reset	Description
19:12	0x0	**CFG_PBUS_NUM**: The primary bus number assigned to the function. The number of bits depends the value of MULTI_DEVICE_AND_BUS_PER_FUNC_EN: If MULTI_DEVICE_AND_BUS_PER_FUNC_EN =0, there are eight bits of cfg_pbus_num ([7:0]). If MULTI_DEVICE_AND_BUS_PER_FUNC_EN =1, there are eight bits of cfg_pbus_num.
11:7	0x0	**CFG_PBUS_DEV_NUM**: The device number assigned to the function. The number of bits depends the value of MULTI_DEVICE_AND_BUS_PER_FUNC_EN
6	0x0	**CFG_MEM_SPACE_EN**: The state of the Memory Space Enable bit in the PCI-compatible Command register. There is 1 bit of cfg_mem_space_en assigned
5:3	0x0	**CFG_MAX_RD_REQ_SIZE**: The value of the Max_Read_Request_Size field in the Device Control register. There are 3 bits of cfg_max_rd_req_size assigned.
2:0	0x0	**CFG_MAX_PAYLOAD_SIZE**: The value of the Max_Payload_Size field in the Device Control register. There are 3 bits of cfg_max_payload_size assigned.

PCIE_RP_APPL_RADM_STATUS0_0

Offset: 0xe4
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	R/W	Reset	Description
3	RO	0x0	**RADM_XFER_PENDING**: Receive request pending status. Indicates Receive TLP requests are pending, that is, requests sent to the RTRGT1 or RTRGT0 interfaces are awaiting a response from your application. For debugging purposes.
2	RO	0x0	**RADM_QOVERFLOW**: one or more of the P/NP/CPL receive queues have overflowed

Bit	R/W	Reset	Description
1	RO	0x0	**RADM_Q_NOT_EMPTY**: Level indicating that the receive queues contain TLP header/data
0	RW	0x0	**APPS_PM_XMT_TURNOFF_STATE**: SELF-CLEAR : Request from your application to generate a PM_Turn_Off message. The core does not return an acknowledgment or grant signal. You must not pulse the same signal again, until the previous message has been transmitted.

PCIE_RP_APPL_RADM_MSG_PAYLOAD0_0

Offset: 0xe8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**RADM_MSG_PAYLOAD0**: LSB of Lastest Payload received

PCIE_RP_APPL_RADM_MSG_PAYLOAD1_0

Offset: 0xec
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**RADM_MSG_PAYLOAD1**: MSB of Latest Payload received

PCIE_RP_APPL_VENDOR_MSG_PAYLOAD0_0

Offset: 0xf0
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VENDOR_MSG_PAYLOAD0**: LSB of Latest vendor message Payload recieved

PCIE_RP_APPL_VENDOR_MSG_PAYLOAD1_0

Offset: 0xf4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**VENDOR_MSG_PAYLOAD1**: MSB of Latest vendor message Payload received

PCIE_RP_APPL_MSG_REQ_ID_0

Offset: 0xf8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**VENDOR_MSG_REQ_ID**: REq_id when radm_completion_timeout == 1b1
15:0	0x0	**RADM_MSG_REQ_ID**: The requester ID of the received Message.

PCIE_RP_APPL_PM_STATUS_0

Offset: 0xfc
Read/Write: See table below
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	R/W	Reset	Description
2	RW	0x0	**APPS_PM_XMT_PME_STATE**: SELF-CLEAR : Wake Up. Used by application logic to wake up the PMC state machine from a D1, D2 or D3 power state. Upon wake-up, the core sends a PM_PME Message. There is a separate apps_pm_xmt_pme input bit for each function in your core configuration. This port is functionally identical to outband_pwrup_cmd. This signal or outband_pwrup_cmd them must be used to request a return from D1/ D2/D3 to D0/
1	RW	0x0	**APP_UNLOCK_MSG_STATE**: SELF-CLEAR : Request from your application to generate an Unlock message.The core does not return an acknowledgment or grant signal. You must not pulse the same signal again, until the previous message has been transmitted.
0	RO	0x0	**PM_PME_EN**: PME Enable bit in the PMCSR.

PCIE_RP_APPL_DM_TYPE_0

Offset: 0x100
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000004 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0100)

Bit	Reset	Description
3:0	0x4	**DEVICE_TYPE**: Mode of operation of the controller END_POINT=0x0, ROOT_PORT=0x4. Dual Mode controller only. **0** = END_POINT **4** = ROOT_PORT

PCIE_RP_APPL_CFG_BASE_ADDR_0

BASE_ADDR is required to perform address decodes to identify whether an outbound request is targeting external devices or the controller's configuration register space, including the shadow registers.

Offset: 0x104
Read/Write: R/W

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x10000000 (0b0001,0000,0000,0000,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:12	0x10000	**CFG_BASE_ADDR**: 8 KB region starting at this Address is allocated. Lower 4KB to access PCIE configuration Space of the device and upper 4KB for Shadow registers.

PCIE_RP_APPL_CFG_IATU_DMA_BASE_ADDR_0

BASE_ADDR is required to perform address decodes to identify whether an outbound request is targeting external devices or the controller's configuration register space, including the shadow registers.

Offset: 0x108
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,00xx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:18	0x0	**CFG_IATU_DMA_BASE_ADDR**: 256 KB region starting at this address would be directed to iATU/DMA regions of the Device. Refer : ARPCIE_IATU_DMA documentation for offsets

PCIE_RP_APPL_COH_IO_ACC_0

Offset: 0x10c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**COH_IO_ACC_OVERRIDE**: Coherency override value : Override value to be used when COH_IO_ACC_OVERRIDE_EN is 1. **0** = DIS **1** = EN
0	0x0	**COH_IO_ACC_OVERRIDE_EN**: Coherency override enable : If set the Snoop attributes on DBB will be overridden with COH_IO_ACC_OVERRIDE. **0** = DIS **1** = EN

PCIE_RP_APPL_CFG_MISC_0

Offset: 0x110
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000cc00 (0bxxxx,xxxx,xxxx,xxxx,1100,1100,00xx,0000)

Bit	Reset	Description
15	0x1	**CFG_SLV_MASK_ERR**: When enabled, Read data on AXI slave interface is set to all 1. when the corresponding slverr/decode error is asserted by PCIE controller core. **0** = DIS **1** = EN
14	0x1	**CFG_SLV_EP_MODE**: When enabled, Read data on AXI slave interface is set to all 1. when the corresponding completions have EP bit set. **0** = DIS **1** = EN
13:10	0x3	**CFG_ARCACHE**: CACHE attrributes for AXI interface for DBB. Indicates how the transactions are required to progress thru the system
9:6	0x0	**CFG_AWCACHE**: CACHE attrributes for AXI interface for DBB. Indicates how the transactions are required to progress thru the system
3	0x0	**CFG_WRITE_RO**: Outbound TLP attributes for CBB : TLP's RO bit
2	0x0	**CFG_WRITE_NS**: Outbound TLP attributes for CBB : TLP's NS bit
1	0x0	**CFG_READ_RO**: Outbound TLP attributes for CBB : TLP's RO bit

Bit	Reset	Description
0	0x0	**CFG_READ_NS**: Outbound TLP attributes for CBB : TLP's NS bit

PCIE_RP_APPL_CFG_SLCG_OVERRIDE_0

Offset: 0x114
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000022 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,0010)
PROD: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx0x,xx0x)

Bit	Reset	PROD	Description
5	0x1	0x0	**SLCG_EN_MISC_APBCLK**: CYA : Bypass for Second Level Clock Gating. If 1, corresponding clock will always be enabled. Not used.
4	0x0	_NONE_	**SLCG_EN_CTLR_AXISCLK**: CYA : Bypass for Second Level Clock Gating. If 1, corresponding clock will always be enabled. Clock gating for AXI bridge Slave interface to CBB
3	0x0	_NONE_	**SLCG_EN_CTLR_AXIMCLK**: CYA : Bypass for Second Level Clock Gating. If 1, corresponding clock will always be enabled. Clock gating for AXI bridge Master interface to DBB
2	0x0	_NONE_	**SLCG_EN_CTLR_RADMCLK**: CYA : Bypass for Second Level Clock Gating. If 1, corresponding clock will always be enabled. Clock gating for Rx data path
1	0x1	0x0	**SLCG_EN_CTLR_AUXCLK**: CYA : Bypass for Second Level Clock Gating. If 1, corresponding clock will always be enabled. Clock gating for controllers Application Logic register accessible via APB.
0	0x0	_NONE_	**SLCG_EN_MASTER**: CYA : Bypass for Second Level Clock Gating. Master Switch to Disable

PCIE_RP_APPL_CFG_PLL_0

Offset: 0x118
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	**DIS_REFPD_IN_L2**: CYA : PLL power down disable for corresponding Link State. If set PLLREF_PD will not be asserted in this Low power state and PLL will lead to higher power consumption.
4	0x0	**DIS_REFPD_IN_L1SUB**: CYA : PLL power down disable for corresponding Link State. If set PLLREF_PD will not be asserted in this Low power state and PLL will lead to higher power consumption.
3	0x0	**DIS_REFPD_IN_L1**: CYA : PLL power down disable for corresponding Link State. If set PLLREF_PD will not be asserted in this Low power state and PLL will lead to higher power consumption.
2	0x0	**DIS_PD_IN_L2**: CYA : PLL power down disable for corresponding Link State. If set PLL_PD will not be asserted in this Low power state and PLL will lead to higher power consumption.
1	0x0	**DIS_PD_IN_L1SUB**: CYA : PLL power down disable for corresponding Link State. If set PLL_PD will not be asserted in this Low power state and PLL will lead to higher power consumption.
0	0x0	**DIS_PD_IN_L1**: CYA : PLL power down disable for corresponding Link State. If set PLL_PD will not be asserted in this Low power state and PLL will lead to higher power consumption.

PCIE_RP_INTERRUPT_FAULT_OVERRIDE_0

Offset: 0x120
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,x000)

Bit	Reset	Description
7	0x0	**FORCE_INTD**: DEBUG : Force corresponging interrupt line for connectivity verification
6	0x0	**FORCE_INTC**: DEBUG : Force corresponging interrupt line for connectivity verification
5	0x0	**FORCE_INTB**: DEBUG : Force corresponging interrupt line for connectivity verification

Bit	Reset	Description
4	0x0	**FORCE_INTA**: DEBUG : Force corresponging interrupt line for connectivity verification
2	0x0	**FORCE_SYS_FAULT**: DEBUG : Force corresponging interrupt line for connectivity verification
1	0x0	**FORCE_SYS_MSI_INTR**: DEBUG : Force corresponging interrupt line for connectivity verification
0	0x0	**FORCE_SYS_INTR**: DEBUG : Force corresponging interrupt line for connectivity verification

PCIE_RP_STAGGER_WINDOW_TIME_0

Offset: 0x13c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,0000,0000,0000,0000)

Bit	Reset	Description
16	0x0	**DISABLE_OPTIONAL_STAGGERING**: Disable RX_CAL/RX_TRAIN_EN/RX_EOM_EN staggering. Un-used for PEX0 (VDD_SOC controllers)
15:0	0x0	**STAGGER_WINDOW_CNT**: count in TIMER clock (38.4 MHz, 26ns),value of N means N+1 cycle delay in HW, PEX1 stagger time window register. Un-used for PEX0 (VDD_SOC controllers)

PCIE_RP_CAR_RESET_PHY_CTRL_0

Offset: 0x140
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**APP_HOLD_PHY_RST**: Hold off Phy reset assertion.

PCIE_RP_PCIE_BUS_DEVICE_NUM_0

Offset: 0x144
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:8	0x0	**APP_DEV_NUM**: Device Number assigned to the port in RC Mode
7:0	0x0	**APP_BUS_NUM**: Bus Number assigned to the port in RC Mode

PCIE_RP_APPL_CFG_UPPER_BASE_ADDR_0

BASE_ADDR is required to perform address decodes to identify whether an outbound request is targeting external devices or the controller's configuration register space, including the shadow registers.

Offset: 0x148
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**CFG_UPPER_BASE_ADDR**: 8 KB region starting at this Address is allocated. Lower 4KB to access PCIE configuration Space of the device and upper 4KB for Shadow registers.

PCIE_RP_APPL_CFG_IATU_DMA_UPPER_BASE_ADDR_0

BASE_ADDR is required to perform address decodes to identify whether an outbound request is targeting external devices or the controller's configuration register space, including the shadow registers.

Offset: 0x14c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**CFG_IATU_DMA_UPPER_BASE_ADDR**: 256 KB region starting at this address would be directed to iATU/DMA regions of the Device. Refer : ARPCIE_IATU_DMA documentation for offsets

PCIE_RP_APPL_CONFIG_REGION_BASE_ADDR_0

BASE_ADDR is required to perform address decodes to identify whether an outbound request is targeting PCIE configuration address space.

Offset: 0x150
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:12	0x0	**CFG_CONFIG_REGION_BASE_ADDR**: If CFG_CONFIG_REGION_EN=1, 256 MB region starting at this address belongs to PCIE configuration space. Value of this field should be aligned to a 256 MB region. Upper bit of the base address are derived from CFG_CONFIG_REGION_UPPER_BASE_ADDR. Size of the region can be reduced by configuring CFG_CONFIG_REGION_BASE_LIMIT.

PCIE_RP_APPL_CFG_CONFIG_REGION_UPPER_BASE_ADDR_0

BASE_ADDR is required to perform address decodes to identify whether an outbound request is targeting PCIE configuration address space.

Offset: 0x154
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**CFG_CONFIG_REGION_UPPER_BASE_ADDR**: Upper bits of base address. Refer CFG_CONFIG_REGION_BASE_ADDR.

PCIE_RP_APPL_CFG_CONFIG_REGION_BASE_ADDR_0

BASE_LIMIT is required to perform address decodes to identify whether an outbound request is targeting PCIE configuration address space.

Offset: 0x158
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x6fffffff (0b011x,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31	0x0	**CFG_CONFIG_REGION_EN**: When set to 1, PCIE configuration region translation is enabled. This region controls the translation to Type0/Type1 CfgRd.
30	0x1	**CFG_CONFIG_ECAM_MODE**: When set to 1, 256 MB Address is translated to form {Bus,Dev,Func,4'b0,Register} as per PCIe specification.
29	0x1	**CFG_CONFIG_LIMIT_MODE**: When ECAM_MODE=0, set this bit to enable complete 4GB region of PCIE configuration space.
27:0	0xfffffff	**CFG_CONFIG_REGION_BASE_LIMIT**: Defines the size of PCIE Configuration to be translated should always be programmed to a value 2^N-1.

PCIE_RP_PCIE_MISC_0

Offset: 0x15c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**INT_SEGREGATION_EN**: When enabled, will route the interrupts to 4 separate lines
0	0x0	**CFG_5G_MODE**: When enabled, will route the inbound reads from link and DMA read requests on AXI to a dedicated low latency port. Applicable only for C5 controller only. Do not set for other controllers.

PCIE_RP_PTM_CTRL_0_0

Offset: 0x160
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PTM_EXTERNAL_MASTER_TIME_LSB**: When written to the value {PTM_EXTERNAL_MASTER_TIME_MSB,PTM_EXTERNAL_MASTER_TIME_LSB} is loaded as PTM local time.

PCIE_RP_PTM_CTRL_1_0

Offset: 0x164
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PTM_EXTERNAL_MASTER_TIME_MSB**: Refer PTM_EXTERNAL_MASTER_TIME_LSB

PCIE_RP_PTM_CTRL_2_0

Offset: 0x168
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000010 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0000)

Bit	Reset	Description
5:0	0x10	**PTM_REF_OUT_SEL**: selects the bit of the ptm_local_time to be sent out as ptm_ref_out. Only bits 10, 16-20 ,30 can be selected. any value other than these would select bit 16. **10** = PTM_BIT_1us **20** = PTM_BIT_1ms **30** = PTM_BIT_1s **16** = PTM_BIT_16 **17** = PTM_BIT_17 **18** = PTM_BIT_18 **19** = PTM_BIT_19

PCIE_RP_APPL_DEBUG_2_0

Offset: 0x16c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:8	0x0	**PM_SLAVE_STATE**: DEBUG STATUS : Power management slave FSM state.
7:3	0x0	**PM_MASTER_STATE**: DEBUG STATUS : Power management master FSM state.
2:0	0x0	**PM_L1SUB_STATE**: DEBUG STATUS : Power management L1 sub-states FSM state.

PCIE_RP_PERFMUX_CONTROL_0

Offset: 0x174
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x0	**PM_EN**: **0** = OFF **1** = ON
7:0	0x0	**PM_SEL**: **0** = READ_CLIENTS **1** = WRITE_CLIENTS **2** = READ_WRITE_CLIENTS **3** = STATIC_PATTERN_1 **4** = STATIC_PATTERN_2

PCIE_RP_PERFMUX_TRIGGER_0

Offset: 0x178
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**PM_STOP**: STOP engine trigger
0	0x0	**PM_START**: START engine trigger

PCIE_RP_APPL_FAULT_EN_L1_19_0

Offset: 0x17c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**CFG_SAFETY_CORR_STATE_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_INTR_STATUS_L1_19.

PCIE_RP_APPL_INTR_EN_L1_19_0

Offset: 0x180
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**CFG_SAFETY_CORR_STATE_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_19.

PCIE_RP_APPL_INTR_STATUS_L1_19_0

Offset: 0x184
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**CFG_SAFETY_CORR_STATE**: Status bit to indicate that there is an Error in safety correctable error status.

PCIE_RP_APPL_FAULT_EN_L1_20_0

Offset: 0x188
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**IF_TIMEOUT_STATUS_FAULT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_20.
0	0x0	**CFG_SAFETY_UNCORR_STATE_FAULT_EN**: Enable for fault generation for the corresponding bit in APPL_INTR_STATUS_L1_20.

PCIE_RP_APPL_INTR_EN_L1_20_0

Offset: 0x18c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**IF_TIMEOUT_STATUS_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_20.
0	0x0	**CFG_SAFETY_UNCORR_STATE_INT_EN**: Enable for interrupt generation for the corresponding bit in APPL_INTR_STATUS_L1_20.

PCIE_RP_APPL_INTR_STATUS_L1_20_0

Offset: 0x190
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**IF_TIMEOUT_STATUS**: Status bit to indicate that there is an Error and one of the interfaces has timed out.
0	0x0	**CFG_SAFETY_UNCORR_STATE**: Status bit to indicate that there is an Error in Safety uncorrectable status.

=== PCIE Secure Page Protected by CBB BLF ===

PCIE_RP_SEC_AXI_PROT_0

Base offset in APB device space (Separate 64kB page for SCR registers)

Offset: 0x10000
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000011 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1,xxx1)

Bit	Reset	Description
4	0x1	**CFG_SECURE_AWPROT**: This register bit drives axi_awprot[1] and wsb_ns on databackbone interface. 0=Secure access 1=Non-secure access **0** = SECURE **1** = NONSECURE
0	0x1	**CFG_SECURE_ARPROT**: This register bit drives axi_arprot[1] and rsb_ns on databackbone interface. 0=Secure access 1=Non-secure access **0** = SECURE **1** = NONSECURE

PCIE_RP_SEC_SRIOV_CTRL_0

PCIE Secure Register for NON MSI REQUSET ID to STREAM ID Mapping register

Offset: 0x10004
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000240 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx10,x100,0000)

Bit	Reset	Description
9	0x1	**MSI_USE_NONZERO_AWID**: If set to 1, MSI request use 0x1 as AWID while other Posted requests use 0x0. Disabling this bit will have performance implications.
8	0x0	**WAIT_MSI_RESP**: If set to 1, Response tracker would wait for MSI response from DBB before forwarding it to PCIe Core. Enabling this bit will have performance implications.
6:3	0x8	**NUM_MSI_OUTSTANG_AXI**: Number of MSI outstanding on AXI - Use cases are 8 and 1.

Bit	Reset	Description
2	0x0	**STRICT_P_ORDERING**: Enable Strict Posted ordering on AXI. Default disabled. 0-> MSI only wait for all Posted Writes issued before MSI. 1-> MSI request wait for proceeding MSI responses.
1	0x0	**MSI_REQID_2_STREAM_ID_ENABLE**: Enable MSI Requester ID to Stream ID mapping, default disabled. When enabled Stream ID is mapped to MSI_STREAM_ID_TARGET[i] if the received requester ID matches with the MSI_SOURCE_REQUESTER_ID<*i*> .
0	0x0	**REQID_2_STREAM_ID_ENABLE**: Enable Non-MSI Requester ID to Stream ID mapping, default disabled. It is applicable for both Reads and Non MSI Writes. When enabled Stream ID is mapped to STREAM_ID_TARGET<*i*> if the received requester ID matches with the SOURCE_REQUESTER_ID<*i*> .

PCIE_RP_SEC_SOURCE_REQUESTER_ID_0

PCIE Secure Register : REQUSETER ID to STREAM ID Mapping register (For Non MSI write and Read)

This is an array of 15 identical register entries; the register fields below apply to each entry.
Full register list is: PCIE_RP_SEC_SOURCE_REQUESTER_ID_<*i*>, among which <*i*> belongs to {0, 1, 2, … , 14}.

Offset: 0x10008,..,0x10040
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**REQUESTER_ID_MASK**: Non MSI Requester ID Mask Register - Mask is used while comparing the Received requester ID compared.
15:0	0x0	**SOURCE_REQUESTER_ID**: Non MSI Requester ID Register used for both Read and Write channel. There is separate MSI requester ID registers. Received requester ID compared the Requester ID table SOURCE_REQUESTER_ID<*i*>.

PCIE_RP_SEC_STREAM_ID_TARGET_0

This is an array of 15 identical register entries; the register fields below apply to each entry.
Full register list is: PCIE_RP_SEC_STREAM_ID_TARGET_<*i*>, among which <*i*> belongs to {0, 1, 2, ... , 14}.

Offset: 0x10044,..,0x1007c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**STREAM_ID_TARGET**: Non MSI Stream ID Target : Used for both read and write channels.

PCIE_RP_SEC_BASE_STREAM_ID_0

Offset: 0x10080
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**BASE_STREAM_ID**: Non MSI Base Stream ID : Used for both read and write channels. It is the default STREAM ID for read and write transactions in case there is no match for received requester ID.

PCIE_RP_SEC_MSI_SOURCE_REQUESTER_ID_0

PCIE Secure Register : MSI REQUSETER ID to STREAM ID Mapping register

This is an array of 15 identical register entries; the register fields below apply to each entry.
Full register list is: PCIE_RP_SEC_MSI_SOURCE_REQUESTER_ID_<*i*>, among which <*i*> belongs to {0, 1, 2, ... , 14}.

Offset: 0x10084,..,0x100bc
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**MSI_REQUESTER_ID_MASK**: MSI Requester ID Mask Register
15:0	0x0	**MSI_SOURCE_REQUESTER_ID**: MSI Requester ID Mask Register: Used only for the MSI transactions Received requester ID compared the Requester ID table MSI_SOURCE_REQUESTER_ID<*i*>

PCIE_RP_SEC_MSI_STREAM_ID_TARGET_0

This is an array of 15 identical register entries; the register fields below apply to each entry.
Full register list is: PCIE_RP_SEC_MSI_STREAM_ID_TARGET_<*i*>, among which <*i*> belongs to {0, 1, 2, ... , 14}.

Offset: 0x100c0,..,0x100f8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**MSI_STREAM_ID_TARGET**: Non MSI Stream ID Target: Used only for the MSI transactions

PCIE_RP_SEC_MSI_BASE_STREAM_ID_0

Offset: 0x100fc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**MSI_BASE_STREAM_ID**: MSI Base Stream ID: Used only for the MSI transactions. It is the default STREAM ID for MSI in case there is no match for received requester ID.

PCIE_RP_SEC_EXTERNAL_MSI_ADDR_H_0

PCIE Secure Register: MSI Address Register for External MSI (Received MSI)

Offset: 0x10100
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EXTERNAL_MSI_ADDR_H**: External MSI Address upper 32 bits, Non zero value would enabled the MSI address translation. Used for MSI Decode: Received Write Address [63:2] is compare with {EXTERNAL_MSI_ADDR_H, EXTERNAL_MSI_ADDR_L}.

PCIE_RP_SEC_EXTERNAL_MSI_ADDR_L_0

Offset: 0x10104
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31:2	0x0	**EXTERNAL_MSI_ADDR_L**: External MSI Address lower 30 bits, Non zero value would enabled the MSI address translation. Used for MSI Decode: Received Write Address [63:2] is compare with {EXTERNAL_MSI_ADDR_H, EXTERNAL_MSI_ADDR_L}.

PCIE_RP_SEC_INTERNAL_MSI_ADDR_H_0

PCIE Secure Register: MSI Address Translation Registers

Offset: 0x10108
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**INTERNAL_MSI_ADDR_H**: Internal MSI Address upper[39:26] bits , Non zero value would enabled the MSI address translation. MSI address [39:32] is translated with INTERNAL_MSI_ADDR_H. Internal_MSI_Address[39:26] = {INTERNAL_MSI_ADDR_H, INTERNAL_MSI_ADDR_L}. MSi Translated address = {Internal_MSI_Address[39:26], Received_Data[9:0], 16'b0000_0000_0000_0000}.

PCIE_RP_SEC_INTERNAL_MSI_ADDR_L_0

Offset: 0x1010c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,00xx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31:26	0x0	**INTERNAL_MSI_ADDR_L**: Internal MSI Address upper[39:26] bits , Non zero value would enabled the MSI address translation. MSI address [31:26] is translated with INTERNAL_MSI_ADDR_L. Internal_MSI_Address[39:26] = {INTERNAL_MSI_ADDR_H, INTERNAL_MSI_ADDR_L}. MSi Translated address = {Internal_MSI_Address[39:26], Received_Data[9:0], 16'b0000_0000_0000_0000}.

9.3.5.3 PCIe x1/x4/x8 Root Complex

For a description of these standard PCIe register fields, see the PCI Express Specification.

NOTE:

There are 3 instances for each of the "X1" registers as listed, one for each of the X1 PCIe module instances, namely C1, C2, and C3.
There are 6 instances for each of the "X4" registers as listed, one for each of the X4 PCIe module instances, namely C0, C4, C6, C8, C9, and C10.
And there are 2 instances for each of the "X8" registers as listed, one for each of the X8 PCIe module instances, namely C5 and C7.

For the base addresses of these different register instances, please refer to the System Address Map.

NOTE:

The binary “x” is turned to “0” for the Reset and PROD hex value. Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as “x” or “Undefined” actually means that the bit could be either “0” or “1” after Reset, whereas a bit PROD value as “x” or “Don’t-care” means that either “0” or “1” can be written to the bit by software Initialization.

PCIE_X<*i*>_RC_PF0_TYPE1_HDR_TYPE1_DEV_ID_VEND_ID_REG_0

where <*i*> = 1, 4, 8.

Description: This register holds the device ID and vendor ID.

Offset: 0x0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_TYPE1_HDR_TYPE1_DEV_ID_VEND_ID_REG_0

Reset: 0x229e10de (0b0010,0010,1001,1110,0001,0000,1101,1110)

Bit	Reset	Description
31:16	0x229e	**DEVICE_ID**: Device ID. The Device ID register identifies the particular Function. This identifier is allocated by the vendor. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x10de	**VENDOR_ID**: Vendor ID. The Vendor ID register identifies the manufacturer of the Function. Valid vendor identifiers are allocated by the PCI-SIG to ensure uniqueness. It is not permitted to populate this register with a value of FFFFh, which is an invalid value for Vendor ID. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X4_RC_PF0_TYPE1_HDR_TYPE1_DEV_ID_VEND_ID_REG_0

Reset: 0x229c10de (0b0010,0010,1001,1100,0001,0000,1101,1110)

Bit	Reset	Description
31:16	0x229c	**DEVICE_ID**: Device ID. The Device ID register identifies the particular Function. This identifier is allocated by the vendor. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x10de	**VENDOR_ID**: Vendor ID. The Vendor ID register identifies the manufacturer of the Function. Valid vendor identifiers are allocated by the PCI-SIG to ensure uniqueness. It is not permitted to populate this register with a value of FFFFh, which is an invalid value for Vendor ID. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X8_RC_PF0_TYPE1_HDR_TYPE1_DEV_ID_VEND_ID_REG_0

Reset: 0x229a10de (0b0010,0010,1001,1010,0001,0000,1101,1110)

Bit	Reset	Description
31:16	0x229a	**DEVICE_ID**: Device ID. The Device ID register identifies the particular Function. This identifier is allocated by the vendor. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x10de	**VENDOR_ID**: Vendor ID. The Vendor ID register identifies the manufacturer of the Function. Valid vendor identifiers are allocated by the PCI-SIG to ensure uniqueness. It is not permitted to populate this register with a value of FFFFh, which is an invalid value for Vendor ID. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_TYPE1_HDR_TYPE1_STATUS_COMMAND_REG_0

where <*i*> = 1, 4, 8.

Description: This register provides the status and controls the behavior of a function.

PCIE_X1_RC_PF0_TYPE1_HDR_TYPE1_STATUS_COMMAND_REG_0

PCIE_X4_RC_PF0_TYPE1_HDR_TYPE1_STATUS_COMMAND_REG_0

PCIE_X8_RC_PF0_TYPE1_HDR_TYPE1_STATUS_COMMAND_REG_0

Offset: 0x4
Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00100000 (0b0000,0000,0001,000x,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**DETECTED_PARITY_ERROR**: Detected Parity Error. The bit is set when the Poisoned TLP is received by a Function's primary side.
30	RW	0x0	**SIGNALED_SYS_ERROR**: Signaled System Error.
29	RW	0x0	**RCVD_MASTER_ABORT**: Received Master Abort. This bit is set when a Requester receives a Completion with Unsupported Request Completion status. The bit is set when the Unsupported Request is received by a Function's primary side.
28	RW	0x0	**RCVD_TARGET_ABORT**: Received Target Abort. This bit is set when a Requester receives a Completion with Completer Abort Completion status. The bit is set when the Completer Abort is received by a Function's primary side.
27	RW	0x0	**SIGNALED_TARGET_ABORT**: Signaled Target Abort. This bit is set when a Function completes a Posted or Non-Posted Request as a Completer Abort error. This applies to a Function when the Completer Abort was generated by its primary side.
26:25	RO	0x0	**DEV_SEL_TIMING**: DEVSEL Timing. This field was originally described in the PCI Local Bus Specification. Its functionality does not apply to PCI Express. The controller hardwires it to 00b.
24	RW	0x0	**MASTER_DPE**: Master Data Parity Error. This bit is set by a Function if the Parity Error Response bit in the Command register is 1b and either of the following two conditions occurs: - Port receives a Poisoned Completion going downstream - Port transmits a Poisoned Request upstream If the Parity Error Response bit is 0b, this bit is never set.
23	RO	0x0	**FAST_B2B_CAP**: Fast Back-to-Back Transactions Capable. This bit was originally described in the PCI Local Bus Specification. Its functionality does not apply to PCI Express. The controller hardwires this bit to 0b.
22	RO	0x0	**RSVDP_22**: Reserved for future use.
21	RO	0x0	**FAST_66MHZ_CAP**: 66 MHz Capable. This bit was originally described in the PCI Local Bus Specification. Its functionality does not apply to PCI Express. The controller hardwires this bit to 0b.
20	RO	0x1	**CAP_LIST**: Capabilities List. Indicates the presence of an Extended Capability list item. Since all PCI Express device Functions are required to implement the PCI Express Capability structure, the controller hardwires this bit to 1b.

Bit	R/W	Reset	Description
19	RO	0x0	**INT_STATUS**: Interrupt Status. INTx emulation interrupts forwarded by Functions from the secondary side are not reflected in this bit. Setting the Interrupt Disable bit has no effect on the state of this bit. For Functions that do not generate INTx interrupts, the controller hardwires this bit to 0b.
18:17	RO	0x0	**RSVDP_17**: Reserved for future use.
15:11	RO	0x0	**RESERV**: Reserved.
10	RW	0x0	**INT_EN**: Interrupt Disable. Controls the ability of a Function to generate INTx emulation interrupts. Note: - Any INTx emulation interrupts already asserted by the Function must be deasserted when this bit is set. INTx interrupts use virtual wires that must, if asserted, be deasserted using the appropriate Deassert_INTx message(s) when this bit is set. - Only the INTx virtual wire interrupt(s) associated with the Function(s) for which this bit is set are affected. - For Functions that generate INTx interrupts on their own behalf, this bit is required. This bit has no effect on interrupts forwarded from the secondary side. For Functions that do not generate INTx interrupts on their own behalf this bit is optional. If this bit is not implemented, the controller hardwires it to 0b.
9	RO	0x0	**RSVDP_9**: Reserved for future use.
8	RW	0x0	**SERREN**: SERR# Enable. Note: - The errors are reported if enabled either through this bit or through the PCI Express specific bits in the Device Control register. - In addition, this bit controls transmission by the primary interface of ERR_NONFATAL and ERR_FATAL error Messages forwarded from the secondary interface. This bit does not affect the transmission of forwarded ERR_COR messages.
7	RO	0x0	**IDSEL**: IDSEL Stepping/Wait Cycle Control. This bit was originally described in the PCI Local Bus Specification. Its functionality does not apply to PCI Express. The controller hardwires this bit to 0b.
6	RW	0x0	**PERREN**: Parity Error Response. This bit controls the logging of poisoned TLPs in the Master Data Parity Error bit in the Status register.
5	RO	0x0	**VGAPS**: VGA Palette Snoop. This bit was originally described in the PCI Local Bus Specification and the PCI-to-PCI Bridge Architecture Specification. Its functionality does not apply to PCI Express. The controller hardwires this bit to 0b.
4	RO	0x0	**MWI_EN**: Memory Write and Invalidate. This bit was originally described in the PCI Local Bus Specification and the PCI-to-PCI Bridge Architecture Specification. Its functionality does not apply to PCI Express. The controller hardwires this bit to 0b. For PCI Express to PCI/PCI-X Bridges, refer to the PCI Express to PCI/PCI-X Bridge Specification for requirements for this register.

Bit	R/W	Reset	Description
3	RO	0x0	**SCO**: Special Cycle Enable. This bit was originally described in the PCI Local Bus Specification. Its functionality does not apply to PCI Express. The controller hardwires this bit to 0b.
2	RW	0x0	**BME**: Bus Master Enable. This bit controls forwarding of Memory or I/O requests by a port in the Upstream direction. When this bit is 0b, Memory and I/O Requests received at a Root Port must be handled as Unsupported Requests (UR) For Non-Posted Requests a Completion with UR completion status must be returned. This bit does not affect forwarding of Completions in either the Upstream or Downstream direction. The forwarding of Requests other than Memory or I/O Requests is not controlled by this bit.
1	RW	0x0	**MSE**: Memory Space Enable. This bit controls a Function's response to Memory Space accesses received on its primary side. You cannot write to this register if your configuration has no MEM BARs; that is, the internal signal has_mem_bar =0. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: !has_mem_bar ? RO : RW
0	RW	0x0	**IO_EN**: IO Space Enable. This bit controls a Function's response to I/O Space accesses received on its primary side. You cannot write to this register if your configuration has no IO BARs; that is, the internal signal has_io_bar =0. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: !has_io_bar ? RO : RW

PCIE_X<*i*>_RC_PF0_TYPE1_HDR_TYPE1_CLASS_CODE_REV_ID_REG_0

where <*i*> = 1, 4, 8.

Description: This register specifies the class code and revision ID of a function.

PCIE_X1_RC_PF0_TYPE1_HDR_TYPE1_CLASS_CODE_REV_ID_REG_0

PCIE_X4_RC_PF0_TYPE1_HDR_TYPE1_CLASS_CODE_REV_ID_REG_0

PCIE_X8_RC_PF0_TYPE1_HDR_TYPE1_CLASS_CODE_REV_ID_REG_0

Offset: 0x8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x060400a1 (0b0000,0110,0000,0100,0000,0000,1010,0001)

Bit	Reset	Description
31:24	0x6	**BASE_CLASS_CODE**: Base Class Code. A code that broadly classifies the type of operation the Function performs. Encodings for base class, are provided in the PCI Code and ID Assignment Specification. All unspecified encodings are reserved. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
23:16	0x4	**SUBCLASS_CODE**: Sub-Class Code. Specifies a base class sub-class, which identifies more specifically the operation of the Function. Encodings for sub-class are provided in the PCI Code and ID Assignment Specification. All unspecified encodings are reserved. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:8	0x0	**PROGRAM_INTERFACE**: Programming Interface. This field identifies a specific register level programming interface (if any) so that device independent software can interact with the Function. Encodings for interface are provided in the PCI Code and ID Assignment Specification. All unspecified encodings are reserved. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
7:0	0xa1	**REVISION_ID**: Revision ID. The value of this field specifies a Function specific revision identifier. The value is chosen by the vendor. Zero is an acceptable value. The Revision ID should be viewed as a vendor defined extension to the Device ID. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_TYPE1_HDR_TYPE1_BIST_HDR_TYPE_LAT_CACHE_LINE_SIZE_REG_0

where <*i*> = 1, 4, 8.

Description: This register provides the status and controls BIST. It also holds information regarding the header layout, latency timer, and cache line size.

PCIE_X1_RC_PF0_TYPE1_HDR_TYPE1_BIST_HDR_TYPE_LAT_CACHE_LINE_SIZE_REG_0

PCIE_X4_RC_PF0_TYPE1_HDR_TYPE1_BIST_HDR_TYPE_LAT_CACHE_LINE_SIZE_REG_0

PCIE_X8_RC_PF0_TYPE1_HDR_TYPE1_BIST_HDR_TYPE_LAT_CACHE_LINE_SIZE_REG_0

Offset: 0xc
Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00010000 (0b0000,0000,0000,0001,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**BIST**: BIST. This register is used for control and status of BIST. Functions that do not support BIST must hardwire the register to 00h. A Function whose BIST is invoked must not prevent normal operation of the PCI Express Link. Bit descriptions: - [31]: BIST Capable. When set, this bit indicates that the Function supports BIST. When Clear, the Function does not support BIST. - [30]: Start BIST. If BIST Capable is set, set this bit to invoke BIST. The Function resets the bit when BIST is complete. Software is permitted to fail the device if this bit is not Clear (BIST is not complete) 2 seconds after it had been set. Writing this bit to 0b has no effect. The controller hardwires this bit to 0b if BIST Capable is clear. - [29:28]: Reserved. - [27:24]: Completion Code. This field encodes the status of the most recent test. A value of 0000b means that the Function has passed its test. Non-zero values mean the Function failed. Function-specific failure codes can be encoded in the non-zero values. This field's value is only meaningful when BIST Capable is set and Start BIST is Clear. This field must be hardwired to 0000b if BIST Capable is clear.
23	RO	0x0	**MULTI_FUNC**: Multi-Function Device. Except where stated otherwise, it is recommended that this bit be set if there are multiple Functions, and clear if there is only one Function. Note: This register field is sticky.
22:16	RO	0x1	**HEADER_TYPE**: Header Layout. This field identifies the layout of the second part of the predefined header. The controller uses 000 0001b encoding. The encoding 000 0010b is reserved. This encoding was originally described in the PC Card Standard Electrical Specification and is used in previous versions of the programming model. Careful consideration should be given to any attempt to re-purpose it.
15:8	RO	0x0	**LATENCY_MASTER_TIMER**: Latency Timer. This register is also referred to as Primary Latency Timer. The Latency Timer was originally described in the PCI Local Bus Specification and the PCI-to-PCI Bridge Architecture Specification. Its functionality does not apply to PCI Express. The controller hardwires this register to 00h.
7:0	RW	0x0	**CACHE_LINE_SIZE**: Cache Line Size. The Cache Line Size register is programmed by the system firmware or the operating system to system cache line size. However, legacy conventional PCI software may not always be able to program this register correctly especially in the case of Hot-Plug devices. This read-write register is implemented for legacy compatibility purposes but has no effect on any PCI Express device behavior.

PCIE_X<*i*>_RC_PF0_TYPE1_HDR_BAR0_REG_0

where <*i*> = 1, 4, 8.

Description: System software must build a consistent address map before booting the machine to an operating system. This means it has to determine how much memory is in the system, and how much address space the Functions in the system require. After determining this information,

system software can map the Functions into reasonable locations and proceed with system boot. In order to do this mapping in a device-independent manner, the base registers for this mapping are placed in the predefined header portion of Configuration Space. It is strongly recommended that power-up firmware/software also support the optional Enhanced Configuration Access Mechanism (ECAM).

Offset: 0x10
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_TYPE1_HDR_BAR0_REG_0

Read/Write: See table below
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:4	RW	0x0	**BAR0_START**: BAR0 Base Address. - Memory Space: Base Address. - IO Space: bits[31:2] are used to map the function into IO space/Base. Address. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R(Sticky)/W(Sticky) if enabled else R(Sticky) Note: This register field is sticky.
3	RO	0x0	**BAR0_PREFETCH**: BAR0 Prefetchable. - Memory Space: Set to one if data is prefetchable. A Function is permitted to mark a range as prefetchable. If there are no side effects on reads, the function returns all bytes on reads regardless of the byte enables, and host bridges can merge processor writes into this range without causing errors. Bit must me clear otherwise. - IO Space: Not applicable Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.
2:1	RO	0x0	**BAR0_TYPE**: BAR0 Type. - Memory Space: Base Address registers that map into Memory Space can be 32 bits or 64 bits wide (to support mapping into a 64-bit address space). The encodings defined in Values: apply. - IO Space: Bit 1 is reserved and must return 0b on reads. Bits[31:2] are used to map the function into IO space. The encodings defined in Values: do not apply. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.
0	RO	0x0	**BAR0_MEM_IO**: BAR0 Memory Space Indicator. This bit is used to determine whether the register maps into Memory or I/O Space. Base Address registers that map to Memory Space must return a 0b. Base Address registers that map to I/O Space must return a 1b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.

PCIE_X4_RC_PF0_TYPE1_HDR_BAR0_REG_0

PCIE_X8_RC_PF0_TYPE1_HDR_BAR0_REG_0

Read/Write: R/W
Reset: 0x0000000c (0b0000,0000,0000,0000,0000,0000,0000,1100)

Bit	Reset	Description
31:4	0x0	**BAR0_START**: BAR0 Base Address. - Memory Space: Base Address. - IO Space: bits[31:2] are used to map the function into IO space/Base. Address. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R(Sticky)/W(Sticky) if enabled else R(Sticky) Note: This register field is sticky.
3	0x1	**BAR0_PREFETCH**: BAR0 Prefetchable. - Memory Space: Set to one if data is prefetchable. A Function is permitted to mark a range as prefetchable. If there are no side effects on reads, the function returns all bytes on reads regardless of the byte enables, and host bridges can merge processor writes into this range without causing errors. Bit must me clear otherwise. - IO Space: Not applicable Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.
2:1	0x2	**BAR0_TYPE**: BAR0 Type. - Memory Space: Base Address registers that map into Memory Space can be 32 bits or 64 bits wide (to support mapping into a 64-bit address space). The encodings defined in Values: apply. - IO Space: Bit 1 is reserved and must return 0b on reads. Bits[31:2] are used to map the function into IO space. The encodings defined in Values: do not apply. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.
0	0x0	**BAR0_MEM_IO**: BAR0 Memory Space Indicator. This bit is used to determine whether the register maps into Memory or I/O Space. Base Address registers that map to Memory Space must return a 0b. Base Address registers that map to I/O Space must return a 1b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_TYPE1_HDR_BAR1_REG_0

where <*i*> = 1, 4, 8.

Description: System software must build a consistent address map before booting the machine to an operating system. This means it has to determine how much memory is in the system, and how much address space the Functions in the system require. After determining this information, system software can map the Functions into reasonable locations and proceed with system boot. In order to do this mapping in a device-independent manner, the base registers for this mapping are placed in the predefined header portion of Configuration Space. It is strongly recommended that power-up firmware/software also support the optional Enhanced Configuration Access Mechanism (ECAM).

Offset: 0x14
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

PCIE_X1_RC_PF0_TYPE1_HDR_BAR1_REG_0

Read/Write: See table below

Bit	R/W	Reset	Description
31:4	RW	0x0	**BAR1_START**: BAR1 Base Address. - Memory Space: Base Address. - IO Space: bits[31:2] are used to map the function into IO space/Base. Address. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R(Sticky)/W(Sticky) if enabled else R(Sticky) Note: This register field is sticky.
3	RO	0x0	**BAR1_PREFETCH**: BAR1 Prefetchable. - Memory Space: Set to one if data is prefetchable. A Function is permitted to mark a range as prefetchable. If there are no side effects on reads, the function returns all bytes on reads regardless of the byte enables, and host bridges can merge processor writes into this range without causing errors. Bit must me clear otherwise. - IO Space: Not applicable Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.
2:1	RO	0x0	**BAR1_TYPE**: BAR1 Type. - Memory Space: Base Address registers that map into Memory Space can be 32 bits or 64 bits wide (to support mapping into a 64-bit address space). The encodings defined in Values: apply. - IO Space: Bit 1 is reserved and must return 0b on reads. Bits[31:2] are used to map the function into IO space. The encodings defined in Values: do not apply. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.
0	RO	0x0	**BAR1_MEM_IO**: BAR1 Memory Space Indicator. This bit is used to determine whether the register maps into Memory or I/O Space. Base Address registers that map to Memory Space must return a 0b. Base Address registers that map to I/O Space must return a 1b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.

PCIE_X4_RC_PF0_TYPE1_HDR_BAR1_REG_0

PCIE_X8_RC_PF0_TYPE1_HDR_BAR1_REG_0

Read/Write: R/W

Bit	Reset	Description
31:4	0x0	**BAR1_START**: BAR1 Base Address. - Memory Space: Base Address. - IO Space: bits[31:2] are used to map the function into IO space/Base. Address. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R(Sticky)/W(Sticky) if enabled else R(Sticky) Note: This register field is sticky.
3	0x0	**BAR1_PREFETCH**: BAR1 Prefetchable. - Memory Space: Set to one if data is prefetchable. A Function is permitted to mark a range as prefetchable. If there are no side effects on reads, the function returns all bytes on reads regardless of the byte enables, and host bridges can merge processor writes into this range without causing errors. Bit must me clear otherwise. - IO Space: Not applicable Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.
2:1	0x0	**BAR1_TYPE**: BAR1 Type. - Memory Space: Base Address registers that map into Memory Space can be 32 bits or 64 bits wide (to support mapping into a 64-bit address space). The encodings defined in Values: apply. - IO Space: Bit 1 is reserved and must return 0b on reads. Bits[31:2] are used to map the function into IO space. The encodings defined in Values: do not apply. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.
0	0x0	**BAR1_MEM_IO**: BAR1 Memory Space Indicator. This bit is used to determine whether the register maps into Memory or I/O Space. Base Address registers that map to Memory Space must return a 0b. Base Address registers that map to I/O Space must return a 1b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_TYPE1_HDR_SEC_LAT_TIMER_SUB_BUS_SEC_BUS_PRI_BUS_REG_0

where <*i*> = 1, 4, 8.

Description: This register holds information regarding secondary latency timer, subordinate bus number, secondary bus number, and primary bus number.

PCIE_X1_RC_PF0_TYPE1_HDR_SEC_LAT_TIMER_SUB_BUS_SEC_BUS_PRI_BUS_REG_0

PCIE_X4_RC_PF0_TYPE1_HDR_SEC_LAT_TIMER_SUB_BUS_SEC_BUS_PRI_BUS_REG_0

PCIE_X8_RC_PF0_TYPE1_HDR_SEC_LAT_TIMER_SUB_BUS_SEC_BUS_PRI_BUS_REG_0

Offset: 0x18
Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**SEC_LAT_TIMER**: Secondary Latency Timer. This register does not apply to PCI Express. The controller hardwires it to 00h.
23:16	RW	0x0	**SUB_BUS**: Subordinate Bus Number. The Subordinate Bus Number register is used to record the bus number of the highest numbered PCI bus segment which is behind (or subordinate to) the bridge. Configuration software programs the value in this register. The bridge uses this register in conjunction with the Secondary Bus Number register to determine when to respond to and pass on a Type 1 configuration transaction on the primary interface to the secondary interface.
15:8	RW	0x0	**SEC_BUS**: Secondary Bus Number. The Secondary Bus Number register is used to record the bus number of the PCI bus segment to which the secondary interface of the bridge is connected. Configuration software programs the value in this register. The bridge uses this register to determine when to respond to and convert a Type 1 configuration transaction on the primary interface into a Type 0 transaction on the secondary interface.
7:0	RW	0x0	**PRIM_BUS**: Primary Bus Number. This register is not used by PCI Express Functions. It is implemented for compatibility with legacy software.

PCIE_X<*i*>_RC_PF0_TYPE1_HDR_SEC_STAT_IO_LIMIT_IO_BASE_REG_0

where <*i*> = 1, 4, 8.

Description: The I/O Limit and I/O Base registers are optional and define an address range that is used by the bridge to determine when to forward I/O transactions from one interface to the other. If a bridge does not implement an I/O address range, then both the I/O Limit and I/O Base registers must be implemented as read-only registers that return zero when read. If a bridge supports an I/O address range, then these registers must be initialized by configuration software so default states are not specified.

PCIE_X1_RC_PF0_TYPE1_HDR_SEC_STAT_IO_LIMIT_IO_BASE_REG_0

PCIE_X4_RC_PF0_TYPE1_HDR_SEC_STAT_IO_LIMIT_IO_BASE_REG_0

PCIE_X8_RC_PF0_TYPE1_HDR_SEC_STAT_IO_LIMIT_IO_BASE_REG_0

Offset: 0x1c
Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000101 (0b0000,0000,0000,0000,0000,0001,0000,0001)

Bit	R/W	Reset	Description
31	RW	0x0	**SEC_STAT_DPE**: Detected Parity Error. This bit is set by a Function when a Poisoned TLP is received by its secondary side, regardless of the state the Parity Error Response Enable bit in the Bridge Control register.
30	RW	0x0	**SEC_STAT_RCVD_SYS_ERR**: Received System Error. This bit is set when the secondary side of a Function receives an ERR_FATAL or ERR_NONFATAL message.
29	RW	0x0	**SEC_STAT_RCVD_MSTR_ABRT**: Received Master Abort. This bit is set when the secondary side of a Function (for requests initiated by the Type 1 header Function itself) receives a Completion with Unsupported Request Completion status.
28	RW	0x0	**SEC_STAT_RCVD_TRGT_ABRT**: Received Target Abort. This bit is set when the secondary side of a Function (for requests initiated by the Type 1 header Function itself) receives a Completion with Completer Abort Completion status.
27	RW	0x0	**SEC_STAT_SIG_TRGT_ABRT**: Signaled Target Abort. This bit is set when the secondary side of the Function (for Requests completed by the Type 1 header Function itself) completes a Posted or Non-Posted request as a Completer Abort error.
26:25	RO	0x0	**RSVDP_25**: Reserved for future use.
24	RW	0x0	**SEC_STAT_MDPE**: Master Data Parity Error. This bit is set by a Function if the Parity Error Response Enable bit in the Bridge Control register is set, and either of the following two conditions occurs: - Port receives a Poisoned Completion coming Upstream - Port transmits a Poisoned Request Downstream If the Parity Error Response Enable bit is clear, this bit is never set.
23	RO	0x0	**RSVDP_23**: Reserved for future use.
22:16	RO	0x0	**SEC_STAT_RESERV**: Reserved.
15:12	RW	0x0	**IO_LIMIT**: I/O Limit Address. These bits correspond to the address[15:12] of IO address range. For the purpose of address decoding, the bridge assumes that the lower 12 address bits, address[11:0], of the I/O limit address (not implemented in the I/O Limit register) are FFFh. The I/O Limit register can be programmed to a smaller value than the I/O Base register, if there are no I/O addresses on the secondary side of the bridge. In this case, the bridge will not forward any I/O transactions from the primary bus to the secondary and will forward all I/O transactions from the secondary bus to the primary bus.
11:9	RO	0x0	**IO_RESERV1**: Reserved.

Bit	R/W	Reset	Description
8	RW	0x0	**IO_DECODE_BIT8**: I/O Addressing Encode (IO Limit Address). This bit encodes the IO addressing capability of the bridge. Note: - For 16bit I/O address decoding, the bridge must still perform a full 32-bit decode of the I/O address (that is, check that Address[31:16] are 0000h). In this case, the I/O address range supported by the bridge will be restricted to the first 64 KB of I/O Space (0000 0000h to 0000 FFFFh). - For 16bit I/O address decoding, the bridge assumes that the upper 16 address bits, Address[31:16], of the I/O limit address (not implemented in I/O Limit register) are zero. - For 32bit I/O address decoding, and the I/O Limit Upper 16 Bits hold the upper 16 bits, corresponding to Address[31:16], of the 32-bit Limit address. In this case, system configuration software is permitted to locate the I/O address range supported by the bridge anywhere in the 4-GB I/O Space. - The 4-KB alignment and granularity restrictions still apply when the bridge supports 32-bit I/O addressing. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R
7:4	RW	0x0	**IO_BASE**: I/O Base Address. These bits correspond to the address[15:12] of IO address range. For the purpose of address decoding, the bridge assumes that the lower 12 address bits, address[11:0], of the I/O base address (not implemented in the I/O Base register) are zero.
3:1	RO	0x0	**IO_RESERV**: Reserved.
0	RW	0x0	**IO_DECODE**: I/O Addressing Encode (IO Base Address). This bit encodes the IO addressing capability of the bridge. Note: - For 16bit I/O address decoding, the bridge must still perform a full 32-bit decode of the I/O address (that is, check that Address[31:16] are 0000h). In this case, the I/O address range supported by the bridge will be restricted to the first 64 KB of I/O Space (0000 0000h to 0000 FFFFh). - For 16bit I/O address decoding, the bridge assumes that the upper 16 address bits, Address[31:16], of the I/O limit address (not implemented in I/O Limit register) are zero. - For 32bit I/O address decoding, and the I/O Limit Upper 16 Bits hold the upper 16 bits, corresponding to Address[31:16], of the 32-bit Limit address. In this case, system configuration software is permitted to locate the I/O address range supported by the bridge anywhere in the 4-GB I/O Space. - The 4-KB alignment and granularity restrictions still apply when the bridge supports 32-bit I/O addressing. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R

PCIE_X<*i*>_RC_PF0_TYPE1_HDR_MEM_LIMIT_MEM_BASE_REG_0

where <*i*> = 1, 4, 8.

Description: The Memory Limit and Memory Base registers define a memory mapped address range which is used by the bridge to determine when to forward memory transactions from one interface to the other. If there is no prefetchable memory space, and there is no memory-mapped space on the secondary side of the bridge, then the bridge will not forward any memory transactions from the primary bus to the secondary bus and will forward all memory transactions from the secondary bus to the primary bus.

PCIE_X1_RC_PF0_TYPE1_HDR_MEM_LIMIT_MEM_BASE_REG_0

PCIE_X4_RC_PF0_TYPE1_HDR_MEM_LIMIT_MEM_BASE_REG_0

PCIE_X8_RC_PF0_TYPE1_HDR_MEM_LIMIT_MEM_BASE_REG_0

Offset: 0x20
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:20	RW	0x0	**MEM_LIMIT**: Memory Limit Address. These bits correspond to the upper 12 address bits, Address[31:20], of 32-bit addresses. For the purpose of address decoding, the bridge assumes that the lower 20 address bits, Address[19:0], of the memory limit address (not implemented in the Memory Limit register) are F FFFFh. The Memory Limit register must be programmed to a smaller value than the Memory Base register if there is no memory-mapped address space on the secondary side of the bridge.
19:16	RO	0x0	**MEM_LIMIT_RESERV**: Reserved.
15:4	RW	0x0	**MEM_BASE**: Memory Base Address. These bits correspond to the upper 12 address bits, Address[31:20], of 32-bit addresses. For the purpose of address decoding, the bridge assumes that the lower 20 address bits, Address[19:0], of the memory base address (not implemented in the Memory Base register) are zero.
3:0	RO	0x0	**MEM_BASE_RESERV**: Reserved.

PCIE_X<*i*>_RC_PF0_TYPE1_HDR_PREF_MEM_LIMIT_PREF_MEM_BASE_REG_0

where <*i*> = 1, 4, 8.

Description: The Prefetchable Memory Limit and Prefetchable Memory Base registers must indicate that 64-bit addresses are supported. The Prefetchable Memory Limit and Prefetchable Memory Base registers are optional. They define a prefetchable memory address range which is used by the bridge to determine when to forward memory transactions from one interface to the other (see the PCI-to-PCI Bridge Architecture Specification for additional details).

PCIE_X1_RC_PF0_TYPE1_HDR_PREF_MEM_LIMIT_PREF_MEM_BASE_REG_0

PCIE_X4_RC_PF0_TYPE1_HDR_PREF_MEM_LIMIT_PREF_MEM_BASE_REG_0

PCIE_X8_RC_PF0_TYPE1_HDR_PREF_MEM_LIMIT_PREF_MEM_BASE_REG_0

Offset: 0x24
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00010001 (0b0000,0000,0000,0001,0000,0000,0000,0001)

Bit	R/W	Reset	Description
31:20	RW	0x0	**PREF_MEM_LIMIT**: Prefetchable Memory Limit Address. - If the Prefetchable Memory Limit register indicates support for 32-bit addressing, then the Prefetchable Limit Upper 32 Bits register is implemented as a read-only register that returns zero when read. - If the Prefetchable Memory Limit registers indicate support for 64-bit addressing, then the Prefetchable Limit Upper 32 Bits register is implemented as a read/write register which must be initialized by configuration software. - If a 64-bit prefetchable memory address range is supported, the Prefetchable Limit Upper 32 Bits register specifies the upper 32 bits, corresponding to Address[63:32], of the 64-bit limit addresses which specify the prefetchable memory address range.
19:17	RO	0x0	**PREF_RESERV1**: Reserved.
16	RO	0x1	**PREF_MEM_LIMIT_DECODE**: Prefetchable Memory Limit Decode. This bit encodes whether or not the bridge supports 64-bit addresses. Note: This bit is a copy of the PREF_MEM_DECODE bit and always reflects the current value of that bit.
15:4	RW	0x0	**PREF_MEM_BASE**: Prefetchable Memory Base Address. - If the Prefetchable Memory Base register indicates support for 32-bit addressing, then the Prefetchable Base Upper 32 Bits register is implemented as a read-only register that returns zero when read. - If the Prefetchable Memory Base register indicates support for 64-bit addressing, then the Prefetchable Limit Upper 32 Bits register is implemented as a read/write register which must be initialized by configuration software. - If a 64-bit prefetchable memory address range is supported, the Prefetchable Base Upper 32 Bits register specifies the upper 32 bits, corresponding to Address[63:32], of the 64-bit base addresses which specify the prefetchable memory address range.
3:1	RO	0x0	**PREF_RESERV**: Reserved.
0	RW	0x1	**PREF_MEM_DECODE**: Prefetchable Memory Base Decode. This bit encodes whether or not the bridge supports 64-bit addresses. Note: By default the bit is set to 1'b1 indicating that 64-bit addresses are supported. If the bridge only supports 32-bit prefetchable memory address range, or if there is no prefetchable memory address range, then the configuration parameter MEM_DECODE_64_0 must be changed to 0. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_TYPE1_HDR_PREF_BASE_UPPER_REG_0

where <*i*> = 1, 4, 8.

Description: The Prefetchable Base Upper 32 Bits register is an optional extension to the Prefetchable Memory Base register.

PCIE_X1_RC_PF0_TYPE1_HDR_PREF_BASE_UPPER_REG_0

PCIE_X4_RC_PF0_TYPE1_HDR_PREF_BASE_UPPER_REG_0

PCIE_X8_RC_PF0_TYPE1_HDR_PREF_BASE_UPPER_REG_0

Offset: 0x28
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PREF_MEM_BASE_UPPER**: Prefetchable Base Upper 32 Bit. - If the Prefetchable Memory Base register indicates support for 32-bit addressing, then this register is implemented as read-only register that returns zero when read. - If the Prefetchable Memory Base register indicate support for 64-bit addressing, then this register is implemented as read/write register which must be initialized by configuration software. - This register specifies the upper 32 bits, corresponding to Address[63:32], of the 64-bit base addresses which specify the prefetchable memory address range. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: PREF_MEM_LIMIT_PREF_MEM_BASE_REG.PREF_MEM_DECODE ? RW : RO

PCIE_X<*i*>_RC_PF0_TYPE1_HDR_PREF_LIMIT_UPPER_REG_0

where <*i*> = 1, 4, 8.

Description: The Prefetchable Limit Upper 32 Bits register is an optional extension to the Prefetchable Memory Limit register.

PCIE_X1_RC_PF0_TYPE1_HDR_PREF_LIMIT_UPPER_REG_0

PCIE_X4_RC_PF0_TYPE1_HDR_PREF_LIMIT_UPPER_REG_0

PCIE_X8_RC_PF0_TYPE1_HDR_PREF_LIMIT_UPPER_REG_0

Offset: 0x2c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PREF_MEM_LIMIT_UPPER**: Prefetchable Limit Upper 32 Bit. - If the Prefetchable Memory Limit register indicate support for 64-bit addressing, then this register is implemented as read/write register which must be initialized by configuration software. - This register specifies the upper 32 bits, corresponding to Address[63:32], of the 64-bit base addresses which specify the prefetchable memory address range. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: PREF_MEM_LIMIT_PREF_MEM_BASE_REG.PREF_MEM_DECODE ? RW : RO

PCIE_X<*i*>_RC_PF0_TYPE1_HDR_IO_LIMIT_UPPER_IO_BASE_UPPER_REG_0

where <*i*> = 1, 4, 8.

Description: The I/O Limit Upper 16 Bits and I/O Base Upper 16 Bits registers are optional extensions to the I/O Limit and I/O Base registers.

Offset: 0x30
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

PCIE_X1_RC_PF0_TYPE1_HDR_IO_LIMIT_UPPER_IO_BASE_UPPER_REG_0

Read/Write: RO

PCIE_X4_RC_PF0_TYPE1_HDR_IO_LIMIT_UPPER_IO_BASE_UPPER_REG_0

PCIE_X8_RC_PF0_TYPE1_HDR_IO_LIMIT_UPPER_IO_BASE_UPPER_REG_0

Read/Write: R/W

Bit	Reset	Description
31:16	0x0	**IO_LIMIT_UPPER**: I/O Limit Upper 16 Bits. - If the I/O Limit register indicates support for 16-bit I/O address decoding, then this register is implemented as a read-only register which return zero when read. - If the I/O Limit register indicates support for 32-bit I/O addressing, then this register must be initialized by configuration software. - If 32-bit I/O address decoding is supported, this register specifies the upper 16 bits, corresponding to Address[31:16], of the 32-bit limit address, that specify the I/O address range. - See the PCI-to-PCI Bridge Architecture Specification for additional details). Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: SEC_STAT_IO_LIMIT_IO_BASE_REG.IO_DECODE ? RW : RO
15:0	0x0	**IO_BASE_UPPER**: I/O Base Upper 16 Bits. - If the I/O Base register indicates support for 16-bit I/O address decoding, then this register is implemented as a read-only register which return zero when read. - If the I/O base register indicates support for 32-bit I/O addressing, then this register must be initialized by configuration software. - If 32-bit I/O address decoding is supported, this register specifies the upper 16 bits, corresponding to Address[31:16], of the 32-bit base address, that specify the I/O address range. - See the PCI-to-PCI Bridge Architecture Specification for additional details. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: SEC_STAT_IO_LIMIT_IO_BASE_REG.IO_DECODE ? RW : RO

PCIE_X<*i*>_RC_PF0_TYPE1_HDR_TYPE1_CAP_PTR_REG_0

where <*i*> = 1, 4, 8.

Description: This register is used to point to a linked list of capabilities implemented by a Function.

PCIE_X1_RC_PF0_TYPE1_HDR_TYPE1_CAP_PTR_REG_0

PCIE_X4_RC_PF0_TYPE1_HDR_TYPE1_CAP_PTR_REG_0

PCIE_X8_RC_PF0_TYPE1_HDR_TYPE1_CAP_PTR_REG_0

Offset: 0x34
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000040 (0b0000,0000,0000,0000,0000,0000,0100,0000)

Bit	R/W	Reset	Description
31:8	RO	0x0	**RSVDP_8**: Reserved for future use.

Bit	R/W	Reset	Description
7:0	RW	0x40	**CAP_POINTER**: Capabilities Pointer. This register is used to point to a linked list of capabilities implemented by this Function. Since all PCI Express Functions are required to implement the PCI Express Capability structure, this register must point to a valid capability structure and either this structure is the PCI Express Capability structure, or a subsequent list item points to the PCI Express Capability structure. The bottom two bits are Reserved and must be set to 00b. Software must mask these bits off before using this register as a pointer in Configuration Space to the first entry of a linked list of new capabilities. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_TYPE1_HDR_TYPE1_EXP_ROM_BASE_REG_0

where <*i*> = 1, 4, 8.

Description: This register is defined to handle the base address and size information for this expansion ROM. The mask for this ROM BAR exists (if implemented) as a shadow register at this address. The assertion of CS2 (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) is required to write to the second register at this address.

PCIE_X1_RC_PF0_TYPE1_HDR_TYPE1_EXP_ROM_BASE_REG_0

PCIE_X4_RC_PF0_TYPE1_HDR_TYPE1_EXP_ROM_BASE_REG_0

PCIE_X8_RC_PF0_TYPE1_HDR_TYPE1_EXP_ROM_BASE_REG_0

Offset: 0x38
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:11	RW	0x0	**EXP_ROM_BASE_ADDRESS**: Expansion ROM Base Address. Upper 21 bits of the Expansion ROM base address. The number of bits (out of these 21) that a Function actually implements depends on how much address space the Function requires. The mask for this ROM BAR exists (if implemented) as a shadow register at this address. The assertion of CS2 (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) is required to write to the second register at this address. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (EXP_ROM_BAR_MASK_REG.ROM_BAR_ENABLED == 1) then R/W else R
10:8	RO	0x0	**RSVDP_8**: Reserved for future use.

Bit	R/W	Reset	Description
7:4	RW	0x0	**ROM_BAR_VALIDATION_DETAILS**: Expansion ROM Validation Details. The field contains optional, implementation-specific details associated with Expansion ROM Validation. - If validation is in progress (Expansion ROM Validation Status is 001b), non-zero values of this field represent implementation-specific indications of the phase of the validation progress (e.g., 50% complete). The value 0000b indicates that no validation progress information is provided. - If validation is completed (Expansion ROM Validation Status 010b to 111b inclusive), non-zero values in this field represent additional implementation-specific information. The value 0000b indicates that no information is provided. - When validation is supported and this field is not implemented, this field must be hardwired to 0000b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (EXP_ROM_BAR_MASK_REG.ROM_BAR_ENABLED == 1 && DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
3:1	RW	0x0	**ROM_BAR_VALIDATION_STATUS**: Expansion ROM Validation Status. When this field is non-zero, it indicates the status of hardware validation of the Expansion ROM contents. - If the Function does not support validation, this field must be hardwired to 000b. - It is optional whether an implementation is capable of returning Validation Status values 011b, 101b, 110b, or 111b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (EXP_ROM_BAR_MASK_REG.ROM_BAR_ENABLED == 1 && DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
0	RW	0x0	**ROM_BAR_ENABLE**: Expansion ROM Enable. This bit controls whether or not the Function accepts accesses to its expansion ROM. The Memory Space Enable bit in the Command register has precedence over the Expansion ROM Enable bit. A Function must claim accesses to its expansion ROM only if both the Memory Space Enable bit and the Expansion ROM Enable bit are set. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (EXP_ROM_BAR_MASK_REG.ROM_BAR_ENABLED == 1) then R/W else R

PCIE_X<*i*>_RC_PF0_TYPE1_HDR_BRIDGE_CTRL_INT_PIN_INT_LINE_REG_0

where <*i*> = 1, 4, 8.

Description: This register provides many of the same controls for the secondary interface that are provided by the Command Register for the primary interface. This register also identifies the legacy interrupt Message(s) a Function uses and communicates interrupt line routing information.

PCIE_X1_RC_PF0_TYPE1_HDR_BRIDGE_CTRL_INT_PIN_INT_LINE_REG_0

PCIE_X4_RC_PF0_TYPE1_HDR_BRIDGE_CTRL_INT_PIN_INT_LINE_REG_0

PCIE_X8_RC_PF0_TYPE1_HDR_BRIDGE_CTRL_INT_PIN_INT_LINE_REG_0

Offset: 0x3c
Read/Write: See table below
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x000001ff (0b0000,0000,0000,0000,0000,0001,1111,1111)

Bit	R/W	Reset	Description
31:23	RO	0x0	**BRIDGE_CTRL_RESERV**: Reserved.
22	RW	0x0	**SBR**: Secondary Bus Reset. Setting this bit triggers a hot reset on the corresponding PCI Express Port. Software must ensure a minimum reset duration (Trst) as defined in the PCI Local Bus Specification. Software and systems must honor first-access-following-reset timing requirements, unless the Readiness Notifications mechanism is used or if the Immediate Readiness bit in the relevant Function's Status Register register is set. Port configuration registers must not be changed, except as required to update Port status.
21	RO	0x0	**MSTR_ABORT_MODE**: Master Abort Mode. This bit was originally described in the PCI-to-PCI Bridge Architecture Specification. Its functionality does not apply to PCI Express. The controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
20	RO	0x0	**VGA_16B_DEC**: VGA 16 bit decode. This bit only has meaning if VGA Enable bit is set. This bit enables system configuration software to select between 10-bit and 16-bit I/O address decoding for all VGA I/O register accesses that are forwarded from primary to secondary. For Functions that do not support VGA, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R
19	RO	0x0	**VGA_EN**: VGA Enable. Modifies the response by the bridge to VGA compatible addresses. If the VGA Enable bit is set, the bridge will positively decode and forward the following accesses on the primary interface to the secondary interface (and, conversely, block the forwarding of these addresses from the secondary to primary interface): - Memory accesses in the range 000A 0000h to 000B FFFFh - I/O addresses in the first 64 KB of the I/O address space (Address[31:16] are 0000h) where Address[9:0] are in the ranges 3B0h to 3BBh and 3C0h to 3DFh (inclusive of ISA address aliases determined by the setting of VGA 16-bit Decode) If the VGA Enable bit is set, forwarding of these accesses is independent of the I/O address range and memory address ranges defined by the I/O Base and Limit registers, the Memory Base and Limit registers, and the Prefetchable Memory Base and Limit registers of the bridge. (Forwarding of these accesses is also independent of the setting of the ISA Enable bit (in the Bridge Control register) when the VGA Enable bit is set. Forwarding of these accesses is qualified by the I/O Space Enable and Memory Space Enable bits in the Command register.) For Functions that do not support VGA, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R
18	RW	0x0	**ISA_EN**: ISA Enable. Modifies the response by the bridge to ISA I/O addresses. This applies only to I/O addresses that are enabled by the I/O Base and I/O Limit registers and are in the first 64 KB of I/O address space (0000 0000h to 0000 FFFFh). If this bit is set, the bridge will block any forwarding from primary to secondary of I/O transactions addressing the last 768 bytes in each 1-KB block. In the opposite direction (secondary to primary), I/O transactions will be forwarded if they address the last 768 bytes in each 1-KB block.

Bit	R/W	Reset	Description
17	RW	0x0	**SERR_EN**: SERR# Enable. This bit controls forwarding of ERR_COR, ERR_NONFATAL and ERR_FATAL from secondary to primary.
16	RW	0x0	**PERE**: Parity Error Response Enable. This bit controls the logging of poisoned TLPs in the Master Data Parity Error bit in the Secondary Status register.
15:8	RW	0x1	**INT_PIN**: Reserved. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
7:0	RW	0xff	**INT_LINE**: Reserved

PCIE_X<*i*>_RC_PF0_PM_CAP_CAP_ID_NXT_PTR_REG_0

where <*i*> = 1, 4, 8.

Description: This register provides information refarding the Power Management Capabilities.

PCIE_X1_RC_PF0_PM_CAP_CAP_ID_NXT_PTR_REG_0

PCIE_X4_RC_PF0_PM_CAP_CAP_ID_NXT_PTR_REG_0

PCIE_X8_RC_PF0_PM_CAP_CAP_ID_NXT_PTR_REG_0

Offset: 0x40
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x48035001 (0b0100,1000,000x,0011,0101,0000,0000,0001)

Bit	R/W	Reset	Description
31:27	RW	0x9	**PME_SUPPORT**: PME_Support. This 5-bit field indicates the power states in which the function may generate a PME and/or forward PME messages. A value of 0b for any bit indicates that the function is not capable of asserting PME while in that power state. - bit(27) X XXX1b - PME can be generated from D0 - bit(28) X XX1Xb - PME can be generated from D1 - bit(29) X X1XXb - PME can be generated from D2 - bit(30) X 1XXXb - PME can be generated from D3hot - bit(31) 1 XXXXb - PME can be generated from D3cold Bit 31 (PME can be asserted from D3cold) represents a special case. Functions that set this bit require some sort of auxiliary power source. Implementation specific mechanisms are recommended to validate that the power source is available before setting this bit. Each bit that corresponds to a supported D-state must be set for PCI-PCI Bridge structures representing Ports on Root Complexes/Switches to indicate that the Bridge will forward PME Messages. Bit 31 must only be set if the Port is still able to forward PME Messages when main power is not available. The read value from this field is the write value && (sys_aux_pwr_det, 1'b1, D2_SUPPORT, D1_SUPPORT, 1'b1), where D1_SUPPORT and D2_SUPPORT are fields in this register. The reset value PME_SUPPORT_n && (sys_aux_pwr_det, 1'b1, D2_SUPPORT, D1_SUPPORT, 1'b1), where PME_SUPPORT_n is a configuration parameter. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
26	RW	0x0	**D2_SUPPORT**: D2_Support. If this bit is set, this function supports the D2 Power Management state. Functions that do not support D2 must always return a value of 0b for this bit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
25	RW	0x0	**D1_SUPPORT**: D1_Support. If this bit is set, this function supports the D1 Power Management state. Functions that do not support D1 must always return a value of 0b for this bit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
24:22	RW	0x0	**AUX_CURR**: Aux_Current. This 3 bit field reports the Vaux auxiliary current requirements for the function. If this function implements the Data Register, the controller hardwires this field to 000b. If PME_Support is 0 xxxxb (PME assertion from D3cold is not supported), the controller hardwires this field to 0000b. For functions where PME_Support is 1 xxxxb (PME assertion from D3cold is supported), and which do not implement the Data field, the encodings defined in Values: apply: Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
21	RW	0x0	**DSI**: Device Specific Initialization. The DSI bit indicates whether special initialization of this function is required. When set, indicates that the function requires a device specific initialization sequence following a transition to the D0uninitialized state. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
19	RO	0x0	**PME_CLK**: PME Clock. Does not apply to PCI Express, the controller hardwires it to 0b. Note: This register field is sticky.

Bit	R/W	Reset	Description
18:16	RW	0x3	**PM_SPEC_VER**: Version. This field provides the Power Management specification version. The controller hardwires this field to 011b for functions compliant to PCI Express Base Specification, Revision 4.0, Version 1.0>. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
15:8	RW	0x50	**PM_NEXT_POINTER**: Next Capability Pointer. This field provides an offset into the function's configuration space pointing to the location of next item in the capabilities list. If there are no additional items in the capabilities list, this field is set to 00h. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
7:0	RO	0x1	**PM_CAP_ID**: Capability ID. This field returns 01h to indicate that this is the PCI Power Management Capability. Each function may have only one item in its capability list with Capability ID set to 01h.

PCIE_X1_RC_PF0_PM_CAP_CON_STATUS_REG_0

where <*i*> = 1, 4, 8.

Description: This register is used to manage the PCI function's power management state as well as to enable/monitor PMEs.

PCIE_X1_RC_PF0_PM_CAP_CON_STATUS_REG_0

PCIE_X4_RC_PF0_PM_CAP_CON_STATUS_REG_0

PCIE_X8_RC_PF0_PM_CAP_CON_STATUS_REG_0

Offset: 0x44
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000008 (0b0000,0000,0000,0000,0000,0000,0000,1000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**DATA_REG_ADD_INFO**: Data. This field is used to report the state dependent data requested by the Data_Select field. The value of this field is scaled by the value reported by the Data_Scale field.
23	RO	0x0	**BUS_PWR_CLK_CON_EN**: Bus Power/Clock Control Enable. If this field is set, Bus Power/Clock Control is Enable.

Bit	R/W	Reset	Description
22	RO	0x0	**B2_B3_SUPPORT**: B2B3 Support for D3hot. If this field is set, B2B3 support for D3hot is available.
21:16	RO	0x0	**RSVDP_16**: Reserved for future use.
15	RW	0x0	**PME_STATUS**: PME_Status. This bit is set when the function normally generates a PME signal. The value of this bit is not affected by the value of the PME_En bit. If PME_Support bit 31 of the Power Management Capabilities register is clear, this bit is permitted to be hardwired to 0b. Functions that consume Aux power must preserve the value of this sticky register when Aux power is available. In such functions, this register value is not modified by Conventional Reset or FLR.
14:13	RO	0x0	**DATA_SCALE**: Data_Scale. This field indicates the scaling factor to be used when interpreting the value of the Data field. The value and meaning of this field varies depending on which data value has been selected by the Data_Select field. For more details, see 7.5.2.3 section of PCI Express Base Specification.
12:9	RO	0x0	**DATA_SELECT**: Data_Select. This 4-bit field is used to select which data is to be reported through the Data and Data_Scale field. If the Data field is not implemented, this field must be hardwired to 0000b.
8	RW	0x0	**PME_ENABLE**: PME_En. - When set, the function is permitted to generate a PME. - When clear, the function is not permitted to generate a PME. If PME_Support is 1 xxxxb (PME generation from D3cold) or the function consumes Aux power and Aux power is available this bit is RWS and the bit is not modified by Conventional Reset or FLR. If PME_Support is 0 xxxxb, this field is not sticky (RW). If PME_Support is 0 0000b, the controller hardwires this bit to 0b. Note: This register field is sticky.
7:4	RO	0x0	**RSVDP_4**: Reserved for future use.
3	RW	0x1	**NO_SOFT_RST**: No_Soft_Reset. This bit indicates the state of the function after writing the PowerState field to transition the function from D3hot to D0. - When set, this transition preserves internal function state. The function is in D0Active and no additional software intervention is required. - When clear, this transition results in undefined internal function state. Regardless of this bit, functions that transition from D3hot to D0 by Fundamental Reset will return to D0Uninitialized with only PME context preserved if PME is supported and enabled. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
2	RO	0x0	**RSVDP_2**: Reserved for future use.

Bit	R/W	Reset	Description
1:0	RW	0x0	**POWER_STATE**: PowerState. This 2-bit field is used both to determine the current power state of a function and to set the function into a new power state. You can write to this register; however, the read-back value is the actual power state, not the write value. If you attempt to write an unsupported, optional state to this field, the write operation completes normally; however, the data is discarded and no state change occurs. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_X<*i*>_RC_PF0_MSI_CAP_PCI_MSI_CAP_ID_NEXT_CTRL_REG_0

where <*i*> = 1, 4, 8.

Description: This register holds MSI Capability Header information and controls the MSI behaviour.

Offset: 0x50
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_MSI_CAP_PCI_MSI_CAP_ID_NEXT_CTRL_REG_0

Reset: 0x00807005 (0b0000,0000,1000,0000,0111,0000,0000,0101)

Bit	R/W	Reset	Description
31:27	RO	0x0	**RSVDP_27**: Reserved for future use.
26	RW	0x0	**PCI_MSI_EXT_DATA_EN**: Extended Message Data Enable. - If set, the function is enabled to provide Extended Message Data. - If clear, the function is not enabled to provide Extended Message Data. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: PCI_MSI_CAP_ID_NEXT_CTRL_REG.PCI_MSI_EXT_DATA_CAP ? RW : RO
25	RW	0x0	**PCI_MSI_EXT_DATA_CAP**: Extended Message Data Capable. - If set, the function is capable of providing Extended Message Data. - If clear, the function does not support providing Extended Message Data. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
24	RO	0x0	**PCI_PVM_SUPPORT**: Per-Vector Masking Capable. - If set, the function supports MSI Per-Vector Masking. - If clear, the function does not support MSI Per-Vector Masking.

Bit	R/W	Reset	Description
23	RW	0x1	**PCI_MSI_64_BIT_ADDR_CAP**: 64 bit address capable. - If set, the function is capable of sending a 64-bit message address. - If clear, the function is not capable of sending a 64-bit message address. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
22:20	RW	0x0	**PCI_MSI_MULTIPLE_MSG_EN**: Multiple Message Enable. Software writes to this field to indicate the number of allocated vectors (equal to or less than the number of requested vectors). The number of allocated vectors is aligned to a power of two. If a function requests four vectors (indicated by a Multiple Message Capable encoding of 010b), system software can allocate either four, two, or one vector by writing a 010b, 001b, or 000b to this field, respectively. When MSI is enabled, a function will be allocated at least 1 vector. All encodings other than the defined encodings are reserved.
19:17	RW	0x0	**PCI_MSI_MULTIPLE_MSG_CAP**: Multiple Message Capable. System software reads this field to determine the number of requested vectors. The number of requested vectors must be aligned to a power of two (if a function requires three vectors, it requests four by initializing this field to 010b). All encodings other than the defined encodings are reserved. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
16	RW	0x0	**PCI_MSI_ENABLE**: MSI Enable. - If set and the MSI-X Enable bit in the MSI-X Message Control register is clear, the function is permitted to use MSI to request service and is prohibited from using INTx interrupts. System configuration software sets this bit to enable MSI. A device driver is prohibited from writing this bit to mask a function's service request. For more details on control of INTx interrupts, see section 7.5.1.1 of PCI Express Base Specification. - If clear, the function is prohibited from using MSI to request service.
15:8	RW	0x70	**PCI_MSI_CAP_NEXT_OFFSET**: Next Capability Pointer. This field contains the offset to the next PCI Capability structure or 00h if no other items exist in the linked list of Capabilities. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
7:0	RO	0x5	**PCI_MSI_CAP_ID**: Capability ID. Indicates the MSI Capability structure. This field returns a Capability ID of 05h indicating that this is an MSI Capability structure.

PCIE_X4_RC_PF0_MSI_CAP_PCI_MSI_CAP_ID_NEXT_CTRL_REG_0

PCIE_X8_RC_PF0_MSI_CAP_PCI_MSI_CAP_ID_NEXT_CTRL_REG_0

Reset: 0x01807005 (0b0000,0001,1000,0000,0111,0000,0000,0101)

Bit	R/W	Reset	Description
31:27	RO	0x0	**RSVDP_27**: Reserved for future use.

Bit	R/W	Reset	Description
26	RW	0x0	**PCI_MSI_EXT_DATA_EN**: Extended Message Data Enable. - If set, the function is enabled to provide Extended Message Data. - If clear, the function is not enabled to provide Extended Message Data. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: PCI_MSI_CAP_ID_NEXT_CTRL_REG.PCI_MSI_EXT_DATA_CAP ? RW : RO
25	RW	0x0	**PCI_MSI_EXT_DATA_CAP**: Extended Message Data Capable. - If set, the function is capable of providing Extended Message Data. - If clear, the function does not support providing Extended Message Data. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
24	RO	0x1	**PCI_PVM_SUPPORT**: Per-Vector Masking Capable. - If set, the function supports MSI Per-Vector Masking. - If clear, the function does not support MSI Per-Vector Masking. This bit must be set if the function is a PF or VF within an SR-IOV Device.
23	RW	0x1	**PCI_MSI_64_BIT_ADDR_CAP**: 64 bit address capable. - If set, the function is capable of sending a 64-bit message address. - If clear, the function is not capable of sending a 64-bit message address. This bit must be set if the function is a PCI Express Endpoint. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
22:20	RW	0x0	**PCI_MSI_MULTIPLE_MSG_EN**: Multiple Message Enable. Software writes to this field to indicate the number of allocated vectors (equal to or less than the number of requested vectors). The number of allocated vectors is aligned to a power of two. If a function requests four vectors (indicated by a Multiple Message Capable encoding of 010b), system software can allocate either four, two, or one vector by writing a 010b, 001b, or 000b to this field, respectively. When MSI is enabled, a function will be allocated at least 1 vector. All encodings other than the defined encodings are reserved.
19:17	RW	0x0	**PCI_MSI_MULTIPLE_MSG_CAP**: Multiple Message Capable. System software reads this field to determine the number of requested vectors. The number of requested vectors must be aligned to a power of two (if a function requires three vectors, it requests four by initializing this field to 010b). All encodings other than the defined encodings are reserved. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
16	RW	0x0	**PCI_MSI_ENABLE**: MSI Enable. - If set and the MSI-X Enable bit in the MSI-X Message Control register is clear, the function is permitted to use MSI to request service and is prohibited from using INTx interrupts. System configuration software sets this bit to enable MSI. A device driver is prohibited from writing this bit to mask a function's service request. For more details on control of INTx interrupts, see section 7.5.1.1 of PCI Express Base Specification. - If clear, the function is prohibited from using MSI to request service.

Bit	R/W	Reset	Description
15:8	RW	0x70	**PCI_MSI_CAP_NEXT_OFFSET**: Next Capability Pointer. This field contains the offset to the next PCI Capability structure or 00h if no other items exist in the linked list of Capabilities. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
7:0	RO	0x5	**PCI_MSI_CAP_ID**: Capability ID. Indicates the MSI Capability structure. This field returns a Capability ID of 05h indicating that this is an MSI Capability structure.

PCIE_X<*i*>_RC_PF0_MSI_CAP_MSI_CAP_OFF_04H_REG_0

where <*i*> = 1, 4, 8.

Description: This register holds the system specified message address for an MSI transaction.

PCIE_X1_RC_PF0_MSI_CAP_MSI_CAP_OFF_04H_REG_0

PCIE_X4_RC_PF0_MSI_CAP_MSI_CAP_OFF_04H_REG_0

PCIE_X8_RC_PF0_MSI_CAP_MSI_CAP_OFF_04H_REG_0

Offset: 0x54
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:2	RW	0x0	**PCI_MSI_CAP_OFF_04H**: Message Address - System-specified message address. If the Message Enable bit (bit 16 of the PCI_MSI_CAP_ID_NEXT_CTRL_REG register) is set, the contents of this field specify the DWORD-aligned address (Address[31:02]) for the MSI transaction. Address[1:0] are set to 00b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
1:0	RO	0x0	**RSVDP_0**: Reserved for future use.

PCIE_X<*i*>_RC_PF0_MSI_CAP_MSI_CAP_OFF_08H_REG_0

where <*i*> = 1, 4, 8.

Description: For a function that supports a 32-bit message address, - bits[31:16] of this register represent the Extended Message Data, and - bits[15:0] of this register represent the Message Data For a function that supports a 64-bit message address (bit 23 in PCI_MSI_CAP_ID_NEXT_CTRL_REG register set), this register represents the Message Upper Address Register for MSI (Offset 08h). It specifies the Message Upper Address (System-specified message upper address). This register is required for PCI Express Endpoints and is optional for other function types. If the Message Enable bit (bit 0 of the Message Control register) is set, the contents of this register (if non-zero) specify the upper 32-bits of a 64-bit message address (Address[63:32]). If the contents of this register are zero, the Function uses the 32 bit address specified by the Message Address register.

PCIE_X1_RC_PF0_MSI_CAP_MSI_CAP_OFF_08H_REG_0

PCIE_X4_RC_PF0_MSI_CAP_MSI_CAP_OFF_08H_REG_0

PCIE_X8_RC_PF0_MSI_CAP_MSI_CAP_OFF_08H_REG_0

Offset: 0x58
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PCI_MSI_CAP_OFF_0AH**: For a function that supports a 32-bit message address, this field contains Extended Message Data (System-specified message data). For the MSI Capability structures without per-vector masking, it must be implemented if the Extended Message Data Capable bit is set; otherwise, it is outside the MSI Capability structure and undefined. For the MSI Capability structures with Per-vector Masking, it must be implemented if the Extended Message Data Capable bit is set; otherwise, it is RsvdP. If the Extended Message Data Enable bit (bit 26 of the PCI_MSI_CAP_ID_NEXT_CTRL_REG) is set, the DWORD Memory Write transaction uses Extended Message Data for the upper 16 bits; otherwise, it uses 0000h for the upper 16 bits. For a function that supports a 64-bit message address, it contains upper 16 bits of the Message Upper Address. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: PCI_MSI_64_BIT_ADDR_CAP || `DEFAULT_EXT_MSI_DATA_CAPABLE ? R/W : R

Bit	Reset	Description
15:0	0x0	**PCI_MSI_CAP_OFF_08H**: For a function that supports a 32-bit message address, this field contains Message Data (System-specified message data). If the Message Enable bit (bit 16 of the PCI_MSI_CAP_ID_NEXT_CTRL_REG) is set, the function sends a DWORD Memory Write transaction using Message Data for the lower 16 bits. All 4 Byte Enables are set. The Multiple Message Enable field (bits 22:20 of the PCI_MSI_CAP_ID_NEXT_CTRL_REG) defines the number of low order message data bits the function is permitted to modify to generate its system software allocated vectors. For example, a Multiple Message Enable encoding of 010b indicates the function has been allocated four vectors and is permitted to modify message data bits 1 and 0 (a function modifies the lower message data bits to generate the allocated number of vectors). If the Multiple Message Enable field is 000b, the Function is not permitted to modify the message data. For a function that supports a 64-bit message address, it contains lower 16 bits of the Message Upper Address. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: PCI_MSI_64_BIT_ADDR_CAP ? R/W : R

PCIE_X<*i*>_RC_PF0_MSI_CAP_MSI_CAP_OFF_0CH_REG_0

where <*i*> = 1, 4, 8.

Description: For a function that supports a 32-bit message address, this register contains the Mask Bits when the Per-Vector Masking Capable bit (bit 24 of PCI_MSI_CAP_ID_NEXT_CTRL_REG) is set. For a function that supports a 64-bit message address, this register contains Message Data.

PCIE_X1_RC_PF0_MSI_CAP_MSI_CAP_OFF_0CH_REG_0

PCIE_X4_RC_PF0_MSI_CAP_MSI_CAP_OFF_0CH_REG_0

PCIE_X8_RC_PF0_MSI_CAP_MSI_CAP_OFF_0CH_REG_0

Offset: 0x5c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PCI_MSI_CAP_OFF_0EH**: For a function that supports a 32-bit message address, this field contains the upper Mask Bits when the Per Vector Masking Capable bit (PCI_MSI_CAP_ID_NEXT_CTRL_REG.PCI_PVM_SUPPORT) is set. For a function that supports a 64-bit message address, this field contains Message Data (System-specified message data). Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: (!MSI_64_EN && MSI_PVM_EN_VALUE) ? RW: MSI_64_EN && DEFAULT_EXT_MSI_DATA_CAPABLE ? RW : RO

Bit	Reset	Description
15:0	0x0	**PCI_MSI_CAP_OFF_0CH**: For a function that supports a 32-bit message address, this field contains the lower Mask Bits when the Per Vector Masking Capable bit (PCI_MSI_CAP_ID_NEXT_CTRL_REG.PCI_PVM_SUPPORT) is set. For a function that supports a 64-bit message address, this field contains Message Data (System-specified message data). If the Message Enable bit (bit 16 of the PCI_MSI_CAP_ID_NEXT_CTRL_REG) is set, the function sends a DWORD Memory Write transaction using Message Data for the lower 16 bits. All 4 Byte Enables are set. The Multiple Message Enable field (bits 22:20 of the PCI_MSI_CAP_ID_NEXT_CTRL_REG) defines the number of low order message data bits the function is permitted to modify to generate its system software allocated vectors. For example, a Multiple Message Enable encoding of 010b indicates the function has been allocated four vectors and is permitted to modify message data bits 1 and 0 (a function modifies the lower message data bits to generate the allocated number of vectors). If the Multiple Message Enable field is 000b, the Function is not permitted to modify the message data. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: PCI_MSI_64_BIT_ADDR_CAP || MSI_PVM_EN ? R/W : R

PCIE_X<*j*>_RC_PF0_MSI_CAP_MSI_CAP_OFF_10H_REG_0

where <*j*> = 4, 8.

Description: For a function that supports a 32-bit message address, this register contains the Pending Bits when the Per-Vector Masking Capable bit (bit 24 of PCI_MSI_CAP_ID_NEXT_CTRL_REG) is set. For a function that supports a 64-bit message address, this register contains the Mask Bits when the Per-Vector Masking Capable bit (bit 24 of PCI_MSI_CAP_ID_NEXT_CTRL_REG) is set.

PCIE_X4_RC_PF0_MSI_CAP_MSI_CAP_OFF_10H_REG_0

PCIE_X8_RC_PF0_MSI_CAP_MSI_CAP_OFF_10H_REG_0

Offset: 0x60
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PCI_MSI_CAP_OFF_10H**: Used for MSI when the Per Vector Masking Capable bit (PCI_MSI_CAP_ID_NEXT_CTRL_REG.PCI_PVM_SUPPORT) is set. For 32-bit contains Pending Bits. For 64-bit, contains Mask Bits. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: PCI_MSI_64_BIT_ADDR_CAP && MSI_PVM_EN ? R/W : R

PCIE_X<*j*>_RC_PF0_MSI_CAP_MSI_CAP_OFF_14H_REG_0

where <*j*> = 4, 8.

Description: Pending Bits Register for MSI. This register is used for a function that supports a 64-bit message address when the Per-Vector Masking Capable bit (bit 24 of PCI_MSI_CAP_ID_NEXT_CTRL_REG) is set.

PCIE_X4_RC_PF0_MSI_CAP_MSI_CAP_OFF_14H_REG_0

PCIE_X8_RC_PF0_MSI_CAP_MSI_CAP_OFF_14H_REG_0

Offset: 0x64
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PCI_MSI_CAP_OFF_14H**: Pending Bits. For each pending bit that is set, the function has a pending associated message.

PCIE_X<*i*>_RC_PF0_PCIE_CAP_PCIE_CAP_ID_PCIE_NEXT_CAP_PTR_PCIE_CAP_REG_0

where <*i*> = 1, 4, 8.

Description: This is the PCI Express Capabilities, ID, and Next Pointer Register.

PCIE_X1_RC_PF0_PCIE_CAP_PCIE_CAP_ID_PCIE_NEXT_CAP_PTR_PCIE_CAP_REG_0

PCIE_X4_RC_PF0_PCIE_CAP_PCIE_CAP_ID_PCIE_NEXT_CAP_PTR_PCIE_CAP_REG_0

PCIE_X8_RC_PF0_PCIE_CAP_PCIE_CAP_ID_PCIE_NEXT_CAP_PTR_PCIE_CAP_REG_0

Offset: 0x70
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0042b010 (0b0000,0000,0100,0010,1011,0000,0001,0000)

Bit	R/W	Reset	Description
31	RO	0x0	**RSVDP_31**: Reserved for future use.
30	RO	0x0	**RSVD**: Reserved.
29:25	RW	0x0	**PCIE_INT_MSG_NUM**: PCIE Interrupt Message Number. Interrupt Message Number. This field indicates which MSI/MSI-X vector is used for the interrupt message generated in association with any of the status bits of this Capability structure. For MSI, the value in this field indicates the offset between the base Message Data and the interrupt message that is generated. Hardware is required to update this field so that it is correct if the number of MSI Messages assigned to the Function changes when software writes to the Multiple Message Enable field in the MSI Message Control register. For MSI-X, the value in this field indicates which MSI-X Table entry is used to generate the interrupt message. The entry must be one of the first 32 entries even if the Function implements more than 32 entries. For a given MSI-X implementation, the entry must remain constant. If both MSI and MSI-X are implemented, they are permitted to use different vectors, though software is permitted to enable only one mechanism at a time. If MSI-X is enabled, the value in this field must indicate the vector for MSI-X. If MSI is enabled or neither is enabled, the value in this field must indicate the vector for MSI. If software enables both MSI and MSI-X at the same time, the value in this field is undefined. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
24	RW	0x0	**PCIE_SLOT_IMP**: Slot Implemented. When set, this bit indicates that the Link associated with this Port is connected to a slot (as compared to being connected to a system-integrated device or being disabled). This bit is valid for Downstream Ports. This bit is undefined for Upstream Ports. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
23:20	RO	0x4	**PCIE_DEV_PORT_TYPE**: Device/Port Type. Indicates the specific type of this PCI Express function. Note: Different functions in a Multi-Function Device can generally be of different types. Defined encodings for functions that implement a Type 00h PCI Configuration Space header are: Defined encodings for functions that implement a Type 01h PCI Configuration Space header are: All other encodings are Reserved. Note: Different Endpoint types have notably different requirements in Section 1.3.2 of PCI Express Base Specification regarding I/O resources, Extended Configuration Space, and other capabilities.
19:16	RO	0x2	**PCIE_CAP_REG**: Capability Version. Indicates PCI-SIG defined PCI Express Capability structure version number. A version of the specification that changes the PCI Express Capability structure in a way that is not otherwise identifiable (for example, through a new Capability field) is permitted to increment this field. All such changes to the PCI Express Capability structure must be software-compatible. Software must check for Capability Version numbers that are greater than or equal to the highest number defined when the software is written, as functions reporting any such Capability Version numbers will contain a PCI Express Capability structure that is compatible with that piece of software. The controller hardwires this field to 2h for functions compliant to PCI Express Base Specification, Revision 4.0, Version 1.0. Note: This register field is sticky.

Bit	R/W	Reset	Description
15:8	RW	0xb0	**PCIE_CAP_NEXT_PTR**: Next Capability Pointer. This field contains the offset to the next PCI Capability structure or 00h if no other items exist in the linked list of Capabilities. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
7:0	RO	0x10	**PCIE_CAP_ID**: Capability ID. Indicates the PCI Express Capability structure. This field must return a Capability ID of 10h indicating that this is a PCI Express Capability structure.

PCIE_X<*i*>_RC_PF0_PCIE_CAP_DEVICE_CAPABILITIES_REG_0

where <*i*> = 1, 4, 8.

Description: The Device Capabilities register identifies PCI Express device function specific capabilities.

PCIE_X1_RC_PF0_PCIE_CAP_DEVICE_CAPABILITIES_REG_0

PCIE_X4_RC_PF0_PCIE_CAP_DEVICE_CAPABILITIES_REG_0

PCIE_X8_RC_PF0_PCIE_CAP_DEVICE_CAPABILITIES_REG_0

Offset: 0x74
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00008001 (0b0000,0000,0000,0000,1000,0000,0000,0001)

Bit	R/W	Reset	Description
31:16	RO	0x0	**RSVDP_16**: Reserved for future use.
15	RW	0x1	**PCIE_CAP_ROLE_BASED_ERR_REPORT**: Role-Based Error Reporting. When set, this bit indicates that the function implements the functionality originally defined in the Error Reporting ECN for PCI Express Base Specification, Revision 1.0a, and later incorporated into PCI Express Base Specification, Revision 1.1. This bit must be set by all functions conforming to the ECN, PCI Express Base Specification, Revision 1.1., or subsequent PCI Express Base Specification revisions. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
14:6	RO	0x0	**RSVDP_6**: Reserved for future use.

Bit	R/W	Reset	Description
5	RW	0x0	**PCIE_CAP_EXT_TAG_SUPP**: Extended Tag Field Supported. This bit, in combination with the 10-Bit Tag Requester Supported bit in the Device Capabilities 2 register, indicates the maximum supported size of the Tag field as a Requester. This bit must be set if the 10-Bit Tag Requester Supported bit is set. Note: 8-bit Tag field generation must be enabled by the Extended Tag Field Enable bit in the Device Control register of the Requester Function before 8-bit Tags can be generated by the Requester. See Section 2.2.6.2 of PCI Express Base Specification for interactions with enabling the use of 10-Bit Tags. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
4:3	RW	0x0	**PCIE_CAP_PHANTOM_FUNC_SUPPORT**: Phantom Functions Supported. This field indicates the support for use of unclaimed function numbers to extend the number of outstanding transactions allowed by logically combining unclaimed function numbers (called Phantom Functions) with the Tag identifier (see Section 2.2.6.2 of PCI Express Base Specification for a description of Tag Extensions). With every Function in an ARI Device, the Phantom Functions Supported field must be set to 00b. The remainder of this field description applies only to non-ARI Multi-Function Devices. This field indicates the number of most significant bits of the Function Number portion of Requester ID that are logically combined with the Tag identifier. Note: Phantom Function support for the function must be enabled by the Phantom Functions Enable field in the Device Control register before the Function is permitted to use the Function Number field in the Requester ID for Phantom Functions. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
2:0	RW	0x1	**PCIE_CAP_MAX_PAYLOAD_SIZE**: Max_Payload_Size Supported. This field indicates the maximum payload size that the function can support for TLPs. All encodings other than the defined encodings are reserved. The functions of a Multi-Function Device are permitted to report different values for this field. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PCIE_CAP_DEVICE_CONTROL_DEVICE_STATUS_0

where <*i*> = 1, 4, 8.

Description: This register controls PCI Express device specific parameters and provides information about PCI Express device (function) specific parameters.

PCIE_X1_RC_PF0_PCIE_CAP_DEVICE_CONTROL_DEVICE_STATUS_0

PCIE_X4_RC_PF0_PCIE_CAP_DEVICE_CONTROL_DEVICE_STATUS_0

PCIE_X8_RC_PF0_PCIE_CAP_DEVICE_CONTROL_DEVICE_STATUS_0

Offset: 0x78
Read/Write: See table below

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00102810 (0b0000,0000,0001,0000,x010,1000,0001,0000)

Bit	R/W	Reset	Description
31:22	RO	0x0	**RSVDP_22**: Reserved for future use.
21	RO	0x0	**PCIE_CAP_TRANS_PENDING**: Transactions Pending. Root and Switch Ports: The controller hardwires this bit to 0b.
20	RO	0x1	**PCIE_CAP_AUX_POWER_DETECTED**: AUX Power Detected. Functions that require Aux power report this bit as set if Aux power is detected by the function. This bit is derived by sampling the sys_aux_pwr_det input.
19	RW	0x0	**PCIE_CAP_UNSUPPORTED_REQ_DETECTED**: Unsupported Request Detected. This bit indicates that the function received an Unsupported Request. Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. For a Multi-Function Device, each function indicates status of errors as perceived by the respective function.
18	RW	0x0	**PCIE_CAP_FATAL_ERR_DETECTED**: Fatal Error Detected. This bit indicates status of Fatal errors detected. Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. For a Multi-Function device, each function indicates status of errors as perceived by the respective Function. For Functions supporting Advanced Error Handling, errors are logged in this register regardless of the settings of the Uncorrectable Error Mask register.
17	RW	0x0	**PCIE_CAP_NON_FATAL_ERR_DETECTED**: Non-Fatal Error Detected. This bit indicates status of Non-fatal errors detected. Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. For a Multi-Function device, each function indicates status of errors as perceived by the respective Function. For functions supporting Advanced Error Handling, errors are logged in this register regardless of the settings of the Uncorrectable Error Mask register.
16	RW	0x0	**PCIE_CAP_CORR_ERR_DETECTED**: Correctable Error Detected. This bit indicates status of correctable errors detected. Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. For a Multi-Function device, each function indicates status of errors as perceived by the respective function. For functions supporting Advanced Error Handling, errors are logged in this register regardless of the settings of the Correctable Error Mask register.
14:12	RW	0x2	**PCIE_CAP_MAX_READ_REQ_SIZE**: Max_Read_Request_Size. This field sets the maximum Read Request size for the function as a Requester. The function must not generate Read Requests with a size exceeding the set value. For functions that do not generate Read Requests larger than 128 bytes and functions that do not generate Read Requests on their own behalf, the controller implements this field as Read Only (RO) with a value of 000b.

Bit	R/W	Reset	Description
11	RW	0x1	**PCIE_CAP_EN_NO_SNOOP**: Enable No Snoop. If this bit is set, the function is permitted to Set the No Snoop bit in the Requester Attributes of transactions it initiates that do not require hardware enforced cache coherency (see section 2.2.6.5 in PCI Express Base Specification). Note: Setting this bit to 1b should not cause a function to set the No Snoop attribute on all transactions that it initiates. Even when this bit is set, a function is only permitted to set the No Snoop attribute on a transaction when it can guarantee that the address of the transaction is not stored in any cache in the system. The controller hardwires this bit 0b if a function would never set the No Snoop attribute in transactions it initiates. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R
10	RW	0x0	**PCIE_CAP_AUX_POWER_PM_EN**: Aux Power PM Enable. This bit is derived by sampling the sys_aux_pwr_det input. When set this bit, enables a function to draw Aux power independent of PME Aux power. Functions that require Aux power on legacy operating systems should continue to indicate PME Aux power requirements. Aux power is allocated as requested in the Aux_Current field of the Power Management Capabilities register (PMC), independent of the PME_En bit in the Power Management Control/Status register (PMCSR). For Multi-Function devices, a component is allowed to draw Aux power if at least one of the functions has this bit set. Note: Functions that consume Aux power must preserve the value of this sticky register when Aux power is available. In such functions, this bit is not modified by Conventional Reset. For functions that do not implement this capability, the controller hardwires this bit to 0b. Note: This register field is sticky.
9	RO	0x0	**PCIE_CAP_PHANTOM_FUNC_EN**: Phantom Functions Enable. This bit, in combination with the 10-Bit Tag Requester Enable bit in the Device Control 2 register, determines how many Tag field bits a Requester is permitted to use. When the 10-Bit Tag Requester Enable bit is clear, - If this bit is set, it enables a function to use unclaimed functions as Phantom functions to extend the number of outstanding transaction identifiers - If this bit is clear, the function is not allowed to use Phantom functions For more details, see section 2.2.6.2 of PCI Express Base Specification. Software should not change the value of this bit while the function has outstanding Non-Posted Requests; otherwise, the result is undefined. For functions that do not implement this capability, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: DEVICE_CAPABILITIES_REG.PCIE_CAP_PHANTOM_FUNC_SUPPORT ? RW : RO
8	RO	0x0	**PCIE_CAP_EXT_TAG_EN**: Extended Tag Field Enable. This bit, in combination with the 10-Bit Tag Requester Enable bit in the Device Control 2 register, determines how many Tag field bits a Requester is permitted to use. When the 10-Bit Tag Requester Enable bit is clear, - If the Extended Tag Field Enable bit is set, the function is permitted to use an 8-bit Tag field as a Requester - If the Extended Tag Field Enable bit is clear, the Function is restricted to a 5-bit Tag field See section 2.2.6.2 of PCI Express Base Specification for required behavior when the 10-Bit Tag Requester Enable bit is set. If software changes the value of the Extended Tag Field Enable bit while the function has outstanding Non-Posted Requests, the result is undefined. For functions that do not implement this capability, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: DEVICE_CAPABILITIES_REG.PCIE_CAP_EXT_TAG_SUPP ? RW : RO

Bit	R/W	Reset	Description
7:5	RW	0x0	**PCIE_CAP_MAX_PAYLOAD_SIZE_CS**: Max_Payload_Size. This field sets maximum TLP payload size for the Function. As a Receiver, the Function must handle TLPs as large as the set value. As a Transmitter, the Function must not generate TLPs exceeding the set value. Permissible values that can be programmed are indicated by the Max_Payload_Size Supported field (PCIE_CAP_MAX_PAYLOAD_SIZE) in the Device Capabilities (DEVICE_CAPABILITIES_REG) register (for more details, see section 7.5.3.3 of PCI Express Base Specification). This field sets the maximum Read Request size for the function as a Requester. The function must not generate Read Requests with a size exceeding the set value. For Functions that support only the 128-byte max payload size, the controller hardwires this field to 000b. System software is not required to program the same value for this field for all the Functions of a Multi-Function device (for more details, see section 2.2.2 of PCI Express Base Specification). For ARI Devices, Max_Payload_Size is determined solely by the setting in Function0. The settings in the other Functions always return whatever value software programmed for each, but otherwise are ignored by the component.
4	RW	0x1	**PCIE_CAP_EN_REL_ORDER**: Enable Relaxed Ordering. If this bit is set, the function is permitted to set the Relaxed Ordering bit in the Attributes field of transactions it initiates that do not require strong write ordering (for more details, see section 2.2.6.4 and section 2.4 of PCI Express Base Specification). For a function that never sets the Relaxed Ordering attribute in transactions it initiates as a Requester, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
3	RW	0x0	**PCIE_CAP_UNSUPPORT_REQ_REP_EN**: Unsupported Request Reporting Enable. This bit, in conjunction with other bits, controls the signaling of Unsupported Request Errors by sending error Messages (for more details, see section 6.2.5 and section 6.2.6 of PCI Express Base Specification). For a Multi-Function Device, this bit controls error reporting for each Function from point-of-view of the respective Function.
2	RW	0x0	**PCIE_CAP_FATAL_ERR_REPORT_EN**: Fatal Error Reporting Enable. This bit, in conjunction with other bits, controls sending ERR_FATAL Messages (for more details, see section 6.2.5 and section 6.2.6 of of PCI Express Base Specification). For a Multi-Function device, this bit controls error reporting for each function from point-of-view of the respective function. For a Root Port, the reporting of Fatal errors is internal to the root. No external ERR_FATAL Message is generated.
1	RW	0x0	**PCIE_CAP_NON_FATAL_ERR_REPORT_EN**: Non-Fatal Error Reporting Enable. This bit, in conjunction with other bits, controls sending ERR_NONFATAL Messages (for more details, see section 6.2.5 and Section 6.2.6 of PCI Express Base Specification). For a Multi-Function Device, this bit controls error reporting for each function from point-of-view of the respective Function. For a Root Port, the reporting of Non-fatal errors is internal to the root. No external ERR_NONFATAL Message is generated.
0	RW	0x0	**PCIE_CAP_CORR_ERR_REPORT_EN**: Correctable Error Reporting Enable. This bit, in conjunction with other bits, controls sending ERR_COR Messages (for more details, see section 6.2.5, section 6.2.6, and section 6.2.10.2 of PCI Express Base Specification). For a Multi-Function device, this bit controls error reporting for each function from point-of-view of the respective function. For a Root Port, the reporting of correctable errors is internal to the root. No external ERR_COR Message is generated.

PCIE_X<*i*>_RC_PF0_PCIE_CAP_LINK_CAPABILITIES_REG_0

where <*i*> = 1, 4, 8.

Description: The Link Capabilities register identifies PCI Express Link specific capabilities.

Offset: 0x7c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_PCIE_CAP_LINK_CAPABILITIES_REG_0

Reset: 0x007b4c14 (0b0000,0000,0111,1011,0100,1100,0001,0100)

Bit	R/W	Reset	Description
31:24	RW	0x0	**PCIE_CAP_PORT_NUM**: Port Number. This field indicates the PCI Express Port number for the given PCI Express Link. Multi-Function Devices associated with an Upstream Port must report the same value in this field for all functions. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
23	RO	0x0	**RSVDP_23**: Reserved for future use.
22	RW	0x1	**PCIE_CAP_ASPM_OPT_COMPLIANCE**: ASPM Optionality Compliance. This field must be set to 1b in all functions. Components implemented against certain earlier versions of this specification will have this bit set to 0b. Software is permitted to use the value of this bit to help determine whether to enable ASPM or whether to run ASPM compliance tests. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
21	RW	0x1	**PCIE_CAP_LINK_BW_NOT_CAP**: Link Bandwidth Notification Capability. A value of 1b indicates support for the Link Bandwidth Notification status and interrupt mechanisms. This capability is required for all Root Ports and Switch Downstream Ports supporting Links wider than x1 and/or multiple Link speeds. This field is not applicable and is Reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches. For functions that do not implement the Link Bandwidth Notification Capability the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

Bit	R/W	Reset	Description
20	RO	0x1	**PCIE_CAP_DLL_ACTIVE_REP_CAP**: Data Link Layer Link Active Reporting Capable. For a Downstream Port, the controller hardwires this bit to 1b if the component supports the optional capability of reporting the DL_Active state of the Data Link Control and Management State Machine. For a hot-plug capable Downstream Port (as indicated by the Hot-Plug Capable bit of the Slot Capabilities register) or a Downstream Port that supports Link speeds greater than 5.0 GT/s, the controller hardwires this bit to 1b. For Upstream Ports and components that do not support this optional capability, the controller hardwires this bit to 0b.
19	RW	0x1	**PCIE_CAP_SURPRISE_DOWN_ERR_REP_CAP**: Surprise Down Error Reporting Capable. For a Downstream Port, this bit must be set if the component supports the optional capability of detecting and reporting a Surprise Down error condition. For Upstream Ports and components that do not support this optional capability, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
18	RO	0x0	**PCIE_CAP_CLOCK_POWER_MAN**: Clock Power Management. For Upstream Ports, a value of 1b in this bit indicates that the component tolerates the removal of any reference clock(s) via the "clock request" (CLKREQ#) mechanism when the Link is in the L1 and L2/L3 Ready Link states. A value of 0b indicates the component does not have this capability and that reference clock(s) must not be removed in these Link states. L1 PM Substates defines other semantics for the CLKREQ# signal, which are managed independently of Clock Power Management. This Capability is applicable only in form factors that support "clock request" (CLKREQ#) capability. For a Multi-Function device associated with an Upstream Port, each Function indicates its capability independently. Power Management configuration software must only permit reference clock removal if all functions of the Multi-Function device indicate a 1b in this bit. For ARI Devices, all Functions must indicate the same value in this bit. For Downstream Ports, the controller hardwires this bit to 0b. Note: This register field is sticky.
17:15	RW	0x6	**PCIE_CAP_L1_EXIT_LATENCY**: L1 Exit Latency. This field indicates the L1 exit latency for the given PCI Express Link. The value reported indicates the length of time this Port requires to complete transition from ASPM L1 to L0. If ASPM L1 is not supported, the value is undefined. Note: Exit latencies may be influenced by PCI Express reference clock configuration depending upon whether a component uses a common or separate reference clock. Multi-Function Devices associated with an Upstream Port must report the same value in this field for all functions. There are two each of these register fields, this one and a shadow one at the same address. The Common Clock bit (PCIE_CAP_COMMON_CLK_CONFIG) of the Link Control Register (LINK_CONTROL_LINK_STATUS_REG) determines which one is used by the controller and which one is accessed by a read request. Common Clock operation is supported (possible) in the controller when one or more of the following expressions is true: - CX_NFTS !=CX_COMM_NFTS - DEFAULT_L0S_EXIT_LATENCY !=DEFAULT_COMM_L0S_EXIT_LATENCY - DEFAULT_L1_EXIT_LATENCY !=DEFAULT_COMM_L1_EXIT_LATENCY Common Clock operation is enabled in the controller when you set the Common Clock bit (PCIE_CAP_COMMON_CLK_CONFIG) of the Link Control Register (LINK_CONTROL_LINK_STATUS_REG). The assertion of CS2 (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) is required to write to the shadow field at this location. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

Bit	R/W	Reset	Description
14:12	RW	0x4	**PCIE_CAP_L0S_EXIT_LATENCY**: L0s Exit Latency. This field indicates the L0s exit latency for the given PCI Express Link. The value reported indicates the length of time this Port requires to complete transition from L0s to L0. If L0s is not supported, the value is undefined; however, see the Implementation Note "Potential Issues With Legacy Software When L0s is Not Supported" in section 5.4.1.1 of PCI Express Base Specification for the recommended value. Note: Exit latencies may be influenced by PCI Express reference clock configuration depending upon whether a component uses a common or separate reference clock. Multi-Function Devices associated with an Upstream Port must report the same value in this field for all functions. There are two each of these register fields, this one and a shadow one at the same address. The Common Clock bit (PCIE_CAP_COMMON_CLK_CONFIG) of the Link Control Register (LINK_CONTROL_LINK_STATUS_REG) determines which one is used by the controller and which one is accessed by a read request. Common Clock operation is supported (possible) in the controller when one or more of the following expressions is true: - CX_NFTS ! =CX_COMM_NFTS - DEFAULT_L0S_EXIT_LATENCY ! =DEFAULT_COMM_L0S_EXIT_LATENCY - DEFAULT_L1_EXIT_LATENCY ! =DEFAULT_COMM_L1_EXIT_LATENCY Common Clock operation is enabled in the controller when you set the Common Clock bit (PCIE_CAP_COMMON_CLK_CONFIG) of the Link Control Register (LINK_CONTROL_LINK_STATUS_REG). The assertion of CS2 (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) is required to write to the shadow field at this location. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
11:10	RW	0x3	**PCIE_CAP_ACTIVE_STATE_LINK_PM_SUPPORT**: Active State Power Management (ASPM) Support. This field indicates the level of ASPM supported on the given PCI Express Link. For more details on ASPM support requirements, see section 5.4.1 of PCI Express Base Specification. Multi-Function devices associated with an Upstream Port must report the same value in this field for all functions. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
9:4	RW	0x1	**PCIE_CAP_MAX_LINK_WIDTH**: Maximum Link Width. This field indicates the maximum Link width (xN - corresponding to N Lanes) implemented by the component. This value is permitted to exceed the number of Lanes routed to the slot (Downstream Port), adapter connector (Upstream Port), or in the case of component-to-component connections, the actual wired connection width. All encodings other than the defined encodings are reserved. Multi-Function devices associated with an Upstream Port must report the same value in this field for all functions. In M-PCIe mode, the reset and dynamic values of this field are calculated by the controller. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
3:0	RW	0x4	**PCIE_CAP_MAX_LINK_SPEED**: Max Link Speed. This field indicates the maximum Link speed of the associated Port. The encoded value specifies a bit location in the Supported Link Speeds Vector (in the Link Capabilities 2 register) that corresponds to the maximum Link speed. All encodings other than the defined encodings are reserved. Multi-Function Devices associated with an Upstream Port must report the same value in this field for all functions. In M-PCIe mode, the reset and dynamic values of this field are calculated by the controller. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

PCIE_X4_RC_PF0_PCIE_CAP_LINK_CAPABILITIES_REG_0

Reset: 0x007b4c44 (0b0000,0000,0111,1011,0100,1100,0100,0100)

Bit	R/W	Reset	Description
Bit	**R/W**	**Reset**	**Description**
31:24	RW	0x0	**PCIE_CAP_PORT_NUM**: Port Number. This field indicates the PCI Express Port number for the given PCI Express Link. Multi-Function Devices associated with an Upstream Port must report the same value in this field for all functions. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
23	RO	0x0	**RSVDP_23**: Reserved for future use.
22	RW	0x1	**PCIE_CAP_ASPM_OPT_COMPLIANCE**: ASPM Optionality Compliance. This field must be set to 1b in all functions. Components implemented against certain earlier versions of this specification will have this bit set to 0b. Software is permitted to use the value of this bit to help determine whether to enable ASPM or whether to run ASPM compliance tests. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
21	RW	0x1	**PCIE_CAP_LINK_BW_NOT_CAP**: Link Bandwidth Notification Capability. A value of 1b indicates support for the Link Bandwidth Notification status and interrupt mechanisms. This capability is required for all Root Ports and Switch Downstream Ports supporting Links wider than x1 and/or multiple Link speeds. This field is not applicable and is Reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches. For functions that do not implement the Link Bandwidth Notification Capability the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
20	RO	0x1	**PCIE_CAP_DLL_ACTIVE_REP_CAP**: Data Link Layer Link Active Reporting Capable. For a Downstream Port, the controller hardwires this bit to 1b if the component supports the optional capability of reporting the DL_Active state of the Data Link Control and Management State Machine. For a hot-plug capable Downstream Port (as indicated by the Hot-Plug Capable bit of the Slot Capabilities register) or a Downstream Port that supports Link speeds greater than 5.0 GT/s, the controller hardwires this bit to 1b. For Upstream Ports and components that do not support this optional capability, the controller hardwires this bit to 0b.
19	RW	0x1	**PCIE_CAP_SURPRISE_DOWN_ERR_REP_CAP**: Surprise Down Error Reporting Capable. For a Downstream Port, this bit must be set if the component supports the optional capability of detecting and reporting a Surprise Down error condition. For Upstream Ports and components that do not support this optional capability, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

Bit	R/W	Reset	Description
Bit	**R/W**	**Reset**	**Description**
18	RO	0x0	**PCIE_CAP_CLOCK_POWER_MAN**: Clock Power Management. For Upstream Ports, a value of 1b in this bit indicates that the component tolerates the removal of any reference clock(s) via the "clock request" (CLKREQ#) mechanism when the Link is in the L1 and L2/L3 Ready Link states. A value of 0b indicates the component does not have this capability and that reference clock(s) must not be removed in these Link states. L1 PM Substates defines other semantics for the CLKREQ# signal, which are managed independently of Clock Power Management. This Capability is applicable only in form factors that support "clock request" (CLKREQ#) capability. For a Multi-Function device associated with an Upstream Port, each Function indicates its capability independently. Power Management configuration software must only permit reference clock removal if all functions of the Multi-Function device indicate a 1b in this bit. For ARI Devices, all Functions must indicate the same value in this bit. For Downstream Ports, the controller hardwires this bit to 0b. Note: This register field is sticky.
17:15	RW	0x6	**PCIE_CAP_L1_EXIT_LATENCY**: L1 Exit Latency. This field indicates the L1 exit latency for the given PCI Express Link. The value reported indicates the length of time this Port requires to complete transition from ASPM L1 to L0. If ASPM L1 is not supported, the value is undefined. Note: Exit latencies may be influenced by PCI Express reference clock configuration depending upon whether a component uses a common or separate reference clock. Multi-Function Devices associated with an Upstream Port must report the same value in this field for all functions. There are two each of these register fields, this one and a shadow one at the same address. The Common Clock bit (PCIE_CAP_COMMON_CLK_CONFIG) of the Link Control Register (LINK_CONTROL_LINK_STATUS_REG) determines which one is used by the controller and which one is accessed by a read request. Common Clock operation is supported (possible) in the controller when one or more of the following expressions is true: - CX_NFTS !=CX_COMM_NFTS - DEFAULT_L0S_EXIT_LATENCY !=DEFAULT_COMM_L0S_EXIT_LATENCY - DEFAULT_L1_EXIT_LATENCY !=DEFAULT_COMM_L1_EXIT_LATENCY Common Clock operation is enabled in the controller when you set the Common Clock bit (PCIE_CAP_COMMON_CLK_CONFIG) of the Link Control Register (LINK_CONTROL_LINK_STATUS_REG). The assertion of CS2 (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) is required to write to the shadow field at this location. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

Bit	R/W	Reset	Description
Bit	**R/W**	**Reset**	**Description**
14:12	RW	0x4	**PCIE_CAP_L0S_EXIT_LATENCY**: L0s Exit Latency. This field indicates the L0s exit latency for the given PCI Express Link. The value reported indicates the length of time this Port requires to complete transition from L0s to L0. If L0s is not supported, the value is undefined; however, see the Implementation Note "Potential Issues With Legacy Software When L0s is Not Supported" in section 5.4.1.1 of PCI Express Base Specification for the recommended value. Note: Exit latencies may be influenced by PCI Express reference clock configuration depending upon whether a component uses a common or separate reference clock. Multi-Function Devices associated with an Upstream Port must report the same value in this field for all functions. There are two each of these register fields, this one and a shadow one at the same address. The Common Clock bit (PCIE_CAP_COMMON_CLK_CONFIG) of the Link Control Register (LINK_CONTROL_LINK_STATUS_REG) determines which one is used by the controller and which one is accessed by a read request. Common Clock operation is supported (possible) in the controller when one or more of the following expressions is true: - CX_NFTS ! =CX_COMM_NFTS - DEFAULT_L0S_EXIT_LATENCY ! =DEFAULT_COMM_L0S_EXIT_LATENCY - DEFAULT_L1_EXIT_LATENCY ! =DEFAULT_COMM_L1_EXIT_LATENCY Common Clock operation is enabled in the controller when you set the Common Clock bit (PCIE_CAP_COMMON_CLK_CONFIG) of the Link Control Register (LINK_CONTROL_LINK_STATUS_REG). The assertion of CS2 (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) is required to write to the shadow field at this location. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
11:10	RW	0x3	**PCIE_CAP_ACTIVE_STATE_LINK_PM_SUPPORT**: Active State Power Management (ASPM) Support. This field indicates the level of ASPM supported on the given PCI Express Link. For more details on ASPM support requirements, see section 5.4.1 of PCI Express Base Specification. Multi-Function devices associated with an Upstream Port must report the same value in this field for all functions. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
9:4	RW	0x4	**PCIE_CAP_MAX_LINK_WIDTH**: Maximum Link Width. This field indicates the maximum Link width (xN - corresponding to N Lanes) implemented by the component. This value is permitted to exceed the number of Lanes routed to the slot (Downstream Port), adapter connector (Upstream Port), or in the case of component-to-component connections, the actual wired connection width. All encodings other than the defined encodings are reserved. Multi-Function devices associated with an Upstream Port must report the same value in this field for all functions. In M-PCIe mode, the reset and dynamic values of this field are calculated by the controller. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
3:0	RW	0x4	**PCIE_CAP_MAX_LINK_SPEED**: Max Link Speed. This field indicates the maximum Link speed of the associated Port. The encoded value specifies a bit location in the Supported Link Speeds Vector (in the Link Capabilities 2 register) that corresponds to the maximum Link speed. All encodings other than the defined encodings are reserved. Multi-Function Devices associated with an Upstream Port must report the same value in this field for all functions. In M-PCIe mode, the reset and dynamic values of this field are calculated by the controller. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

PCIE_X8_RC_PF0_PCIE_CAP_LINK_CAPABILITIES_REG_0

Reset: 0x007b4c84 (0b0000,0000,0111,1011,0100,1100,1000,0100)

Bit	R/W	Reset	Description
Bit	**R/W**	**Reset**	**Description**
31:24	RW	0x0	**PCIE_CAP_PORT_NUM**: Port Number. This field indicates the PCI Express Port number for the given PCI Express Link. Multi-Function Devices associated with an Upstream Port must report the same value in this field for all functions. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
23	RO	0x0	**RSVDP_23**: Reserved for future use.
22	RW	0x1	**PCIE_CAP_ASPM_OPT_COMPLIANCE**: ASPM Optionality Compliance. This field must be set to 1b in all functions. Components implemented against certain earlier versions of this specification will have this bit set to 0b. Software is permitted to use the value of this bit to help determine whether to enable ASPM or whether to run ASPM compliance tests. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
21	RW	0x1	**PCIE_CAP_LINK_BW_NOT_CAP**: Link Bandwidth Notification Capability. A value of 1b indicates support for the Link Bandwidth Notification status and interrupt mechanisms. This capability is required for all Root Ports and Switch Downstream Ports supporting Links wider than x1 and/or multiple Link speeds. This field is not applicable and is Reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches. For functions that do not implement the Link Bandwidth Notification Capability the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
20	RO	0x1	**PCIE_CAP_DLL_ACTIVE_REP_CAP**: Data Link Layer Link Active Reporting Capable. For a Downstream Port, the controller hardwires this bit to 1b if the component supports the optional capability of reporting the DL_Active state of the Data Link Control and Management State Machine. For a hot-plug capable Downstream Port (as indicated by the Hot-Plug Capable bit of the Slot Capabilities register) or a Downstream Port that supports Link speeds greater than 5.0 GT/s, the controller hardwires this bit to 1b. For Upstream Ports and components that do not support this optional capability, the controller hardwires this bit to 0b.
19	RW	0x1	**PCIE_CAP_SURPRISE_DOWN_ERR_REP_CAP**: Surprise Down Error Reporting Capable. For a Downstream Port, this bit must be set if the component supports the optional capability of detecting and reporting a Surprise Down error condition. For Upstream Ports and components that do not support this optional capability, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

Bit	R/W	Reset	Description
Bit	**R/W**	**Reset**	**Description**
18	RO	0x0	**PCIE_CAP_CLOCK_POWER_MAN**: Clock Power Management. For Upstream Ports, a value of 1b in this bit indicates that the component tolerates the removal of any reference clock(s) via the "clock request" (CLKREQ#) mechanism when the Link is in the L1 and L2/L3 Ready Link states. A value of 0b indicates the component does not have this capability and that reference clock(s) must not be removed in these Link states. L1 PM Substates defines other semantics for the CLKREQ# signal, which are managed independently of Clock Power Management. This Capability is applicable only in form factors that support "clock request" (CLKREQ#) capability. For a Multi-Function device associated with an Upstream Port, each Function indicates its capability independently. Power Management configuration software must only permit reference clock removal if all functions of the Multi-Function device indicate a 1b in this bit. For ARI Devices, all Functions must indicate the same value in this bit. For Downstream Ports, the controller hardwires this bit to 0b. Note: This register field is sticky.
17:15	RW	0x6	**PCIE_CAP_L1_EXIT_LATENCY**: L1 Exit Latency. This field indicates the L1 exit latency for the given PCI Express Link. The value reported indicates the length of time this Port requires to complete transition from ASPM L1 to L0. If ASPM L1 is not supported, the value is undefined. Note: Exit latencies may be influenced by PCI Express reference clock configuration depending upon whether a component uses a common or separate reference clock. Multi-Function Devices associated with an Upstream Port must report the same value in this field for all functions. There are two each of these register fields, this one and a shadow one at the same address. The Common Clock bit (PCIE_CAP_COMMON_CLK_CONFIG) of the Link Control Register (LINK_CONTROL_LINK_STATUS_REG) determines which one is used by the controller and which one is accessed by a read request. Common Clock operation is supported (possible) in the controller when one or more of the following expressions is true: - CX_NFTS !=CX_COMM_NFTS - DEFAULT_L0S_EXIT_LATENCY !=DEFAULT_COMM_L0S_EXIT_LATENCY - DEFAULT_L1_EXIT_LATENCY !=DEFAULT_COMM_L1_EXIT_LATENCY Common Clock operation is enabled in the controller when you set the Common Clock bit (PCIE_CAP_COMMON_CLK_CONFIG) of the Link Control Register (LINK_CONTROL_LINK_STATUS_REG). The assertion of CS2 (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) is required to write to the shadow field at this location. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

Bit	R/W	Reset	Description
Bit	**R/W**	**Reset**	**Description**
14:12	RW	0x4	**PCIE_CAP_L0S_EXIT_LATENCY**: L0s Exit Latency. This field indicates the L0s exit latency for the given PCI Express Link. The value reported indicates the length of time this Port requires to complete transition from L0s to L0. If L0s is not supported, the value is undefined; however, see the Implementation Note "Potential Issues With Legacy Software When L0s is Not Supported" in section 5.4.1.1 of PCI Express Base Specification for the recommended value. Note: Exit latencies may be influenced by PCI Express reference clock configuration depending upon whether a component uses a common or separate reference clock. Multi-Function Devices associated with an Upstream Port must report the same value in this field for all functions. There are two each of these register fields, this one and a shadow one at the same address. The Common Clock bit (PCIE_CAP_COMMON_CLK_CONFIG) of the Link Control Register (LINK_CONTROL_LINK_STATUS_REG) determines which one is used by the controller and which one is accessed by a read request. Common Clock operation is supported (possible) in the controller when one or more of the following expressions is true: - CX_NFTS ! =CX_COMM_NFTS - DEFAULT_L0S_EXIT_LATENCY ! =DEFAULT_COMM_L0S_EXIT_LATENCY - DEFAULT_L1_EXIT_LATENCY ! =DEFAULT_COMM_L1_EXIT_LATENCY Common Clock operation is enabled in the controller when you set the Common Clock bit (PCIE_CAP_COMMON_CLK_CONFIG) of the Link Control Register (LINK_CONTROL_LINK_STATUS_REG). The assertion of CS2 (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) is required to write to the shadow field at this location. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
11:10	RW	0x3	**PCIE_CAP_ACTIVE_STATE_LINK_PM_SUPPORT**: Active State Power Management (ASPM) Support. This field indicates the level of ASPM supported on the given PCI Express Link. For more details on ASPM support requirements, see section 5.4.1 of PCI Express Base Specification. Multi-Function devices associated with an Upstream Port must report the same value in this field for all functions. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
9:4	RW	0x8	**PCIE_CAP_MAX_LINK_WIDTH**: Maximum Link Width. This field indicates the maximum Link width (xN - corresponding to N Lanes) implemented by the component. This value is permitted to exceed the number of Lanes routed to the slot (Downstream Port), adapter connector (Upstream Port), or in the case of component-to-component connections, the actual wired connection width. All encodings other than the defined encodings are reserved. Multi-Function devices associated with an Upstream Port must report the same value in this field for all functions. In M-PCIe mode, the reset and dynamic values of this field are calculated by the controller. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
3:0	RW	0x4	**PCIE_CAP_MAX_LINK_SPEED**: Max Link Speed. This field indicates the maximum Link speed of the associated Port. The encoded value specifies a bit location in the Supported Link Speeds Vector (in the Link Capabilities 2 register) that corresponds to the maximum Link speed. All encodings other than the defined encodings are reserved. Multi-Function Devices associated with an Upstream Port must report the same value in this field for all functions. In M-PCIe mode, the reset and dynamic values of this field are calculated by the controller. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PCIE_CAP_LINK_CONTROL_LINK_STATUS_REG_0

where <*i*> = 1, 4, 8.

Description: This register controls and provides information about PCI Express Link specific parameters.

PCIE_X1_RC_PF0_PCIE_CAP_LINK_CONTROL_LINK_STATUS_REG_0

PCIE_X4_RC_PF0_PCIE_CAP_LINK_CONTROL_LINK_STATUS_REG_0

PCIE_X8_RC_PF0_PCIE_CAP_LINK_CONTROL_LINK_STATUS_REG_0

Offset: 0x80
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x10110000 (0b0001,0000,0001,0001,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PCIE_CAP_LINK_AUTO_BW_STATUS**: Link Autonomous Bandwidth Status. This bit is set by hardware to indicate that hardware has autonomously changed Link speed or width, without the Port transitioning through DL_Down status, for reasons other than to attempt to correct unreliable Link operation. This bit must be set if the Physical Layer reports a speed or width change was initiated by the Downstream component that was indicated as an autonomous change. The write value is gated with the PCIE_CAP_LINK_BW_NOT_CAP field in LINK_CAPABILITIES_REG. This bit is not applicable and is Reserved for Endpoints, PCI Express-to-PCI/PCI-X bridges, and Upstream Ports of Switches. For functions that do not implement the Link Bandwidth Notification Capability, the controller hardwires this bit to 0b.
30	RW	0x0	**PCIE_CAP_LINK_BW_MAN_STATUS**: Link Bandwidth Management Status. This bit is set by hardware to indicate that either of the following has occurred without the Port transitioning through DL_Down status: - A Link retraining has completed following a write of 1b to the Retrain Link bit. Note: This bit is set following any write of 1b to the Retrain Link bit, including when the Link is in the process of retraining for some other reason. - Hardware has changed Link speed or width to attempt to correct unreliable Link operation, either through an LTSSM timeout or a higher level process. This bit must be set if the Physical Layer reports a speed or width change was initiated by the Downstream component that was not indicated as an autonomous change. This bit is not applicable and is Reserved for Endpoints, PCI Express-to-PCI/PCI-X bridges, and Upstream Ports of Switches. For functions that do not implement the Link Bandwidth Notification Capability, the controller hardwires this bit to 0b. The default value of this bit is 0b. The write value is gated with the PCIE_CAP_LINK_BW_NOT_CAP field in LINK_CAPABILITIES_REG.

Bit	R/W	Reset	Description
29	RO	0x0	**PCIE_CAP_DLL_ACTIVE**: Data Link Layer Link Active. This bit indicates the status of the Data Link Control and Management State Machine. It returns a 1b to indicate the DL_Active state, 0b otherwise. This bit must be implemented if the Data Link Layer Link Active Reporting Capable bit is 1b. Otherwise, the controller hardwires it to 0b.
28	RW	0x1	**PCIE_CAP_SLOT_CLK_CONFIG**: Slot Clock Configuration. This bit indicates that the component uses the same physical reference clock that the platform provides on the connector. If the device uses an independent clock irrespective of the presence of a reference clock on the connector, this bit must be clear. For a Multi-Function Device, each Function must report the same value for this bit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
27	RO	0x0	**PCIE_CAP_LINK_TRAINING**: Link Training. This read-only bit indicates that the Physical Layer LTSSM is in the Configuration or Recovery state, or that 1b was written to the Retrain Link bit but Link training has not yet begun. Hardware clears this bit when the LTSSM exits the Configuration/Recovery state. This bit is not applicable and Reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches, and the controller hardwires it to 0b.
26	RO	0x0	**RSVDP_26**: Reserved for future use.
25:20	RO	0x1	**PCIE_CAP_NEGO_LINK_WIDTH**: Negotiated Link Width. This field indicates the negotiated width of the given PCI Express Link. All encodings other than the defined encodings are reserved. The value in this field is undefined when the Link is not up.
19:16	RO	0x1	**PCIE_CAP_LINK_SPEED**: Current Link Speed. This field indicates the negotiated Link speed of the given PCI Express Link. The encoded value specifies a bit location in the Supported Link Speeds Vector (in the Link Capabilities 2 register) that corresponds to the current Link speed. All encodings other than the defined encodings are reserved. The value in this field is undefined when the Link is not up.
15:14	RW	0x0	**PCIE_CAP_DRS_SIGNALING_CONTROL**: DRS Signaling Control. Indicates the mechanism used to report reception of a DRS message. Must be implemented for Downstream Ports with the DRS Supported bit Set in the Link Capabilities 2 Register. Encodings are: If DRS Supported is set, receiving a DRS Message will set DRS Message Received in the Link Status 2 Register but will otherwise have no effect If the DRS Message Received bit in the Link Status 2 Register transitions from 0 to 1, and either MSI or MSI-X is enabled, an MSI or MSI-X interrupt is generated using the vector in Interrupt Message Number (section 7.5.3.2) If the DRS Message Received bit in the Link Status 2 Register transitions from 0 to 1, the Port must send an FRS Message Upstream with the FRS Reason field set to DRS Message Received. Behavior is undefined if this field is set to 10b and the FRS Supported bit in the Device Capabilities 2 Register is Clear. Behavior is undefined if this field is set to 11b. For Downstream Ports with the DRS Supported bit clear in the Link Capabilities 2 register, the controller hardwires this field to 00b. This field is Reserved for Upstream Ports. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: LINK_CAPABILITIES2_REG.DRS_SUPPORTED ? RW : RO
13:12	RO	0x0	**RSVDP_12**: Reserved for future use.

Bit	R/W	Reset	Description
11	RW	0x0	**PCIE_CAP_LINK_AUTO_BW_INT_EN**: Link Autonomous Bandwidth Management Interrupt Enable. When set, this bit enables the generation of an interrupt to indicate that the Link Autonomous Bandwidth Status bit has been set. The write value is gated with the PCIE_CAP_LINK_BW_NOT_CAP field in LINK_CAPABILITIES_REG. This bit is not applicable and is Reserved for Endpoints, PCI Express-to-PCI/PCI-X bridges, and Upstream Ports of Switches. For functions that do not implement the Link Bandwidth Notification Capability, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: LINK_CAPABILITIES_REG.PCIE_CAP_LINK_BW_NOT_CAP ? RW : RO
10	RW	0x0	**PCIE_CAP_LINK_BW_MAN_INT_EN**: Link Bandwidth Management Interrupt Enable. When set, this bit enables the generation of an interrupt to indicate that the Link Bandwidth Management Status bit has been set. The write value is gated with the PCIE_CAP_LINK_BW_NOT_CAP field in LINK_CAPABILITIES_REG. This bit is not applicable and is Reserved for Endpoints, PCI Express-to-PCI/PCI-X bridges, and Upstream Ports of Switches. For functions that do not implement the Link Bandwidth Notification Capability, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: LINK_CAPABILITIES_REG.PCIE_CAP_LINK_BW_NOT_CAP ? RW : RO
9	RW	0x0	**PCIE_CAP_HW_AUTO_WIDTH_DISABLE**: Hardware Autonomous Width Disable. When set, this bit disables hardware from changing the Link width for reasons other than attempting to correct unreliable Link operation by reducing Link width. For a Multi-Function Device associated with an Upstream Port, the bit in Function 0 is of type RW, and only Function 0 controls the component's Link behavior. In all other Functions of that device, this bit is of type RsvdP. For components that do not implement the ability autonomously to change Link width, the ciontroller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
8	RW	0x0	**PCIE_CAP_EN_CLK_POWER_MAN**: Enable Clock Power Management. Applicable only for Upstream Ports and with form factors that support a "Clock Request" (CLKREQ#) mechanism, this bit operates as follows: For a non-ARI Multi-Function Device, power-management-configuration software must only Set this bit if all Functions of the Multi-Function Device indicate a 1b in the Clock Power Management bit of the Link Capabilities register. The component is permitted to use the CLKREQ# signal to power manage Link clock only if this bit is Set for all Functions. For ARI Devices, Clock Power Management is enabled solely by the setting in Function 0. The settings in the other Functions always return whatever value software programmed for each, but otherwise are ignored by the component. The CLKREQ# signal may also be controlled via the L1 PM Substates mechanism. Such control is not affected by the setting of this bit. For Downstream Ports and components that do not support Clock Power Management (as indicated by a 0b value in the Clock Power Management bit of the Link Capabilities register), the controller hardwires this bit to 0b. The write value is gated with the PCIE_CAP_CLOCK_POWER_MAN field in LINK_CAPABILITIES_REG. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: LINK_CAPABILITIES_REG.PCIE_CAP_CLOCK_POWER_MAN ? RWS : ROS Note: This register field is sticky.

Bit	R/W	Reset	Description
7	RW	0x0	**PCIE_CAP_EXTENDED_SYNCH**: Extended Synch. When set, this bit forces the transmission of additional Ordered Sets when exiting the L0s state (see section 4.2.4.5 of PCI Express Base Specification) and when in the Recovery state (see section 4.2.6.4.1 of PCI Express Base Specification). This mode provides external devices (for example, logic analyzers) monitoring the Link time to achieve bit and Symbol lock before the Link enters the L0 state and resumes communication. For Multi-Function devices if any function has this bit set, then the component must transmit the additional Ordered Sets when exiting L0s or when in Recovery.
6	RW	0x0	**PCIE_CAP_COMMON_CLK_CONFIG**: Common Clock Configuration. When set, this bit indicates that this component and the component at the opposite end of this Link are operating with a distributed common reference clock. A value of 0b indicates that this component and the component at the opposite end of this Link are operating with asynchronous reference clock. For non-ARI Multi-Function Devices, software must program the same value for this bit in all Functions. If not all Functions are Set, then the component must as a whole assume that its reference clock is not common with the Upstream component. For ARI Devices, Common Clock Configuration is determined solely by the setting in Function 0. The settings in the other Functions always return whatever value software programmed for each, but otherwise are ignored by the component. Components utilize this common clock configuration information to report the correct L0s and L1 Exit Latencies. After changing the value in this bit in both components on a Link, software must trigger the Link to retrain by writing a 1b to the Retrain Link bit of the Downstream Port.
5	RW	0x0	**PCIE_CAP_RETRAIN_LINK**: Retrain Link. A write of 1b to this bit initiates Link retraining by directing the Physical Layer LTSSM to the Recovery state. If the LTSSM is already in Recovery or Configuration, re-entering Recovery is permitted but not required. If the Port is in DPC when a write of 1b to this bit occurs, the result is undefined. Reads of this bit always return 0b. It is permitted to write 1b to this bit while simultaneously writing modified values to other fields in this register. If the LTSSM is not already in Recovery or Configuration, the resulting Link training must use the modified values. If the LTSSM is already in Recovery or Configuration, the modified values are not required to affect the Link training that's already in progress. This bit is not applicable and is Reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches. This bit always returns 0b when read. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: see description
4	RW	0x0	**PCIE_CAP_LINK_DISABLE**: Link Disable. This bit disables the Link by directing the LTSSM to the Disabled state when set; this bit is Reserved on Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches. Writes to this bit are immediately reflected in the value read from the bit, regardless of actual Link state. After clearing this bit, software must honor timing requirements defined in Section 6.6.1 with respect to the first Configuration Read following a Conventional Reset. In a DSP that supports crosslink, the controller gates the write value with the CROSS_LINK_EN field in PORT_LINK_CTRL_OFF. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: CX_CROSSLINK_ENABLE=1 && PORT_LINK_CTRL_OFF.CROSS_LINK_EN=1||CX_CROSSLINK_ENABLE=0 && dsp=1? RW : RO

Bit	R/W	Reset	Description
3	RW	0x0	**PCIE_CAP_RCB**: Read Completion Boundary (RCB). Root Ports: Indicates the RCB value for the Root Port. Refer to section 2.3.1.1 of PCI Express Base Specification for the definition of the parameter RCB. The controller hardwires this bit for a Root Port and returns its RCB support capabilities. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
2	RO	0x0	**RSVDP_2**: Reserved for future use.
1:0	RW	0x0	**PCIE_CAP_ACTIVE_STATE_LINK_PM_CONTROL**: Active State Power Management (ASPM) Control. This field controls the level of ASPM enabled on the given PCI Express Link. See section 5.4.1.3 of PCI Express Base Specification for requirements on when and how to enable ASPM. Note: "L0s Entry Enabled" enables the Transmitter to enter L0s. If L0s is supported, the Receiver must be capable of entering L0s even when the Transmitter is disabled from entering L0s (00b or 10b). ASPM L1 must be enabled by software in the Upstream component on a Link prior to enabling ASPM L1 in the Downstream component on that Link. When disabling ASPM L1, software must disable ASPM L1 in the Downstream component on a Link prior to disabling ASPM L1 in the Upstream component on that Link. ASPM L1 must only be enabled on the Downstream component if both components on a Link support ASPM L1. For Multi-Function Devices (including ARI Devices), it is recommended that software program the same value for this field in all Functions. For non-ARI Multi-Function Devices, only capabilities enabled in all Functions are enabled for the component as a whole. For ARI Devices, ASPM Control is determined solely by the setting in Function0, regardless of Function 0's D-state. The settings in the other Functions always return whatever value software programmed for each, but otherwise are ignored by the component. Software must not enable L0s in either direction on a given Link unless components on both sides of the Link each support L0s; otherwise, the result is undefined.

PCIE_X<*i*>_RC_PF0_PCIE_CAP_SLOT_CAPABILITIES_REG_0

where <*i*> = 1, 4, 8.

Description: The Slot Capabilities register identifies PCI Express slot specific capabilities.

PCIE_X1_RC_PF0_PCIE_CAP_SLOT_CAPABILITIES_REG_0

PCIE_X4_RC_PF0_PCIE_CAP_SLOT_CAPABILITIES_REG_0

PCIE_X8_RC_PF0_PCIE_CAP_SLOT_CAPABILITIES_REG_0

Offset: 0x84
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:19	0x0	**PCIE_CAP_PHY_SLOT_NUM**: Physical Slot Number. This field indicates the physical slot number attached to this Port. This field must be hardware initialized to a value that assigns a slot number that is unique within the chassis, regardless of the form factor associated with the slot. This field must be initialized to zero for Ports connected to devices that are either integrated on the system board or integrated within the same silicon as the Switch device or Root Port. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
18	0x0	**PCIE_CAP_NO_CMD_CPL_SUPPORT**: No Command Completed Support. When set, this bit indicates that this slot does not generate software notification when an issued command is completed by the Hot-Plug Controller. This bit is only permitted to be set if the hot-plug capable Port is able to accept writes to all fields of the Slot Control register without delay between successive writes. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
17	0x0	**PCIE_CAP_ELECTROMECH_INTERLOCK**: Electromechanical Interlock Present. When set, this bit indicates that an Electromechanical Interlock is implemented on the chassis for this slot. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
16:15	0x0	**PCIE_CAP_SLOT_POWER_LIMIT_SCALE**: Slot Power Limit Scale. Specifies the scale used for the Slot Power Limit Value (for more details, see Section 6.9 of PCI Express Base Specification). This register must be implemented if the Slot Implemented bit is set. Writes to this register also cause the Port to send the Set_Slot_Power_Limit Message. The default value prior to hardware/firmware initialization is 00b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
14:7	0x0	**PCIE_CAP_SLOT_POWER_LIMIT_VALUE**: Slot Power Limit Value. In combination with the Slot Power Limit Scale value, specifies the upper limit on power supplied by the slot (for more detais, see Section 6.9 of PCI Express Base Specification) or by other means to the adapter. Power limit (in Watts) is calculated by multiplying the value in this field by the value in the Slot Power Limit Scale field except when the Slot Power Limit Scale field equals 00b (1.0x) and Slot Power Limit Value exceeds EFh, the alternative encodings defined in Values: are used. Value F3h to FFh are Reserved for Slot Power Limit values above 300 W. This register must be implemented if the Slot Implemented bit is set. Writes to this register also cause the Port to send the Set_Slot_Power_Limit Message. The default value prior to hardware/firmware initialization is 0000 0000b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
6	0x0	**PCIE_CAP_HOT_PLUG_CAPABLE**: Hot-Plug Capable. When set, this bit indicates that this slot is capable of supporting hot-plug operations. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
5	0x0	**PCIE_CAP_HOT_PLUG_SURPRISE**: Hot-Plug Surprise. When set, this bit indicates that an adapter present in this slot might be removed from the system without any prior notification. This is a form factor specific capability. This bit is an indication to the operating system to allow for such removal without impacting continued software operation. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R

Bit	Reset	Description
4	0x0	**PCIE_CAP_POWER_INDICATOR**: Power Indicator Present. When set, this bit indicates that a Power Indicator is electrically controlled by the chassis for this slot. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
3	0x0	**PCIE_CAP_ATTENTION_INDICATOR**: Attention Indicator Present. When set, this bit indicates that an Attention Indicator is electrically controlled by the chassis. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
2	0x0	**PCIE_CAP_MRL_SENSOR**: MRL Sensor Present. When set, this bit indicates that an MRL Sensor is implemented on the chassis for this slot. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
1	0x0	**PCIE_CAP_POWER_CONTROLLER**: Power Controller Present. When set, this bit indicates that a software programmable Power Controller is implemented for this slot/adapter (depending on form factor). Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
0	0x0	**PCIE_CAP_ATTENTION_INDICATOR_BUTTON**: Attention Button Present. When set, this bit indicates that an Attention Button for this slot is electrically controlled by the chassis. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R

PCIE_X<*i*>_RC_PF0_PCIE_CAP_SLOT_CONTROL_SLOT_STATUS_0

where <*i*> = 1, 4, 8.

Description: This register controls and provides information about PCI Express Slot specific parameters.

PCIE_X1_RC_PF0_PCIE_CAP_SLOT_CONTROL_SLOT_STATUS_0

PCIE_X4_RC_PF0_PCIE_CAP_SLOT_CONTROL_SLOT_STATUS_0

PCIE_X8_RC_PF0_PCIE_CAP_SLOT_CONTROL_SLOT_STATUS_0

Offset: 0x88
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x004003c0 (0b0000,0000,0100,0000,00x0,0011,1100,0000)

Bit	R/W	Reset	Description
31:25	RO	0x0	**RSVDP_25**: Reserved for future use.
24	RW	0x0	**PCIE_CAP_DLL_STATE_CHANGED**: Data Link Layer State Changed. This bit is set when the value reported in the Data Link Layer Link Active bit of the Link Status register is changed. In response to a Data Link Layer State Changed event, software must read the Data Link Layer Link Active bit of the Link Status register to determine if the Link is active before initiating configuration cycles to the hot plugged device.
23	RO	0x0	**PCIE_CAP_ELECTROMECH_INTERLOCK_STATUS**: Electromechanical Interlock Status. If an Electromechanical Interlock is implemented, this bit indicates the status of the Electromechanical Interlock.
22	RO	0x1	**PCIE_CAP_PRESENCE_DETECT_STATE**: Presence Detect State. This bit indicates the presence of an adapter in the slot, reflected by the logical "OR" of the Physical Layer in-band presence detect mechanism and, if present, any out-of-band presence detect mechanism defined for the slot's corresponding form factor. Note that the in-band presence detect mechanism requires that power be applied to an adapter for its presence to be detected. Consequently, form factors that require a power controller for hot-plug must implement a physical pin presence detect mechanism. This bit must be implemented on all Downstream Ports that implement slots. For Downstream Ports not connected to slots (where the Slot Implemented bit of the PCI Express Capabilities register is 0b), the controller hardwires this bit to 1b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R
21	RO	0x0	**PCIE_CAP_MRL_SENSOR_STATE**: MRL Sensor State. This bit reports the status of the MRL sensor if implemented. Encodings are define as above.
20	RW	0x0	**PCIE_CAP_CMD_CPLD**: Command Completed. If Command Completed notification is supported (if the No Command Completed Support bit in the Slot Capabilities register is 0b), this bit is set when a hot-plug command has completed and the Hot-Plug Controller is ready to accept a subsequent command. The Command Completed status bit is set as an indication to host software that the Hot-Plug Controller has processed the previous command and is ready to receive the next command; it provides no guarantee that the action corresponding to the command is complete. If Command Completed notification is not supported, the controller hardwires this bit to 0b.
19	RW	0x0	**PCIE_CAP_PRESENCE_DETECTED_CHANGED**: Presence Detect Changed. This bit is set when the value reported in the Presence Detect State bit is changed.
18	RW	0x0	**PCIE_CAP_MRL_SENSOR_CHANGED**: MRL Sensor Changed. If an MRL sensor is implemented, this bit is set when a MRL Sensor state change is detected. If an MRL sensor is not implemented, this bit must not be set.

Bit	R/W	Reset	Description
17	RW	0x0	**PCIE_CAP_POWER_FAULT_DETECTED**: Power Fault Detected. If a Power Controller that supports power fault detection is implemented, this bit issSet when the Power Controller detects a power fault at this slot. Note: Depending on hardware capability, it is possible that a power fault can be detected at any time, independent of the Power Controller Control setting or the occupancy of the slot. If power fault detection is not supported, this bit must not be set.
16	RW	0x0	**PCIE_CAP_ATTENTION_BUTTON_PRESSED**: Attention Button Pressed. If an Attention Button is implemented, this bit is set when the attention button is pressed. If an Attention Button is not supported, this bit must not be set.
15:14	RO	0x0	**RSVDP_14**: Reserved for future use.
12	RW	0x0	**PCIE_CAP_DLL_STATE_CHANGED_EN**: Data Link Layer State Changed Enable. If the Data Link Layer Link Active Reporting capability is 1b, this bit enables software notification when Data Link Layer Link Active bit is changed. If the Data Link Layer Link Active Reporting Capable bit is 0b, this bit is permitted to be read-only with a value of 0b.
11	RW	0x0	**PCIE_CAP_ELECTROMECH_INTERLOCK_CTRL**: Electromechanical Interlock Control. If an Electromechanical Interlock is implemented, a write of 1b to this bit causes the state of the interlock to toggle. A write of 0b to this bit has no effect. A read of this bit always returns a 0b.
10	RW	0x0	**PCIE_CAP_POWER_CONTROLLER_CTRL**: Power Controller Control. If a Power Controller is implemented, this bit when written sets the power state of the slot per the defined encodings. Reads of this bit must reflect the value from the latest write, even if the corresponding hot-plug command is not complete, unless software issues a write, if required to, without waiting for the previous command to complete in which case the read value is undefined. Note: In some cases the power controller may autonomously remove slot power or not respond to a power-up request based on a detected fault condition, independent of the Power Controller Control setting. If the Power Controller Present bit in the Slot Capabilities register is clear, then writes to this bit have no effect and the read value of this bit is undefined.
9:8	RW	0x3	**PCIE_CAP_POWER_INDICATOR_CTRL**: Power Indicator Control. If a Power Indicator is implemented, writes to this field set the Power Indicator to the written state. Reads of this field must reflect the value from the latest write, even if the corresponding hot-plug command is not complete, unless software issues a write without waiting, if required to, for the previous command to complete in which case the read value is undefined. Note: The default value of this field must be one of the non-Reserved values. If the Power Indicator Present bit in the Slot Capabilities register is 0b, this bit is permitted to be read-only with a value of 00b.

Bit	R/W	Reset	Description
7:6	RW	0x3	**PCIE_CAP_ATTENTION_INDICATOR_CTRL**: Attention Indicator Control. If an Attention Indicator is implemented, writes to this field set the Attention Indicator to the written state. Reads of this field must reflect the value from the latest write, even if the corresponding hot-plug command is not complete, unless software issues a write without waiting, if required to, for the previous command to complete in which case the read value is undefined. Note: The default value of this field must be one of the non-Reserved values. If the Attention Indicator Present bit in the Slot Capabilities register is 0b, this bit is permitted to be read-only with a value of 00b.
5	RW	0x0	**PCIE_CAP_HOT_PLUG_INT_EN**: Hot-Plug Interrupt Enable. When set, this bit enables generation of an interrupt on enabled hot-plug events. If the Hot Plug Capable bit in the Slot Capabilities register is clear, this bit is permitted to be read-only with a value of 0b.
4	RW	0x0	**PCIE_CAP_CMD_CPL_INT_EN**: Command Completed Interrupt Enable. If Command Completed notification is supported (if the No Command Completed Support bit in the Slot Capabilities register is 0b), when set, this bit enables software notification when a hot-plug command is completed by the Hot-Plug Controller. If Command Completed notification is not supported, the controller hardwires this bit must to 0b. Write value is gated with PCIE_CAP_NO_CMD_CPL_SUPPORT field in SLOT_CAPABILITIES_REG. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: SLOT_CAPABILITIES_REG.PCIE_CAP_NO_CMD_CPL_SUPPORT ? RO : RW
3	RW	0x0	**PCIE_CAP_PRESENCE_DETECT_CHANGE_EN**: Presence Detect Changed Enable. When set, this bit enables software notification on a presence detect changed event (for more details, see Section 6.7.3 of PCI Express Base Specification). If the Hot-Plug Capable bit in the Slot Capabilities register is 0b, this bit is permitted to be read-only with a value of 0b.
2	RW	0x0	**PCIE_CAP_MRL_SENSOR_CHANGED_EN**: MRL Sensor Changed Enable. When set, this bit enables software notification on a MRL sensor changed event (for more details, see Section 6.7.3 of PCI Express Base Specification). If the MRL Sensor Present bit in the Slot Capabilities register is Clear, this bit is permitted to be read-only with a value of 0b.
1	RW	0x0	**PCIE_CAP_POWER_FAULT_DETECTED_EN**: Power Fault Detected Enable. When set, this bit enables software notification on a power fault event (for more details, see Section 6.7.3 of PCI Express Base Specification). If a Power Controller that supports power fault detection is not implemented, this bit is permitted to be read-only with a value of 0b.
0	RW	0x0	**PCIE_CAP_ATTENTION_BUTTON_PRESSED_EN**: Attention Button Pressed Enable. When set to 1b, this bit enables software notification on an attention button pressed event (for more details, see Section 6.7.3 of PCI Express Base Specification). If the Attention Button Present bit in the Slot Capabilities register is 0b, this bit is permitted to be read-only with a value of 0b.

PCIE_X<*i*>_RC_PF0_PCIE_CAP_ROOT_CONTROL_ROOT_CAPABILITIES_REG_0

where <*i*> = 1, 4, 8.

Description: This register controls PCI Express Root Complex specific parameters and identifies PCI Express Root Port specific capabilities.

PCIE_X1_RC_PF0_PCIE_CAP_ROOT_CONTROL_ROOT_CAPABILITIES_REG_0

PCIE_X4_RC_PF0_PCIE_CAP_ROOT_CONTROL_ROOT_CAPABILITIES_REG_0

PCIE_X8_RC_PF0_PCIE_CAP_ROOT_CONTROL_ROOT_CAPABILITIES_REG_0

Offset: 0x8c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00010000 (0b0000,0000,0000,0001,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:17	RO	0x0	**RSVDP_17**: Reserved for future use.
16	RW	0x1	**PCIE_CAP_CRS_SW_VISIBILITY**: CRS Software Visibility Capable. When set, this bit indicates that the Root Port is capable of returning Configuration Request Retry Status (CRS) Completion Status to software (for more details, see section 2.3.1 of PCI Express Base Specification). Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W (Sticky) else R (Sticky) Note: This register field is sticky.
15:5	RO	0x0	**RSVDP_5**: Reserved for future use.
4	RW	0x0	**PCIE_CAP_CRS_SW_VISIBILITY_EN**: CRS Software Visibility Enable. When set, this bit enables the Root Port to return Configuration Request Retry Status (CRS) Completion Status to software (for more details, see section 2.3.1 of PCI Express Base Specification). For Root Ports that do not implement this capability, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: ROOT_CONTROL_ROOT_CAPABILITIES_REG.PCIE_CAP_CRS_SW_VISIBILITY ? RW : RO
3	RW	0x0	**PCIE_CAP_PME_INT_EN**: PME Interrupt Enable. When set, this bit enables PME interrupt generation upon receipt of a PME Message as reflected in the PME Status bit (for more details, see Table 7-29 of PCI Express Base Specification). A PME interrupt is also generated if the PME Status bit is set when this bit is changed from clear to set.
2	RW	0x0	**PCIE_CAP_SYS_ERR_ON_FATAL_ERR_EN**: System Error on Fatal Error Enable. If set, this bit indicates that a System Error should be generated if a Fatal error (ERR_FATAL) is reported by any of the devices in the Hierarchy Domain associated with this Root Port, or by the Root Port itself. The mechanism for signaling a System Error to the system is system specific.

Bit	R/W	Reset	Description
1	RW	0x0	**PCIE_CAP_SYS_ERR_ON_NON_FATAL_ERR_EN**: System Error on Non-Fatal Error Enable. If set, this bit indicates that a System Error should be generated if a Non-fatal error (ERR_NONFATAL) is reported by any of the devices in the Hierarchy Domain associated with this Root Port, or by the Root Port itself. The mechanism for signaling a System Error to the system is system specific.
0	RW	0x0	**PCIE_CAP_SYS_ERR_ON_CORR_ERR_EN**: System Error on Correctable Error Enable. If set, this bit indicates that a System Error should be generated if a correctable error (ERR_COR) is reported by any of the devices in the Hierarchy Domain associated with this Root Port, or by the Root Port itself. The mechanism for signaling a System Error to the system is system specific.

PCIE_X<*i*>_RC_PF0_PCIE_CAP_ROOT_STATUS_REG_0

where <*i*> = 1, 4, 8.

Description: The Root Status register provides information about PCI Express device specific parameters.

PCIE_X1_RC_PF0_PCIE_CAP_ROOT_STATUS_REG_0

PCIE_X4_RC_PF0_PCIE_CAP_ROOT_STATUS_REG_0

PCIE_X8_RC_PF0_PCIE_CAP_ROOT_STATUS_REG_0

Offset: 0x90
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:18	RO	0x0	**RSVDP_18**: Reserved for future use.
17	RO	0x0	**PCIE_CAP_PME_PENDING**: PME Pending. This bit indicates that another PME is pending when the PME Status bit is s et. When the PME Status bit is cleared by software; the PME is delivered by hardware by setting the PME Status bit again and updating the PME Requester ID field appropriately. The PME Pending bit is cleared by hardware if no more PMEs are pending.

Bit	R/W	Reset	Description
16	RW	0x0	**PCIE_CAP_PME_STATUS**: PME Status. This bit indicates that PME was asserted by the PME Requester indicated in the PME Requester ID field. Subsequent PMEs are kept pending until the status register is cleared by software by writing a 1b.
15:0	RO	0x0	**PCIE_CAP_PME_REQ_ID**: PME Requester ID. This field indicates the PCI Requester ID of the last PME Requester. This field is only valid when the PME Status bit is set.

PCIE_X<*i*>_RC_PF0_PCIE_CAP_DEVICE_CAPABILITIES2_REG_0

where <*i*> = 1, 4, 8.

Description: This register identifies PCI Express device specific capabilities; in addition to the Device Capabilities Register.

PCIE_X1_RC_PF0_PCIE_CAP_DEVICE_CAPABILITIES2_REG_0

PCIE_X4_RC_PF0_PCIE_CAP_DEVICE_CAPABILITIES2_REG_0

PCIE_X8_RC_PF0_PCIE_CAP_DEVICE_CAPABILITIES2_REG_0

Offset: 0x94
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00010c3f (0bx000,0000,xxxx,xx01,xx00,1100,0011,1111)

Bit	R/W	Reset	Description
30:24	RO	0x0	**RSVDP_24**: Reserved for future use.
17	RO	0x0	**PCIE_CAP2_10_BIT_TAG_REQ_SUPPORT**: 10-Bit Tag Requester Supported. If this bit is set, the Function supports 10-Bit Tag Requester capability; otherwise, the Function does not. This bit must not be set if the 10-Bit Tag Completer Supported bit is clear. Note: 10-Bit Tag field generation must be enabled by the 10-Bit Tag Requester Enable bit in the Device Control 2 register of the Requester Function before 10-Bit Tags can be generated by the Requester. For more details, see section 2.2.6.2. of PCI Express Base Specification.
16	RO	0x1	**PCIE_CAP2_10_BIT_TAG_COMP_SUPPORT**: 10-Bit Tag Completer Supported. If this bit is set, the Function supports 10-Bit Tag Completer capability; otherwise, the Function does not. For more details, see section 2.2.6.2. of PCI Express Base Specification.

Bit	R/W	Reset	Description
13	RO	0x0	**PCIE_CAP_TPH_CMPLT_SUPPORT_1**: TPH Completer Supported Bit 1.
12	RO	0x0	**PCIE_CAP_TPH_CMPLT_SUPPORT_0**: TPH Completer Supported Bit 0. Value of this bit along with TPH Completer Supported Bit 1 indicates Completer support for TPH or Extended TPH. Applicable only to Root Ports and Endpoints. For all other Functions, this field is Reserved. When TPH completer support bit 0 is clear. When TPH completer support bit 1 is set. For more details, see section 6.17 of PCI Express Base Specification.
11	RW	0x1	**PCIE_CAP_LTR_SUPP**: LTR Mechanism Supported. A value of 1b indicates support for the optional Latency Tolerance Reporting (LTR) mechanism. Root Ports, Switches and Endpoints are permitted to implement this capability. For a Multi-Function Device associated with an Upstream Port, each Function must report the same value for this bit. For Bridges and other Functions that do not implement this capability, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(Sticky) else R(Sticky) Note: This register field is sticky.
10	RO	0x1	**PCIE_CAP_NO_RO_EN_PR2PR_PAR**: No RO-enabled PR-PR Passing. If this bit is set, the routing element never carries out the passing permitted by Table 2-39 of PCI Express Base Specification entry A2b that is associated with the Relaxed Ordering Attribute field being Set. This bit applies only for Switches and RCs that support peer-to-peer traffic between Root Ports. This bit applies only to Posted Requests being forwarded through the Switch or RC and does not apply to traffic originating or terminating within the Switch or RC itself. All Ports on a Switch or RC must report the same value for this bit. For all other functions, this bit must be 0b.
9	RO	0x0	**PCIE_CAP_128_CAS_CPL_SUPP**: 128-bit CAS Completer Supported. Applicable to Functions with Memory Space BARs as well as all Root Ports; must be 0b otherwise. This bit must be set to 1b if the Function supports this optional capability. For more details, see section 6.15 of PCI Express Base Specification.
8	RO	0x0	**PCIE_CAP_64_ATOMIC_CPL_SUPP**: 64-bit AtomicOp Completer Supported. Applicable to Functions with Memory Space BARs as well as all Root Ports; must be 0b otherwise. Includes FetchAdd, Swap, and CAS AtomicOps. This bit must be set to 1b if the Function supports this optional capability. For more details on additional RC requirements, see section 6.15.3.1 of PCI Express Base Specification.
7	RO	0x0	**PCIE_CAP_32_ATOMIC_CPL_SUPP**: 32-bit AtomicOp Completer Supported. Applicable to Functions with Memory Space BARs as well as all Root Ports; must be 0b otherwise. Includes FetchAdd, Swap, and CAS AtomicOps. This bit must be set to 1b if the Function supports this optional capability. For more details on additional RC requirements, see section 6.15.3.1 of PCI Express Base Specification.
6	RO	0x0	**PCIE_CAP_ATOMIC_ROUTING_SUPP**: AtomicOp Routing Supported. Applicable only to Switch Upstream Ports, Switch Downstream Ports, and Root Ports; must be 0b for other Function types. This bit must be set to 1b if the Port supports this optional capability. For more details, see section 6.15 of PCI Express Base Specification.

Bit	R/W	Reset	Description
5	RO	0x1	**PCIE_CAP_ARI_FORWARD_SUPPORT**: ARI Forwarding Supported. Applicable only to Switch Downstream Ports and Root Ports; must be 0b for other Function types. This bit must be set to 1b if a Switch Downstream Port or Root Port supports this optional capability. For more details, see section 6.13 of PCI Express Base Specification.
4	RO	0x1	**PCIE_CAP_CPL_TIMEOUT_DISABLE_SUPPORT**: Completion Timeout Disable Supported. A value of 1b indicates support for the Completion Timeout Disable mechanism. The Completion Timeout Disable mechanism is required for Endpoints that issue Requests on their own behalf and PCI Express to PCI/PCI-X Bridges that take ownership of Requests issued on PCI Express. This mechanism is optional for Root Ports. For all other Functions this field is Reserved and the controller hardwires this bit to 0b.
3:0	RO	0xf	**PCIE_CAP_CPL_TIMEOUT_RANGE**: Completion Timeout Ranges Supported. This field indicates device Function support for the optional Completion Timeout programmability mechanism. This mechanism allows system software to modify the Completion Timeout value. This field is applicable only to Root Ports, Endpoints that issue Requests on their own behalf, and PCI Express to PCI/PCI-X Bridges that take ownership of Requests issued on PCI Express. For all other Functions this field is Reserved and must be hardwired to 0000b. Four time value ranges are defined: - Range A: 50 us to 10 ms - Range B: 10 ms to 250 ms - Range C: 250 ms to 4 s - Range D: 4 s to 64 s Bits are set according to the list below to show timeout value ranges supported. All encodings other than the defined encodings are reserved. It is strongly recommended that the Completion Timeout mechanism not expire in less than 10 ms.

PCIE_X<*i*>_RC_PF0_PCIE_CAP_DEVICE_CONTROL2_DEVICE_STATUS2_REG_0

where <*i*> = 1, 4, 8.

Description: This register controls PCI Express device specific parameters and provides information about PCI Express device (function) specific parameters; in addition to the Device Control and Device Status Register.

PCIE_X1_RC_PF0_PCIE_CAP_DEVICE_CONTROL2_DEVICE_STATUS2_REG_0

PCIE_X4_RC_PF0_PCIE_CAP_DEVICE_CONTROL2_DEVICE_STATUS2_REG_0

PCIE_X8_RC_PF0_PCIE_CAP_DEVICE_CONTROL2_DEVICE_STATUS2_REG_0

Offset: 0x98
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,x0xx,xx00,0000)

Bit	Reset	Description
10	0x0	**PCIE_CAP_LTR_EN**: LTR Mechanism Enable. When set to 1b, this bit enables Upstream Ports to send LTR messages and Downstream Ports to process LTR Messages. For a Multi-Function Device associated with an Upstream Port of a device that implements LTR, the bit in Function 0 is RW, and only Function 0 controls the component's Link behavior. In all other Functions of that device, this bit is RsvdP. Functions that do not implement the LTR mechanism are permitted to hardwire this bit to 0b. For Downstream Ports, this bit must be reset to the default value if the Port goes to DL_Down status. The write value is gated with the PCIE_CAP_LTR_SUPP field of DEVICE_CAPABILITIES2_REG. Note: RW for function #0 and RsdvP for all other functions. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (pf=0 && DEVICE_CAPABILITIES2_REG.PCIE_CAP_LTR_SUPP) then R/W else R
5	0x0	**PCIE_CAP_ARI_FORWARD_SUPPORT_CS**: ARI Forwarding Enable. When set, the Downstream Port disables its traditional Device Number field being 0 enforcement when turning a Type 1 Configuration Request into a Type 0 Configuration Request, permitting access to Extended Functions in an ARI Device immediately below the Port. For more details, see Section 6.13 of PCI Express Base Specification. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
4	0x0	**PCIE_CAP_CPL_TIMEOUT_DISABLE**: Completion Timeout Disable. When set, this bit disables the Completion Timeout mechanism. This bit is required for all Functions that support the Completion Timeout Disable Capability. Functions that do not support this optional capability are permitted to hardwire this bit to 0b Software is permitted to set or clear this bit at any time. When set, the Completion Timeout detection mechanism is disabled. If there are outstanding Requests when the bit is cleared, it is permitted but not required for hardware to apply the completion timeout mechanism to the outstanding Requests. If this is done, it is permitted to base the start time for each Request on either the time this bit was cleared or the time each Request was issued.
3:0	0x0	**PCIE_CAP_CPL_TIMEOUT_VALUE**: Completion Timeout Value. In device Functions that support Completion Timeout programmability, this field allows system software to modify the Completion Timeout value. This field is applicable to Root Ports, Endpoints that issue Requests on their own behalf, and PCI Express to PCI/PCI-X Bridges that take ownership of Requests issued on PCI Express. For all other Functions this field is Reserved and controller hardwires it to 0000b. A Function that does not support this optional capability must hardwire this field to 0000b and is required to implement a timeout value in the range 50 us to 50 ms. Functions that support Completion Timeout programmability must support the values given below corresponding to the programmability ranges indicated in the Completion Timeout Ranges Supported field. All encodings other than the defined encodings are reserved. It is strongly recommended that the Completion Timeout mechanism not expire in less than 10 ms. Values available if Range A (50 us to 10 ms) programmability range is supported: Values available if Range B (10 ms to 250 ms) programmability range is supported: Values available if Range C (250 ms to 4 s) programmability range is supported: Values available if the Range D (4 s to 64 s) programmability range is supported: Software is permitted to change the value in this field at any time. For Requests already pending when the Completion Timeout Value is changed, hardware is permitted to use either the new or the old value for the outstanding Requests, and is permitted to base the start time for each Request either on when this value was changed or on when each request was issued. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_X<*i*>_RC_PF0_PCIE_CAP_LINK_CAPABILITIES2_REG_0

where <*i*> = 1, 4, 8.

Description: This register identifies PCI Express Link specific capabilities; in addition to the Link Capabilities Register.

PCIE_X1_RC_PF0_PCIE_CAP_LINK_CAPABILITIES2_REG_0

PCIE_X4_RC_PF0_PCIE_CAP_LINK_CAPABILITIES2_REG_0

PCIE_X8_RC_PF0_PCIE_CAP_LINK_CAPABILITIES2_REG_0

Offset: 0x9c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0180001e (0bx000,0001,1000,0000,0000,0000,0001,1110)

Bit	R/W	Reset	Description
30:25	RO	0x0	**RSVDP_25**: Reserved for future use.
24	RW	0x1	**PCIE_CAP_TWO_RETIMERS_PRE_DET_SUPPORT**: Two Retimers Presence Detect Supported. When set to 1b, this bit indicates that the associated Port supports detection and reporting of two Retimers presence. This bit must be set to 1b in a Port when the Supported Link Speeds Vector of the Link Capabilities 2 register indicates support for a Link speed of 16.0 GT/s or higher. It is permitted to be set to 1b regardless of the supported Link speeds if the Retimer Presence Detect Supported bit is also set to 1b. Multi-Function Devices associated with an Upstream Port must report the same value in this field for all Functions. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky)
23	RW	0x1	**PCIE_CAP_RETIMER_PRE_DET_SUPPORT**: Retimer Presence Detect Supported. When set to 1b, this bit indicates that the associated Port supports detection and reporting of Retimer presence. This bit must be set to 1b in a Port when the Supported Link Speeds Vector of the Link Capabilities 2 register indicates support for a Link speed of 16.0 GT/s or higher. It is permitted to be set to 1b regardless of the supported Link speeds. Multi-Function Devices associated with an Upstream Port must report the same value in this field for all Functions. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky)
22:9	RO	0x0	**RSVDP_9**: Reserved for future use.

Bit	R/W	Reset	Description
8	RO	0x0	**PCIE_CAP_CROSS_LINK_SUPPORT**: Crosslink Supported. When set to 1b, this bit indicates that the associated Port supports crosslinks (for more details, see section 4.2.6.3.1 of PCI Express Base Specification). When set to 0b on a Port that supports Link speeds of 8.0 GT/s or higher, this bit indicates that the associated Port does not support crosslinks. When set to 0b on a Port that only supports Link speeds of 2.5 GT/s or 5.0 GT/s, this bit provides no information regarding the Port's level of crosslink support. It is recommended that this bit be Set in any Port that supports crosslinks even though doing so is only required for Ports that also support operating at 8.0 GT/s or higher Link speeds. Note: Software should use this bit when referencing fields whose definition depends on whether or not the Port supports crosslinks (for more details, see section 7.7.3.4 of PCI Express Base Specification). Multi-Function Devices associated with an Upstream Port must report the same value in this field for all Functions.
7:1	RO	0xf	**PCIE_CAP_SUPPORT_LINK_SPEED_VECTOR**: Supported Link Speeds Vector. This field indicates the supported Link speed(s) of the associated Port. For each bit, a value of 1b indicates that the corresponding Link speed is supported; otherwise, the Link speed is not supported. For more details, see section 8.2.1 of PCI Express Base Specification. Bit definitions within this field are: - Bit 0 2.5 GT/s - Bit 1 5.0 GT/s - Bit 2 8.0 GT/s - Bit 3 16.0 GT/s - Bit 4 32.0 GT/s - Bits 6:5 RsvdP Multi-Function Devices associated with an Upstream Port must report the same value in this field for all Functions. This field has a default of (PCIE_CAP_MAX_LINK_SPEED == 0101) ? 0011111 : (PCIE_CAP_MAX_LINK_SPEED == 0100) ? 0001111 : (PCIE_CAP_MAX_LINK_SPEED == 0011) ? 0000111 : (PCIE_CAP_MAX_LINK_SPEED == 0010) ? 0000011 : 0000001 where PCIE_CAP_MAX_LINK_SPEED is a field in the LINK_CAPABILITIES_REG register.
0	RO	0x0	**RSVDP_0**: Reserved for future use.

PCIE_X<*i*>_RC_PF0_PCIE_CAP_LINK_CONTROL2_LINK_STATUS2_REG_0

where <*i*> = 1, 4, 8.

Description: This register controls and provides information about PCI Express Link specific parameters; in addition to the Link Control and Link Status Register.

PCIE_X1_RC_PF0_PCIE_CAP_LINK_CONTROL2_LINK_STATUS2_REG_0

PCIE_X4_RC_PF0_PCIE_CAP_LINK_CONTROL2_LINK_STATUS2_REG_0

PCIE_X8_RC_PF0_PCIE_CAP_LINK_CONTROL2_LINK_STATUS2_REG_0

Offset: 0xa0
Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x02010004 (0bxxxx,0010,0000,0001,0000,0000,0000,0100)

Bit	R/W	Reset	Description
27:26	RO	0x0	**RSVDP_26**: Reserved for future use.
25:24	RO	0x2	**PCIE_CAP_CROSSLINK_RESOLUTION**: Crosslink Resolution. This field indicates the state of the Crosslink negotiation. It must be implemented if Crosslink Supported is Set and the Port supports 16.0 GT/s or higher data rate. It is permitted to be implemented in all other Ports. If Crosslink Supported is clear, the controller hardwires this field to 01b or 10b. Once a value of 01b or 10b is returned in this field, that value must continue to be returned while the Link is Up.
23	RO	0x0	**PCIE_CAP_TWO_RETIMERS_PRE_DET**: Two Retimers Presence Detected. When set to 1b, this bit indicates that two Retimers were present during the most recent Link negotiation. For more details, see section 4.2.6.3.5.1 of PCI Express Base Specification. This bit is required for Ports that have the Two Retimers Presence Detect Supported bit of the Link Capabilities 2 register set to 1b. Ports that have the Two Retimers Presence Detect Supported bit set to 0b are permitted to hardwire this bit to 0b. For Multi-Function Devices associated with an Upstream Port, this bit must be implemented in Function 0 and RsvdZ in all other Functions. Note: This register field is sticky.
22	RO	0x0	**PCIE_CAP_RETIMER_PRE_DET**: Retimer Presence Detected. When set to 1b, this bit indicates that a Retimer was present during the most recent Link negotiation. For more details, see section 4.2.6.3.5.1 of PCI Express Base Specification. This bit is required for Ports that have the Retimer Presence Detect Supported bit of the Link Capabilities 2 register set to 1b. For Ports that have the Retimer Presence Detect Supported bit set to 0b, the controller hardwires this bit to 0b. For Multi-Function Devices associated with an Upstream Port, this bit must be implemented in Function 0 and is RsvdZ in all other Functions. Note: This register field is sticky.
21	RW	0x0	**PCIE_CAP_LINK_EQ_REQ**: Link Equalization Request 8.0 GT/s. This bit is set by hardware to request the 8.0 GT/s Link equalization process to be performed on the Link. For more details, see sections 4.2.3 and 4.2.6.4.2 of PCI Express Base Specification. For Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions. For components that only support speeds below 8.0 GT/s, the controller hardwires this bit to 0b.
20	RO	0x0	**PCIE_CAP_EQ_CPL_P3**: EEqualization 8.0 GT/s Phase 3 Successful. When set to 1b, this bit indicates that Phase 3 of the 8.0 GT/s Transmitter Equalization procedure has successfully completed. Details of the Transmitter Equalization process and when this bit needs to be set to 1b is provided in section 4.2.6.4.2 of PCI Express Base Specification. For Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions. For components that only support speeds below 8.0 GT/s, the controller hardwires this bit to 0b. Note: This register field is sticky.

Bit	R/W	Reset	Description
19	RO	0x0	**PCIE_CAP_EQ_CPL_P2**: Equalization 8.0 GT/s Phase 2 Successful. When set to 1b, this bit indicates that Phase 2 of the 8.0 GT/s Transmitter Equalization procedure has successfully completed. Details of the Transmitter Equalization process and when this bit needs to be set to 1b is provided in section 4.2.6.4.2 of PCI Express Base Specification. For Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions. For components that only support speeds below 8.0 GT/s, the controller hardwires this bit to 0b. Note: This register field is sticky.
18	RO	0x0	**PCIE_CAP_EQ_CPL_P1**: Equalization 8.0 GT/s Phase 1 Successful. When set to 1b, this bit indicates that Phase 1 of the 8.0 GT/s Transmitter Equalization procedure has successfully completed. Details of the Transmitter Equalization process and when this bit needs to be set to 1b is provided in section 4.2.6.4.2 of PCI Express Base Specification. For Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions. For components that only support speeds below 8.0 GT/s, the controller hardwires this bit to 0b. Note: This register field is sticky.
17	RO	0x0	**PCIE_CAP_EQ_CPL**: Equalization 8.0 GT/s Complete. When set to 1b, this bit indicates that the Transmitter Equalization procedure at the 8.0 GT/s data rate has completed. Details of the Transmitter Equalization process and when this bit needs to be set to 1b is provided in section 4.2.6.4.2 of PCI Express Base Specification. For Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions. For components that only support speeds below 8.0 GT/s, the controller hardwires this bit to 0b. Note: This register field is sticky.
16	RO	0x1	**PCIE_CAP_CURR_DEEMPHASIS**: Current De-emphasis Level. When the Link is operating at 5.0 GT/s speed, this bit reflects the level of de-emphasis. The value in this bit is undefined when the Link is not operating at 5.0 GT/s speed. Components that support only the 2.5 GT/s speed are permitted to hardwire this bit to 0b. For components that support speeds greater than 2.5 GT/s, Multi-Function Devices associated with an Upstream Port must report the same value in this field for all Functions of the Port. In M-PCIe mode this register is always 0x0. In C-PCIe mode, its contents are derived by sampling the PIPE.
15:12	RW	0x0	**PCIE_CAP_COMPLIANCE_PRESET**: Compliance Preset/De-emphasis. - For 8.0 GT/s and higher Data Rate: This field sets the Transmitter Preset in Polling.Compliance state if the entry occurred due to the Enter Compliance bit being 1b. The encodings are defined in section 4.2.3.2 of PCI Express Base Specification . Results are undefined if a reserved preset encoding is used when entering Polling.Compliance in this way. - For 5.0 GT/s Data Rate: This field sets the de-emphasis level in Polling.Compliance state if the entry occurred due to the Enter Compliance bit being 1b. - When the Link is operating at 2.5 GT/s, the setting of this field has no effect. Components that support only 2.5 GT/s speed are permitted to hardwire this field to 0000b. - For a Multi-Function Device associated with an Upstream Port, the field in Function 0 is of type RWS, and only Function 0 controls the component's Link behavior. In all other Functions of that device, this field is of type RsvdP. - This field is intended for debug and compliance testing purposes. System firmware and software is allowed to modify this field only during debug or compliance testing. In all other cases, the system must ensure that this field is set to the default value. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.

Bit	R/W	Reset	Description
11	RW	0x0	**PCIE_CAP_COMPLIANCE_SOS**: Compliance SOS. When set to 1b, the LTSSM is required to send SKP Ordered Sets between sequences when sending the Compliance Pattern or Modified Compliance Pattern. For a Multi-Function Device associated with an Upstream Port, the bit in Function 0 is of type RWS, and only Function 0 controls the component's Link behavior. In all other Functions of that device, this bit is of type RsvdP. This bit is applicable when the Link is operating at 2.5 GT/s or 5.0 GT/s data rates only. For components that support only the 2.5 GT/s speed, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.
10	RW	0x0	**PCIE_CAP_ENTER_MODIFIED_COMPLIANCE**: Enter Modified Compliance. When this bit is set to 1b, the device transmits Modified Compliance Pattern if the LTSSM enters Polling.Compliance substate. Components that support only the 2.5 GT/s speed are permitted to hardwire this bit to 0b. For a Multi-Function Device associated with an Upstream Port, the bit in Function 0 is of type RWS, and only Function 0 controls the component's Link behavior. In all other Functions of that device, this bit is of type RsvdP. This bit is intended for debug, compliance testing purposes only. System firmware and software is allowed to modify this bit only during debug or compliance testing. In all other cases, the system must ensure that this bit is set to the default value. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.
9:7	RW	0x0	**PCIE_CAP_TX_MARGIN**: Transmit Margin, This field controls the value of the non-deemphasized voltage level at the Transmitter pins. This field is reset to 000b on entry to the LTSSM Polling.Configuration substate (see Chapter 4 of PCI Express Base Specification for details of how the Transmitter voltage level is determined in various states). - 001b-111b: As defined in Section 8.3.4 not all encodings are required to be implemented. For a Multi-Function Device associated with an Upstream Port, the field in Function 0 is of type RWS, and only Function 0 controls the component's Link behavior. In all other Functions of that device, this field is of type RsvdP. For components that support only the 2.5 GT/s speed, the controller hardwires this bit to 000b. This field is intended for debug, compliance testing purposes only. System firmware and software is allowed to modify this field only during debug or compliance testing. In all other cases, the system must ensure that this field is set to the default value. Note: This register field is sticky.
6	RW	0x0	**PCIE_CAP_SEL_DEEMPHASIS**: Selectable De-emphasis. When the Link is operating at 5.0 GT/s speed, this bit is used to control the transmit de-emphasis of the link in specific situations. For more details, see section 4.2.6 of PCI Express Base Specification. When the Link is not operating at 5.0 GT/s speed, the setting of this bit has no effect. Components that support only the 2.5 GT/s speed are permitted to hardwire this bit to 0b. This bit is not applicable and Reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

Bit	R/W	Reset	Description
5	RW	0x0	**PCIE_CAP_HW_AUTO_SPEED_DISABLE**: Hardware Autonomous Speed Disable. When set, this bit disables hardware from changing the Link speed for device-specific reasons other than attempting to correct unreliable Link operation by reducing Link speed. Initial transition to the highest supported common link speed is not blocked by this bit. For a Multi-Function Device associated with an Upstream Port, the bit in Function 0 is of type RWS, and only Function 0 controls the component's Link behavior. In all other Functions of that device, this bit is of type RsvdP. Functions that do not implement the associated mechanism are permitted to hardwire this bit to 0b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.
4	RW	0x0	**PCIE_CAP_ENTER_COMPLIANCE**: Enter Compliance. Software is permitted to force a Link to enter Compliance mode (at the speed indicated in the Target Link Speed field and the de-emphasis/preset level indicated by the Compliance Preset/De-emphasis field) by setting this bit to 1b in both components on a Link and then initiating a hot reset on the Link. Default value of this bit following Fundamental Reset is 0b. For a Multi-Function Device associated with an Upstream Port, the bit in Function 0 is of type RWS, and only Function 0 controls the component's Link behavior. In all other Functions of that device, this bit is of type RsvdP. Components that support only the 2.5 GT/s speed are permitted to hardwire this bit to 0b. This bit is intended for debug, compliance testing purposes only. System firmware and software is allowed to modify this bit only during debug or compliance testing. In all other cases, the system must ensure that this bit is set to the default value. Note: This register field is sticky.
3:0	RW	0x4	**PCIE_CAP_TARGET_LINK_SPEED**: Target Link Speed. For Downstream Ports, this field sets an upper limit on Link operational speed by restricting the values advertised by the Upstream component in its training sequences. The encoded value specifies a bit location in the Supported Link Speeds Vector (in the Link Capabilities 2 register) that corresponds to the desired target Link speed. All encodings other than the defined encodings are reserved. If a value is written to this field that does not correspond to a supported speed (as indicated by the Supported Link Speeds Vector), the result is undefined. If either of the Enter Compliance or Enter Modified Compliance bits are implemented, then this field must also be implemented. The default value of this field is the highest Link speed supported by the component (as reported in the Max Link Speed field of the Link Capabilities register) unless the corresponding platform/form factor requires a different default value. For both Upstream and Downstream Ports, this field is used to set the target compliance mode speed when software is using the Enter Compliance bit to force a Link into compliance mode. For Upstream Ports, if the Enter Compliance bit is Clear, this field is permitted to have no effect. For a Multi-Function Device associated with an Upstream Port, the field in Function 0 is of type RWS, and only Function 0 controls the component's Link behavior. In all other Functions of that device, this field is of type RsvdP. Components that support only the 2.5 GT/s speed are permitted to hardwire this field to 0000b. In M-PCIe mode, the contents of this field are derived from other registers. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_MSIX_CAP_PCI_MSIX_CAP_ID_NEXT_CTRL_REG_0

where <*i*> = 1, 4, 8.

Description: This Register holds MSI-X Capability ID, Next Capability pointer. It also controls the MSI-X behaviour.

Offset: 0xb0
Read/Write: See table below
Parity Protection: N
Shadow: NSCR Protection: 0

PCIE_X1_RC_PF0_MSIX_CAP_PCI_MSIX_CAP_ID_NEXT_CTRL_REG_0

Reset: 0x00000011 (0b0000,0000,0000,0000,0000,0000,0001,0001)

Bit	R/W	Reset	Description
31	RW	0x0	**PCI_MSIX_ENABLE**: MSI-X Enable. If Set and the MSI Enable bit in the MSI Message Control Register for MSI is Clear, the Function is permitted to use MSI-X to request service and is prohibited from using INTx interrupts (if implemented). System configuration software Sets this bit to enable MSI-X. A device driver is prohibited from writing this bit to mask a Function's service request. If Clear, the Function is prohibited from using MSI-X to request service.
30	RW	0x0	**PCI_MSIX_FUNCTION_MASK**: Function Mask. If Set, all of the vectors associated with the Function are masked, regardless of their per-vector Mask bit values. If Clear, each vector's Mask bit determines whether the vector is masked or not. Setting or Clearing the MSI-X Function Mask bit has no effect on the value of the per-vector Mask bits. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
29:27	RO	0x0	**RSVDP_27**: Reserved for future use.
26:16	RW	0x0	**PCI_MSIX_TABLE_SIZE**: MSI-X Table Size. System software reads this field to determine the MSI-X Table Size N, which is encoded as N-1. For example, a returned value of 000 0000 0011b indicates a table size of 4. SRIOV Note: All VFs in a single PF have the same value for "MSI-X Table Size" (PCI_MSIX_TABLE_SIZE field in SHADOW_PCI_MSIX_CAP_ID_NEXT_CTRL_REG). To write this common value, you must perform a DBI_CS2 write (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) while accessing the PCI_MSIX_TABLE_SIZE field in the PF PCI_MSIX_CAP_ID_NEXT_CTRL_REG register. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
15:8	RW	0x0	**PCI_MSIX_CAP_NEXT_OFFSET**: MSI-X Next Capability Pointer. This field contains the offset to the next PCI Capability structure or 00h if no other items exist in the linked list of Capabilities. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
7:0	RO	0x11	**PCI_MSIX_CAP_ID**: MSI-X Capability ID. This field indicates the MSI-X Capability structure. This field must return a Capability ID of 11h indicating that this is an MSI-X Capability structure.

PCIE_X4_RC_PF0_MSIX_CAP_PCI_MSIX_CAP_ID_NEXT_CTRL_REG_0

PCIE_X8_RC_PF0_MSIX_CAP_PCI_MSIX_CAP_ID_NEXT_CTRL_REG_0

Reset: 0x00070011 (0b0000,0000,0000,0111,0000,0000,0001,0001)

Bit	R/W	Reset	Description
31	RW	0x0	**PCI_MSIX_ENABLE**: MSI-X Enable. If Set and the MSI Enable bit in the MSI Message Control Register for MSI is Clear, the Function is permitted to use MSI-X to request service and is prohibited from using INTx interrupts (if implemented). System configuration software Sets this bit to enable MSI-X. A device driver is prohibited from writing this bit to mask a Function's service request. If Clear, the Function is prohibited from using MSI-X to request service.
30	RW	0x0	**PCI_MSIX_FUNCTION_MASK**: Function Mask. If Set, all of the vectors associated with the Function are masked, regardless of their per-vector Mask bit values. If Clear, each vector's Mask bit determines whether the vector is masked or not. Setting or Clearing the MSI-X Function Mask bit has no effect on the value of the per-vector Mask bits. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
29:27	RO	0x0	**RSVDP_27**: Reserved for future use.
26:16	RW	0x7	**PCI_MSIX_TABLE_SIZE**: MSI-X Table Size. System software reads this field to determine the MSI-X Table Size N, which is encoded as N-1. For example, a returned value of 000 0000 0011b indicates a table size of 4. SRIOV Note: All VFs in a single PF have the same value for "MSI-X Table Size" (PCI_MSIX_TABLE_SIZE field in SHADOW_PCI_MSIX_CAP_ID_NEXT_CTRL_REG). To write this common value, you must perform a DBI_CS2 write (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) while accessing the PCI_MSIX_TABLE_SIZE field in the PF PCI_MSIX_CAP_ID_NEXT_CTRL_REG register. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
15:8	RW	0x0	**PCI_MSIX_CAP_NEXT_OFFSET**: MSI-X Next Capability Pointer. This field contains the offset to the next PCI Capability structure or 00h if no other items exist in the linked list of Capabilities. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
7:0	RO	0x11	**PCI_MSIX_CAP_ID**: MSI-X Capability ID. This field indicates the MSI-X Capability structure. This field must return a Capability ID of 11h indicating that this is an MSI-X Capability structure.

PCIE_X<*i*>_RC_PF0_MSIX_CAP_MSIX_TABLE_OFFSET_REG_0

where <*i*> = 1, 4, 8.

Description: This register provides Table BIR and MSI-x Table offset select.

Offset: 0xb4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_MSIX_CAP_MSIX_TABLE_OFFSET_REG_0

Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:3	0x0	**PCI_MSIX_TABLE_OFFSET**: MSI-X Table Offset. Used as an offset from the address contained by one of the Function's Base Address Registers to point to the base of the MSI-X Table. The lower 3 Table BIR bits are masked off (set to zero) by software to form a 32-bit QWORD-aligned offset. SRIOV Note: All VFs in a single PF have the same value for "MSI-X Table Offset" (PCI_MSIX_TABLE_OFFSET field in SHADOW_MSIX_TABLE_OFFSET_REG). To write this common value, you must perform a DBI_CS2 write (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) while accessing the PCI_MSIX_TABLE_OFFSET field in the PF MSIX_TABLE_OFFSET_REG register. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
2:0	0x0	**PCI_MSIX_BIR**: MSI-X Table BAR Indicator Register Field. This field indicates which one of a Function's Base Address Registers , located beginning at 10h in Configuration Space, or entry in the Enhanced Allocation capability with a matching BEI , is used to map the Function's MSI-X Table into Memory Space. All encodings other than the defined encodings are reserved. For a 64-bit Base Address Register , the Table BIR indicates the lower DWORD. For Functions with Type 1 Configuration Space headers, BIR values 2 through 5 are also Reserved. SRIOV Note: All VFs in a single PF have the same value for "MSI-X Table BAR Indicator Register" (PCI_MSIX_BIR field in SHADOW_MSIX_TABLE_OFFSET_REG). To write this common value, you must perform a DBI_CS2 write (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) while accessing the PCI_MSIX_BIR field in the PF MSIX_TABLE_OFFSET_REG register. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

PCIE_X4_RC_PF0_MSIX_CAP_MSIX_TABLE_OFFSET_REG_0

PCIE_X8_RC_PF0_MSIX_CAP_MSIX_TABLE_OFFSET_REG_0

Reset: 0x00000002 (0b0000,0000,0000,0000,0000,0000,0000,0010)

Bit	Reset	Description
31:3	0x0	**PCI_MSIX_TABLE_OFFSET**: MSI-X Table Offset. Used as an offset from the address contained by one of the Function's Base Address Registers to point to the base of the MSI-X Table. The lower 3 Table BIR bits are masked off (set to zero) by software to form a 32-bit QWORD-aligned offset. SRIOV Note: All VFs in a single PF have the same value for "MSI-X Table Offset" (PCI_MSIX_TABLE_OFFSET field in SHADOW_MSIX_TABLE_OFFSET_REG). To write this common value, you must perform a DBI_CS2 write (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) while accessing the PCI_MSIX_TABLE_OFFSET field in the PF MSIX_TABLE_OFFSET_REG register. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
2:0	0x2	**PCI_MSIX_BIR**: MSI-X Table BAR Indicator Register Field. This field indicates which one of a Function's Base Address Registers , located beginning at 10h in Configuration Space, or entry in the Enhanced Allocation capability with a matching BEI , is used to map the Function's MSI-X Table into Memory Space. All encodings other than the defined encodings are reserved. For a 64-bit Base Address Register , the Table BIR indicates the lower DWORD. For Functions with Type 1 Configuration Space headers, BIR values 2 through 5 are also Reserved. SRIOV Note: All VFs in a single PF have the same value for "MSI-X Table BAR Indicator Register" (PCI_MSIX_BIR field in SHADOW_MSIX_TABLE_OFFSET_REG). To write this common value, you must perform a DBI_CS2 write (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) while accessing the PCI_MSIX_BIR field in the PF MSIX_TABLE_OFFSET_REG register. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_MSIX_CAP_MSIX_PBA_OFFSET_REG_0

where <*i*> = 1, 4, 8.

Description: This register provides PBA Offset and PBA BIR value.

Offset: 0xb8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_MSIX_CAP_MSIX_PBA_OFFSET_REG_0

Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:3	0x0	**PCI_MSIX_PBA_OFFSET**: MSI-X PBA Offset. Used as an offset from the address contained by one of the Function's Base Address Registers to point to the base of the MSI-X PBA. The lower 3 PBA BIR bits are masked off (set to zero) by software to form a 32-bit QWORD-aligned offset. SRIOV Note: All VFs in a single PF have the same value for "MSI-X PBA Offset" (PCI_MSIX_PBA_OFFSET field in SHADOW_MSIX_PBA_OFFSET_REG). To write this common value, you must perform a DBI_CS2 write (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) while accessing the PCI_MSIX_PBA_OFFSET field in the PF MSIX_PBA_OFFSET_REG register. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
2:0	0x0	**PCI_MSIX_PBA_BIR**: MSI-X PBA BIR. This field indicates which one of a Function's Base Address Registers , located beginning at 10h in Configuration Space, or entry in the Enhanced Allocation capability with a matching BEI, is used to map the Function's MSI-X PBA into Memory Space. The PBA BIR value definitions are identical to those for the Table BIR . SRIOV Note: All VFs in a single PF have the same value for "MSI-X PBA BIR" (PCI_MSIX_PBA_BIR field in SHADOW_MSIX_PBA_OFFSET_REG). To write this common value, you must perform a DBI_CS2 write (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) while accessing the PCI_MSIX_PBA_BIR field in the PF MSIX_PBA_OFFSET_REG register. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

PCIE_X4_RC_PF0_MSIX_CAP_MSIX_PBA_OFFSET_REG_0

PCIE_X8_RC_PF0_MSIX_CAP_MSIX_PBA_OFFSET_REG_0

Reset: 0x00010002 (0b0000,0000,0000,0001,0000,0000,0000,0010)

Bit	Reset	Description
31:3	0x2000	**PCI_MSIX_PBA_OFFSET**: MSI-X PBA Offset. Used as an offset from the address contained by one of the Function's Base Address Registers to point to the base of the MSI-X PBA. The lower 3 PBA BIR bits are masked off (set to zero) by software to form a 32-bit QWORD-aligned offset. SRIOV Note: All VFs in a single PF have the same value for "MSI-X PBA Offset" (PCI_MSIX_PBA_OFFSET field in SHADOW_MSIX_PBA_OFFSET_REG). To write this common value, you must perform a DBI_CS2 write (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) while accessing the PCI_MSIX_PBA_OFFSET field in the PF MSIX_PBA_OFFSET_REG register. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
2:0	0x2	**PCI_MSIX_PBA_BIR**: MSI-X PBA BIR. This field indicates which one of a Function's Base Address Registers , located beginning at 10h in Configuration Space, or entry in the Enhanced Allocation capability with a matching BEI, is used to map the Function's MSI-X PBA into Memory Space. The PBA BIR value definitions are identical to those for the Table BIR . SRIOV Note: All VFs in a single PF have the same value for "MSI-X PBA BIR" (PCI_MSIX_PBA_BIR field in SHADOW_MSIX_PBA_OFFSET_REG). To write this common value, you must perform a DBI_CS2 write (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) while accessing the PCI_MSIX_PBA_BIR field in the PF MSIX_PBA_OFFSET_REG register. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_AER_CAP_AER_EXT_CAP_HDR_OFF_0

where <*i*> = 1, 4, 8.

Description: Advanced Error Reporting Extended Capability Header provides information about Capability ID, Version, and next offset.

PCIE_X1_RC_PF0_AER_CAP_AER_EXT_CAP_HDR_OFF_0

PCIE_X4_RC_PF0_AER_CAP_AER_EXT_CAP_HDR_OFF_0

PCIE_X8_RC_PF0_AER_CAP_AER_EXT_CAP_HDR_OFF_0

Offset: 0x100
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x14820001 (0b0001,0100,1000,0010,0000,0000,0000,0001)

Bit	Reset	Description
31:20	0x148	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x2	**CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. This field must be 2h if the End-End TLP Prefix Supported bit is set and must be 1h or 2h otherwise. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x1	**CAP_ID**: AER Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. The Extended Capability ID for the Advanced Error Reporting Capability is 0001h. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_AER_CAP_UNCORR_ERR_STATUS_OFF_0

where <*i*> = 1, 4, 8.

Description: The Uncorrectable Error Status Register (Offset 04h) indicates error detection status of individual errors on a PCI Express device Function. An individual error status bit that is Set indicates that a particular error was detected; software may clear an error status by writing a 1b to the respective bit. Register bits not implemented by the Function are hardwired to 0b.

PCIE_X1_RC_PF0_AER_CAP_UNCORR_ERR_STATUS_OFF_0

PCIE_X4_RC_PF0_AER_CAP_UNCORR_ERR_STATUS_OFF_0

PCIE_X8_RC_PF0_AER_CAP_UNCORR_ERR_STATUS_OFF_0

Offset: 0x104
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0xxx,00x0,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:27	RO	0x0	**RSVDP_27**: Reserved for future use.
23	RO	0x0	**RSVDP_23**: Reserved for future use.
22	RW	0x0	**INTERNAL_ERR_STATUS**: Uncorrectable Internal Error Status. This field gives status of the Uncorrectable Internal Error. The controller sets this bit when your application asserts app_err_bus[9]. It does not set this bit when it detects internal uncorrectable internal errors such as parity and ECC failures. You should use the outputs from these errors to drive the app_err_bus[9] input. For more details, see the "Data Integrity (Wire, Datapath, and RAM Protection)" section in the Databook.
20	RW	0x0	**UNSUPPORTED_REQ_ERR_STATUS**: Unsupported Request Error Status. This field represents status of Unsupported Request Error.
19	RW	0x0	**ECRC_ERR_STATUS**: ECRC Error Status. This field represents status of ECRC Error. Note:If CX_ECRC_ENABLE=0 the register field always reads 0.
18	RW	0x0	**MALF_TLP_ERR_STATUS**: Malformed TLP Status. This field represents status of Malformed TLP.
17	RW	0x0	**REC_OVERFLOW_ERR_STATUS**: Receiver Overflow Status. Status bit for Receiver Overflow.
16	RW	0x0	**UNEXP_CMPLT_ERR_STATUS**: Unexpected Completion Status. Status bit for Unexpected Completion.
15	RW	0x0	**CMPLT_ABORT_ERR_STATUS**: Completer Abort Status. Status bit for Completer Abort.
14	RW	0x0	**CMPLT_TIMEOUT_ERR_STATUS**: Completion Timeout Status. Status for Completion Timeout.

Bit	R/W	Reset	Description
13	RW	0x0	**FC_PROTOCOL_ERR_STATUS**: Flow Control Protocol Error Status. Status bit for Flow Control Protocol Error.
12	RW	0x0	**POIS_TLP_ERR_STATUS**: Poisoned TLP Status. Status bit for Poisoned TLP.
11:6	RO	0x0	**RSVDP_6**: Reserved for future use.
5	RW	0x0	**SURPRISE_DOWN_ERR_STATUS**: Surprise Down Error Status (Optional). Status bit for Surprise Down Error.
4	RW	0x0	**DL_PROTOCOL_ERR_STATUS**: Data Link Protocol Error Status. Status bit for Data Link Protocol Error.
3:0	RO	0x0	**RSVDP_0**: Reserved for future use.

PCIE_X<*i*>_RC_PF0_AER_CAP_UNCORR_ERR_MASK_OFF_0

where <*i*> = 1, 4, 8.

Description: The Uncorrectable Error Mask Register controls reporting of individual errors by the device Function to the PCI Express Root Complex through a PCI Express error Message. A masked error (respective bit Set in the mask register) is not recorded or reported in the Header Log, TLP Prefix Log, or First Error Pointer, and is not reported to the PCI Express Root Complex by this Function. There is a mask bit per error bit of the Uncorrectable Error Status register. Register fields for bits not implemented by the Function are hardwired to 0b.

PCIE_X1_RC_PF0_AER_CAP_UNCORR_ERR_MASK_OFF_0

PCIE_X4_RC_PF0_AER_CAP_UNCORR_ERR_MASK_OFF_0

PCIE_X8_RC_PF0_AER_CAP_UNCORR_ERR_MASK_OFF_0

Offset: 0x108
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00400000 (0b0000,0xxx,01x0,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:27	RO	0x0	**RSVDP_27**: Reserved for future use.

Bit	R/W	Reset	Description
23	RO	0x0	**RSVDP_23**: Reserved for future use.
22	RW	0x1	**INTERNAL_ERR_MASK**: Uncorrectable Internal Error Mask (Optional). Mask bit for Uncorrectable Internal Error. Note: This register field is sticky.
20	RW	0x0	**UNSUPPORTED_REQ_ERR_MASK**: Unsupported Request Error Mask. Mask bit for Unsupported Request Error. Note: This register field is sticky.
19	RW	0x0	**ECRC_ERR_MASK**: ECRC Error Mask (Optional). Mask bit for ECRC Error. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.
18	RW	0x0	**MALF_TLP_ERR_MASK**: Malformed TLP Mask. Mask bit for Malformed TLP. Note: This register field is sticky.
17	RW	0x0	**REC_OVERFLOW_ERR_MASK**: Receiver Overflow Mask (Optional). This field represents Receiver Overflow Mask. Note: This register field is sticky.
16	RW	0x0	**UNEXP_CMPLT_ERR_MASK**: Unexpected Completion Mask. Mask bit for Unexpected Completion Error. Note: This register field is sticky.
15	RW	0x0	**CMPLT_ABORT_ERR_MASK**: Completer Abort Error Mask (Optional). Mask bit for Completer Abort Error. Note: This register field is sticky.
14	RW	0x0	**CMPLT_TIMEOUT_ERR_MASK**: Completion Timeout Error Mask. Mask bit for Completion Timeout Error. Note: This register field is sticky.
13	RW	0x0	**FC_PROTOCOL_ERR_MASK**: Flow Control Protocol Error Mask. Mask bit for Flow Control Protocol Error. Note: This register field is sticky.
12	RW	0x0	**POIS_TLP_ERR_MASK**: Poisoned TLP Error Mask. Mask bit for Poisoned TLP Error. Note: This register field is sticky.
11:6	RO	0x0	**RSVDP_6**: Reserved for future use.
5	RW	0x0	**SURPRISE_DOWN_ERR_MASK**: Surprise Down Error Mask. Mask bit for Surprise Down Error. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: LINK_CAPABILITIES_REG.PCIE_CAP_SURPRISE_DOWN_ERR_REP_CAP ? RW : RO Note: This register field is sticky.
4	RW	0x0	**DL_PROTOCOL_ERR_MASK**: Data Link Protocol Error Mask. This field informs whether Data Link Protocol Error is masked or not. Note: This register field is sticky.
3:0	RO	0x0	**RSVDP_0**: Reserved for future use.

PCIE_X<*i*>_RC_PF0_AER_CAP_UNCORR_ERR_SEV_OFF_0

where <*i*> = 1, 4, 8.

Description: The Uncorrectable Error Severity Register controls whether an individual error is reported as a Non-fatal or Fatal error. An error is reported as fatal when the corresponding error bit in the severity register is Set. If the bit is Clear, the corresponding error is considered non-fatal.

PCIE_X1_RC_PF0_AER_CAP_UNCORR_ERR_SEV_OFF_0

PCIE_X4_RC_PF0_AER_CAP_UNCORR_ERR_SEV_OFF_0

PCIE_X8_RC_PF0_AER_CAP_UNCORR_ERR_SEV_OFF_0

Offset: 0x10c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00462030 (0b0000,0xx0,01x0,0110,0010,0000,0011,0000)

Bit	R/W	Reset	Description
31:27	RO	0x0	**RSVDP_27**: Reserved for future use.
24	RO	0x0	**ATOMIC_EGRESS_BLOCKED_ERR_SEVERITY**: AtomicOp Egress Blocked Severity (Optional). Severity bit for AtomicOp Egress Blocked Error. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.
23	RO	0x0	**RSVDP_23**: Reserved for future use.
22	RW	0x1	**INTERNAL_ERR_SEVERITY**: Uncorrectable Internal Error Severity (Optional). Severity bit for Uncorrectable Internal Error. Note: This register field is sticky.
20	RW	0x0	**UNSUPPORTED_REQ_ERR_SEVERITY**: Unsupported Request Error Severity. Severity bit for Unsupported Request Error. Note: This register field is sticky.
19	RW	0x0	**ECRC_ERR_SEVERITY**: ECRC Error Severity (Optional). Severity bit for ECRC Error. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.
18	RW	0x1	**MALF_TLP_ERR_SEVERITY**: Malformed TLP Severity. Severity bit for Malformed TLP. Note: This register field is sticky.

Bit	R/W	Reset	Description
17	RW	0x1	**REC_OVERFLOW_ERR_SEVERITY**: Receiver Overflow Error Severity (Optional). Severity bit for Receiver Overflow Error. Note: This register field is sticky.
16	RW	0x0	**UNEXP_CMPLT_ERR_SEVERITY**: Unexpected Completion Error Severity. Severity bit for Unexpected Completion Error. Note: This register field is sticky.
15	RW	0x0	**CMPLT_ABORT_ERR_SEVERITY**: Completer Abort Error Severity (Optional). Severity bit for Completer Abort Error. Note: This register field is sticky.
14	RW	0x0	**CMPLT_TIMEOUT_ERR_SEVERITY**: Completion Timeout Error Severity. Severity bit for Completion Timeout Error. Note: This register field is sticky.
13	RW	0x1	**FC_PROTOCOL_ERR_SEVERITY**: Flow Control Protocol Error Severity (Optional). Severity bit for Flow Control Protocol Error. Note: This register field is sticky.
12	RW	0x0	**POIS_TLP_ERR_SEVERITY**: Poisoned TLP Severity. Severity bit for Poisoned TLP. Note: This register field is sticky.
11:6	RO	0x0	**RSVDP_6**: Reserved for future use.
5	RW	0x1	**SURPRISE_DOWN_ERR_SVRITY**: Surprise Down Error Severity (Optional). Severity bit for Surprise Down Error. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: LINK_CAPABILITIES_REG.PCIE_CAP_SURPRISE_DOWN_ERR_REP_CAP ? RW : RO Note: This register field is sticky.
4	RW	0x1	**DL_PROTOCOL_ERR_SEVERITY**: Data Link Protocol Error Severity. Severity bit for Data Link Protocol Error. Note: This register field is sticky.
3:0	RO	0x0	**RSVDP_0**: Reserved for future use.

PCIE_X<*i*>_RC_PF0_AER_CAP_CORR_ERR_STATUS_OFF_0

where <*i*> = 1, 4, 8.

Description: The Correctable Error Status register reports error status of individual correctable error sources on a PCI Express device Function. When an individual error status bit is Set, it indicates that a particular error occurred; software may clear an error status by writing a 1b to the respective bit. Register bits not implemented by the Function are hardwired to 0b by the controller.

PCIE_X1_RC_PF0_AER_CAP_CORR_ERR_STATUS_OFF_0

PCIE_X4_RC_PF0_AER_CAP_CORR_ERR_STATUS_OFF_0

PCIE_X8_RC_PF0_AER_CAP_CORR_ERR_STATUS_OFF_0

Offset: 0x110
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:16	RO	0x0	**RSVDP_16**: Reserved for future use.
15	RW	0x0	**HEADER_LOG_OVERFLOW_STATUS**: Header Log Overflow Error Status (Optional). This field provides status of Header Log Overflow Error.
14	RW	0x0	**CORRECTED_INT_ERR_STATUS**: Corrected Internal Error Status (Optional). This field provides status of Corrected Internal Error.
13	RW	0x0	**ADVISORY_NON_FATAL_ERR_STATUS**: Advisory Non-Fatal Error Status. Status bit for Advisory Non-Fatal Error.
12	RW	0x0	**RPL_TIMER_TIMEOUT_STATUS**: Replay Timer Timeout Status. Status bit for Replay Timer Timeout.
11:9	RO	0x0	**RSVDP_9**: Reserved for future use.
8	RW	0x0	**REPLAY_NO_ROLEOVER_STATUS**: REPLAY_NUM Rollover Status. Status bit for REPLAY_NUM Rollover.
7	RW	0x0	**BAD_DLLP_STATUS**: Bad DLLP Status. Status bit for Bad DLLP.
6	RW	0x0	**BAD_TLP_STATUS**: Bad TLP Status. Status bit for Bad TLP.
5:1	RO	0x0	**RSVDP_1**: Reserved for future use.
0	RW	0x0	**RX_ERR_STATUS**: Receiver Error Status (Optional). This field provides status of Receiver Error.

PCIE_X<*i*>_RC_PF0_AER_CAP_CORR_ERR_MASK_OFF_0

where <*i*> = 1, 4, 8.

Description: The Correctable Error Mask Register controls reporting of individual correctable errors by this Function to the PCI Express Root Complex through a PCI Express error Message. A masked error (respective bit Set in the mask register) is not reported to the PCI Express Root Complex by this Function. There is a mask bit per error bit in the Correctable Error Status register. Register fields for bits not implemented by the Function are hardwired to 0b by the controller.

PCIE_X1_RC_PF0_AER_CAP_CORR_ERR_MASK_OFF_0

PCIE_X4_RC_PF0_AER_CAP_CORR_ERR_MASK_OFF_0

PCIE_X8_RC_PF0_AER_CAP_CORR_ERR_MASK_OFF_0

Offset: 0x114
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000e000 (0b0000,0000,0000,0000,1110,0000,0000,0000)

Bit	R/W	Reset	Description
31:16	RO	0x0	**RSVDP_16**: Reserved for future use.
15	RW	0x1	**HEADER_LOG_OVERFLOW_MASK**: Header Log Overflow Error Mask (Optional). Masking bit for Header Log Overflow Error. Note: This register field is sticky.
14	RW	0x1	**CORRECTED_INT_ERR_MASK**: Corrected Internal Error Mask (Optional). Masking bit for Corrected Internal Error Mask. Note: This register field is sticky.
13	RW	0x1	**ADVISORY_NON_FATAL_ERR_MASK**: Advisory Non-Fatal Error Mask. Masking bit for Advisory Non-Fatal Error. Note: This register field is sticky.
12	RW	0x0	**RPL_TIMER_TIMEOUT_MASK**: Replay Timer Timeout Mask. Masking bit for Replay Timer Timeout. Note: This register field is sticky.
11:9	RO	0x0	**RSVDP_9**: Reserved for future use.
8	RW	0x0	**REPLAY_NO_ROLEOVER_MASK**: REPLAY_NUM Rollover Mask. Masking bit for REPLAY_NUM Rollover. Note: This register field is sticky.
7	RW	0x0	**BAD_DLLP_MASK**: Bad DLLP Mask. Masking bit for Bad DLLP. Note: This register field is sticky.
6	RW	0x0	**BAD_TLP_MASK**: Bad TLP Mask. Masking bit for Bad TLP. Note: This register field is sticky.

Bit	R/W	Reset	Description
5:1	RO	0x0	**RSVDP_1**: Reserved for future use.
0	RW	0x0	**RX_ERR_MASK**: Receiver Error Mask (Optional). Masking bit for Receiver Error. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_AER_CAP_ADV_ERR_CAP_CTRL_OFF_0

where <*i*> = 1, 4, 8.

Description: Advanced Error Capabilities and Control Register provides information whether the individual capability is supported or not. If the capability is supported then it is enabled or not.

PCIE_X1_RC_PF0_AER_CAP_ADV_ERR_CAP_CTRL_OFF_0

PCIE_X4_RC_PF0_AER_CAP_ADV_ERR_CAP_CTRL_OFF_0

PCIE_X8_RC_PF0_AER_CAP_ADV_ERR_CAP_CTRL_OFF_0

Offset: 0x118
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000002a0 (0b0000,0000,0000,0000,0000,x010,1010,0000)

Bit	R/W	Reset	Description
31:13	RO	0x0	**RSVDP_13**: Reserved for future use.
12	RO	0x0	**CTO_PRFX_HDR_LOG_CAP**: TLP Prefix Log Present. If Set, this bit indicates that the Function records the prefix/header of Request TLPs that experience a Completion Timeout error.
10	RW	0x0	**MULTIPLE_HEADER_EN**: Multiple Header Recording Enable. When Set, this bit enables the Function to record more than one error header. Functions that do not implement the associated mechanism are permitted to hardwire this bit to 0b. Default value of this bit is 0b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R (sticky) Note: This register field is sticky.
9	RO	0x1	**MULTIPLE_HEADER_CAP**: Multiple Header Recording Capable. If Set, this bit indicates that the Function is capable of recording more than one error header. Note: This register field is sticky.

Bit	R/W	Reset	Description
8	RW	0x0	**ECRC_CHECK_EN**: ECRC Check Enable. When Set, ECRC checking is enabled. Functions that do not implement the associated mechanism are permitted to hardwire this bit to 0b. Default value of this bit is 0b. Note: This register field is sticky.
7	RO	0x1	**ECRC_CHECK_CAP**: ECRC Check Capable. If Set, this bit indicates that the Function is capable of checking ECRC. Note: This register field is sticky.
6	RW	0x0	**ECRC_GEN_EN**: ECRC Generation Enable. When Set, ECRC generation is enabled. Functions that do not implement the associated mechanism are permitted to hardwire this bit to 0b. Default value of this bit is 0b. Note: This register field is sticky.
5	RO	0x1	**ECRC_GEN_CAP**: ECRC Generation Capable. If Set, this bit indicates that the Function is capable of generating ECRC. Note: This register field is sticky.
4:0	RO	0x0	**FIRST_ERR_POINTER**: First Error Pointer. The First Error Pointer is a field that identifies the bit position of the first error reported in the Uncorrectable Error Status register. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_AER_CAP_HDR_LOG_0_OFF_0

where <*i*> = 1, 4, 8.

Description: The Header Log Register 0 contains the header for the TLP corresponding to a detected error; The header is captured such that, when read using DW accesses, the fields of the header are laid out in the same way the headers are presented in the specification. Therefore, byte 0 of the header is located in byte 3 of the Header Log Register 0, byte 1 of the header is in byte 2 of the Header Log Register 0, and so forth. For 12-byte headers, only bytes 0 through 11 of the Header Log Register are used and values in bytes 12 through 15 are undefined. In certain cases where a Malformed TLP is reported, the Header Log Register may contain TLP Prefix information.

PCIE_X1_RC_PF0_AER_CAP_HDR_LOG_0_OFF_0

PCIE_X4_RC_PF0_AER_CAP_HDR_LOG_0_OFF_0

PCIE_X8_RC_PF0_AER_CAP_HDR_LOG_0_OFF_0

Offset: 0x11c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**FIRST_DWORD_FOURTH_BYTE**: Byte 3 of Header log register of First 32-bit Data Word. This field represents fourth byte of First DW of Header. Note: This register field is sticky.
23:16	0x0	**FIRST_DWORD_THIRD_BYTE**: Byte 2 of Header log register of First 32-bit Data Word. This field represents third byte of First DW of Header. Note: This register field is sticky.
15:8	0x0	**FIRST_DWORD_SECOND_BYTE**: Byte 1 of Header log register of First 32-bit Data Word. This field represents second byte of First DW of Header. Note: This register field is sticky.
7:0	0x0	**FIRST_DWORD_FIRST_BYTE**: Byte 0 of Header log register of First 32-bit Data Word. This field represents first byte of First DW of Header. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_AER_CAP_HDR_LOG_1_OFF_0

where <*i*> = 1, 4, 8.

Description: The Header Log Register 1 contains the header for the TLP corresponding to a detected error; The header is captured such that, when read using DW accesses, the fields of the header are laid out in the same way the headers are presented in the specification. Therefore, byte 0 of the header is located in byte 3 of the Header Log Register 1, byte 1 of the header is in byte 2 of the Header Log Register 1 and so forth. For 12-byte headers, only bytes 0 through 11 of the Header Log Register are used and values in bytes 12 through 15 are undefined. In certain cases where a Malformed TLP is reported, the Header Log Register may contain TLP Prefix information.

PCIE_X1_RC_PF0_AER_CAP_HDR_LOG_1_OFF_0

PCIE_X4_RC_PF0_AER_CAP_HDR_LOG_1_OFF_0

PCIE_X8_RC_PF0_AER_CAP_HDR_LOG_1_OFF_0

Offset: 0x120
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**SECOND_DWORD_FOURTH_BYTE**: Byte 3 of Header log register of Second 32-bit Data Word. This field represents fourth byte of Second DW of Header. Note: This register field is sticky.
23:16	0x0	**SECOND_DWORD_THIRD_BYTE**: Byte 2 of Header log register of Second 32-bit Data Word. This field represents third byte of Second DW of Header. Note: This register field is sticky.
15:8	0x0	**SECOND_DWORD_SECOND_BYTE**: Byte 1 of Header log register of Second 32-bit Data Word. This field represents second byte of Second DW of Header. Note: This register field is sticky.
7:0	0x0	**SECOND_DWORD_FIRST_BYTE**: Byte 0 of Header log register of Second 32-bit Data Word. This field represents first byte of Second DW of Header. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_AER_CAP_HDR_LOG_2_OFF_0

where <*i*> = 1, 4, 8.

Description: The Header Log Register 2 contains the header for the TLP corresponding to a detected error; The header is captured such that, when read using DW accesses, the fields of the header are laid out in the same way the headers are presented in the specification. Therefore, byte 0 of the header is located in byte 3 of the Header Log Register 2, byte 1 of the header is in byte 2 of the Header Log Register 2 and so forth. For 12-byte headers, only bytes 0 through 11 of the Header Log Register are used and values in bytes 12 through 15 are undefined. In certain cases where a Malformed TLP is reported, the Header Log Register may contain TLP Prefix information.

PCIE_X1_RC_PF0_AER_CAP_HDR_LOG_2_OFF_0

PCIE_X4_RC_PF0_AER_CAP_HDR_LOG_2_OFF_0

PCIE_X8_RC_PF0_AER_CAP_HDR_LOG_2_OFF_0

Offset: 0x124
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**THIRD_DWORD_FOURTH_BYTE**: Byte 3 of Header log register of Third 32-bit Data Word. This field represents fourth byte of Third DW of Header. Note: This register field is sticky.
23:16	0x0	**THIRD_DWORD_THIRD_BYTE**: Byte 2 of Header log register of Third 32-bit Data Word. This field represents third byte of Third DW of Header. Note: This register field is sticky.
15:8	0x0	**THIRD_DWORD_SECOND_BYTE**: Byte 1 of Header log register of Third 32-bit Data Word. This field represents second byte of Third DW of Header. Note: This register field is sticky.
7:0	0x0	**THIRD_DWORD_FIRST_BYTE**: Byte 0 of Header log register of Third 32-bit Data Word. This field represents first byte of Third DW of Header. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_AER_CAP_HDR_LOG_3_OFF_0

where <*i*> = 1, 4, 8.

Description: The Header Log Register 3 contains the header for the TLP corresponding to a detected error; The header is captured such that, when read using DW accesses, the fields of the header are laid out in the same way the headers are presented in the specification. Therefore, byte 0 of the header is located in byte 3 of the Header Log Register 3, byte 1 of the header is in byte 2 of the Header Log Register 3 and so forth. For 12-byte headers, only bytes 0 through 11 of the Header Log Register are used and values in bytes 12 through 15 are undefined. In certain cases where a Malformed TLP is reported, the Header Log Register may contain TLP Prefix information.

PCIE_X1_RC_PF0_AER_CAP_HDR_LOG_3_OFF_0

PCIE_X4_RC_PF0_AER_CAP_HDR_LOG_3_OFF_0

PCIE_X8_RC_PF0_AER_CAP_HDR_LOG_3_OFF_0

Offset: 0x128
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**FOURTH_DWORD_FOURTH_BYTE**: Byte 3 of Header log register of Fourth 32-bit Data Word. This field represents fourth byte of Fourth DW of Header. Note: This register field is sticky.
23:16	0x0	**FOURTH_DWORD_THIRD_BYTE**: Byte 2 of Header log register of Fourth 32-bit Data Word. This field represents third byte of Fourth DW of Header. Note: This register field is sticky.
15:8	0x0	**FOURTH_DWORD_SECOND_BYTE**: Byte 1 of Header log register of Fourth 32-bit Data Word. This field represents second byte of Fourth DW of Header. Note: This register field is sticky.
7:0	0x0	**FOURTH_DWORD_FIRST_BYTE**: Byte 0 of Header log register of Fourth 32-bit Data Word. This field represents first byte of Fourth DW of Header. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_AER_CAP_ROOT_ERR_CMD_OFF_0

where <*i*> = 1, 4, 8.

Description: The Root Error Command Register allows further control of Root Complex response to Correctable, Non-Fatal, and Fatal error Messages than the basic Root Complex capability to generate system errors in response to error Messages.

PCIE_X1_RC_PF0_AER_CAP_ROOT_ERR_CMD_OFF_0

PCIE_X4_RC_PF0_AER_CAP_ROOT_ERR_CMD_OFF_0

PCIE_X8_RC_PF0_AER_CAP_ROOT_ERR_CMD_OFF_0

Offset: 0x12c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:3	RO	0x0	**RSVDP_3**: Reserved for future use.

Bit	R/W	Reset	Description
2	RW	0x0	**FATAL_ERR_REPORTING_EN**: Fatal Error Reporting Enable. When Set, this bit enables the generation of an interrupt when a Fatal error is reported by any of the Functions in the Hierarchy Domain associated with this Root Port. Root Complex Event Collectors provide support for the above described functionality for RCiEPs.
1	RW	0x0	**NON_FATAL_ERR_REPORTING_EN**: Non-Fatal Error Reporting Enable. When Set, this bit enables the generation of an interrupt when a Non-fatal error is reported by any of the Functions in the Hierarchy Domain associated with this Root Port. Root Complex Event Collectors provide support for the above described functionality for RCiEPs.
0	RW	0x0	**CORR_ERR_REPORTING_EN**: Correctable Error Reporting Enable. When Set, this bit enables the generation of an interrupt when a correctable error is reported by any of the Functions in the Hierarchy Domain associated with this Root Port. Root Complex Event Collectors provide support for the above described functionality for RCiEPs (Root Complex integrated Endpoint).

PCIE_X<*i*>_RC_PF0_AER_CAP_ROOT_ERR_STATUS_OFF_0

where <*i*> = 1, 4, 8.

Description: The Root Error Status Register reports status of error Messages (ERR_COR, ERR_NONFATAL, and ERR_FATAL) received by the Root Port, and of errors detected by the Root Port itself(The Root Port had sent an error Message to itself). In order to update this register, error Messages received by the Root Port and/or internally generated error Messages must be enabled for transmission by the primary interface of the Root Port.

PCIE_X1_RC_PF0_AER_CAP_ROOT_ERR_STATUS_OFF_0

PCIE_X4_RC_PF0_AER_CAP_ROOT_ERR_STATUS_OFF_0

PCIE_X8_RC_PF0_AER_CAP_ROOT_ERR_STATUS_OFF_0

Offset: 0x130
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:27	RW	0x0	**ADV_ERR_INT_MSG_NUM**: Advanced Error Interrupt Message Number. This register indicates which MSI/MSI-X vector is used for the interrupt message generated in association with any of the status bits of this Capability. For MSI, the value in this register indicates the offset between the base Message Data and the interrupt message that is generated. For MSI-X, the value in this register indicates which MSI-X Table entry is used to generate the interrupt message. The entry must be one of the first 32 entries even if the Function implements more than 32 entries. For a given MSI-X implementation, the entry must remain constant. If MSI-X is enabled, the value in this register must indicate the vector for MSI-X. If MSI is enabled or neither is enabled, the value in this register must indicate the vector for MSI. If software enables both MSI and MSI-X at the same time, the value in this register is undefined. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
26:7	RO	0x0	**RSVDP_7**: Reserved for future use.
6	RW	0x0	**FATAL_ERR_MSG_RX**: One or more Fatal Error Messages Received. Set when one or more Fatal Uncorrectable error Messages have been received.
5	RW	0x0	**NON_FATAL_ERR_MSG_RX**: One or more Non-Fatal Error Messages Received. Set when one or more Non-Fatal Uncorrectable error Messages have been received.
4	RW	0x0	**FIRST_UNCORR_FATAL**: First Uncorrectable Error is Fatal. Set when the first Uncorrectable error Message received is for a Fatal error.
3	RW	0x0	**MUL_ERR_FATAL_NON_FATAL_RX**: Multiple Fatal or Non-Fatal Errors Received. Set when either a Fatal or a Non-fatal error is received and ERR_FATAL_NON_FATAL_RX is already Set.
2	RW	0x0	**ERR_FATAL_NON_FATAL_RX**: Fatal or Non-Fatal Error Received. Set when either a Fatal or a Non-fatal error Message is received and this bit is not already Set.
1	RW	0x0	**MUL_ERR_COR_RX**: Multiple Correctable Errors Received. Set when a Correctable error Message is received and ERR_COR_RX is already Set.
0	RW	0x0	**ERR_COR_RX**: Correctable Error Received. Set when a Correctable error Message is received and this bit is not already Set.

PCIE_X<*i*>_RC_PF0_AER_CAP_ERR_SRC_ID_OFF_0

where <*i*> = 1, 4, 8.

Description: The Error Source Identification Register identifies the source (Requester ID) of first correctable and uncorrectable (Non-fatal/Fatal) errors reported in the Root Error Status Register.

This register is updated regardless of the settings of the Root Control register and the Root Error Command Register.

PCIE_X1_RC_PF0_AER_CAP_ERR_SRC_ID_OFF_0

PCIE_X4_RC_PF0_AER_CAP_ERR_SRC_ID_OFF_0

PCIE_X8_RC_PF0_AER_CAP_ERR_SRC_ID_OFF_0

Offset: 0x134
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**ERR_FATAL_NON_FATAL_SOURCE_ID**: Source of Fatal/Non-Fatal Error. Loaded with the Requester ID indicated in the received ERR_FATAL or ERR_NONFATAL Message when the ERR_FATAL/NONFATAL Received bit is not already set. Note: This register field is sticky.
15:0	0x0	**ERR_COR_SOURCE_ID**: Source of Correctable Error. Loaded with the Requester ID indicated in the received ERR_COR Message when the ERR_COR Received bit is not already set. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_AER_CAP_TLP_PREFIX_LOG_1_OFF_0

where <*i*> = 1, 4, 8.

Description: The First TLP Prefix Log Register contains the first End-End TLP Prefix from the TLP corresponding to the detected error. The TLP Prefix Log Register is only meaningful when the TLP Prefix Log Present bit is Set. The TLP Prefixes are captured such that, when read using DW accesses, the fields of the TLP Prefix are laid out in the same way the fields of the TLP Prefix are described. Therefore, byte 0 of a TLP Prefix is located in byte 3 of the associated TLP Prefix Log Register; byte 1 of a TLP Prefix is located in byte 2; and so forth. The TLP Prefix Log Registers beyond the number supported by the Function are hardwired to zero by controller. If the End-End TLP Prefix Supported bit is Clear, the TLP Prefix Log Register is not required to be implemented.

PCIE_X1_RC_PF0_AER_CAP_TLP_PREFIX_LOG_1_OFF_0

PCIE_X4_RC_PF0_AER_CAP_TLP_PREFIX_LOG_1_OFF_0

PCIE_X8_RC_PF0_AER_CAP_TLP_PREFIX_LOG_1_OFF_0

Offset: 0x138
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**CFG_TLP_PFX_LOG_1_FOURTH_BYTE**: Byte 3 of Error TLP Prefix Log 1. This field contains fourth byte of First DW of TLP Prefix. Note: This register field is sticky.
23:16	0x0	**CFG_TLP_PFX_LOG_1_THIRD_BYTE**: Byte 2 of Error TLP Prefix Log 1. This field contains third byte of First DW of TLP Prefix. Note: This register field is sticky.
15:8	0x0	**CFG_TLP_PFX_LOG_1_SECOND_BYTE**: Byte 1 of Error TLP Prefix Log 1. This field contains second byte of First DW of TLP Prefix. Note: This register field is sticky.
7:0	0x0	**CFG_TLP_PFX_LOG_1_FIRST_BYTE**: Byte 0 of Error TLP Prefix Log 1. This field contains first byte of First DW of TLP Prefix. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_AER_CAP_TLP_PREFIX_LOG_2_OFF_0

where <*i*> = 1, 4, 8.

Description: The Second TLP Prefix Log Register contains the second End-End TLP Prefix from the TLP corresponding to the detected error. The TLP Prefix Log Register is only meaningful when the TLP Prefix Log Present bit is Set. The TLP Prefixes are captured such that, when read using DW accesses, the fields of the TLP Prefix are laid out in the same way the fields of the TLP Prefix are described. Therefore, byte 0 of a TLP Prefix is located in byte 3 of the associated TLP Prefix Log Register; byte 1 of a TLP Prefix is located in byte 2; and so forth. The TLP Prefix Log Registers beyond the number supported by the Function are hardwired to zero by controller. If the End-End TLP Prefix Supported bit is Clear, the TLP Prefix Log Register is not required to be implemented.

PCIE_X1_RC_PF0_AER_CAP_TLP_PREFIX_LOG_2_OFF_0

PCIE_X4_RC_PF0_AER_CAP_TLP_PREFIX_LOG_2_OFF_0

PCIE_X8_RC_PF0_AER_CAP_TLP_PREFIX_LOG_2_OFF_0

Offset: 0x13c
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**CFG_TLP_PFX_LOG_2_FOURTH_BYTE**: Byte 3 Error TLP Prefix Log 2. This field contains fourth byte of Second DW of TLP Prefix. Note: This register field is sticky.
23:16	0x0	**CFG_TLP_PFX_LOG_2_THIRD_BYTE**: Byte 2 Error TLP Prefix Log 2. This field contains third byte of Second DW of TLP Prefix. Note: This register field is sticky.
15:8	0x0	**CFG_TLP_PFX_LOG_2_SECOND_BYTE**: Byte 1 Error TLP Prefix Log 2. This field contains second byte of Second DW of TLP Prefix. Note: This register field is sticky.
7:0	0x0	**CFG_TLP_PFX_LOG_2_FIRST_BYTE**: Byte 0 Error TLP Prefix Log 2. This field contains first byte of Second DW of TLP Prefix. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_AER_CAP_TLP_PREFIX_LOG_3_OFF_0

where <*i*> = 1, 4, 8.

Description: The Third TLP Prefix Log Register contains the third End-End TLP Prefix from the TLP corresponding to the detected error. The TLP Prefix Log Register is only meaningful when the TLP Prefix Log Present bit is Set. The TLP Prefixes are captured such that, when read using DW accesses, the fields of the TLP Prefix are laid out in the same way the fields of the TLP Prefix are described. Therefore, byte 0 of a TLP Prefix is located in byte 3 of the associated TLP Prefix Log Register; byte 1 of a TLP Prefix is located in byte 2; and so forth. The TLP Prefix Log Registers beyond the number supported by the Function are hardwired to zero by controller. If the End-End TLP Prefix Supported bit is Clear, the TLP Prefix Log Register is not required to be implemented.

PCIE_X1_RC_PF0_AER_CAP_TLP_PREFIX_LOG_3_OFF_0

PCIE_X4_RC_PF0_AER_CAP_TLP_PREFIX_LOG_3_OFF_0

PCIE_X8_RC_PF0_AER_CAP_TLP_PREFIX_LOG_3_OFF_0

Offset: 0x140
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**CFG_TLP_PFX_LOG_3_FOURTH_BYTE**: Byte 3 Error TLP Prefix Log 3. This field contains fourth byte of Third DW of TLP Prefix. Note: This register field is sticky.
23:16	0x0	**CFG_TLP_PFX_LOG_3_THIRD_BYTE**: Byte 2 Error TLP Prefix Log 3. This field contains third byte of Third DW of TLP Prefix. Note: This register field is sticky.
15:8	0x0	**CFG_TLP_PFX_LOG_3_SECOND_BYTE**: Byte 1 Error TLP Prefix Log 3. This field contains second byte of Third DW of TLP Prefix. Note: This register field is sticky.
7:0	0x0	**CFG_TLP_PFX_LOG_3_FIRST_BYTE**: Byte 0 Error TLP Prefix Log 3. This field contains first byte of Third DW of TLP Prefix. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_AER_CAP_TLP_PREFIX_LOG_4_OFF_0

where <*i*> = 1, 4, 8.

Description: The Fourth TLP Prefix Log Register contains the fourth End-End TLP Prefix from the TLP corresponding to the detected error. The TLP Prefix Log Register is only meaningful when the TLP Prefix Log Present bit is Set. The TLP Prefixes are captured such that, when read using DW accesses, the fields of the TLP Prefix are laid out in the same way the fields of the TLP Prefix are described. Therefore, byte 0 of a TLP Prefix is located in byte 3 of the associated TLP Prefix Log Register; byte 1 of a TLP Prefix is located in byte 2; and so forth. The TLP Prefix Log Registers beyond the number supported by the Function are hardwired to zero by controller. If the End-End TLP Prefix Supported bit is Clear, the TLP Prefix Log Register is not required to be implemented.

PCIE_X1_RC_PF0_AER_CAP_TLP_PREFIX_LOG_4_OFF_0

PCIE_X4_RC_PF0_AER_CAP_TLP_PREFIX_LOG_4_OFF_0

PCIE_X8_RC_PF0_AER_CAP_TLP_PREFIX_LOG_4_OFF_0

Offset: 0x144
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**CFG_TLP_PFX_LOG_4_FOURTH_BYTE**: Byte 3 Error TLP Prefix Log 4. This field contains fourth byte of Fourth DW of TLP Prefix. Note: This register field is sticky.

Bit	Reset	Description
23:16	0x0	**CFG_TLP_PFX_LOG_4_THIRD_BYTE**: Byte 2 Error TLP Prefix Log 4. This field contains third byte of Fourth DW of TLP Prefix. Note: This register field is sticky.
15:8	0x0	**CFG_TLP_PFX_LOG_4_SECOND_BYTE**: Byte 1 Error TLP Prefix Log 4. This field contains second byte of Fourth DW of TLP Prefix. Note: This register field is sticky.
7:0	0x0	**CFG_TLP_PFX_LOG_4_FIRST_BYTE**: Byte 0 Error TLP Prefix Log 4. This field contains first byte of Fourth DW of TLP Prefix. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_SPCIE_CAP_SPCIE_CAP_HEADER_REG_0

where <*i*> = 1, 4, 8.

Description: This Register provides Capability Id, Capability Version, and next Offset of SPCIE Structure.

Offset: 0x148
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_SPCIE_CAP_SPCIE_CAP_HEADER_REG_0

Reset: 0x15810019 (0b0001,0101,1000,0001,0000,0000,0001,1001)

Bit	Reset	Description
31:20	0x158	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Extended Capability structure or 000h if no other items exist in the linked list of Capabilities. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x19	**EXTENDED_CAP_ID**: Secondary PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. PCI Express Extended Capability ID for the Secondary PCI Express Extended Capability is 0019h. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X4_RC_PF0_SPCIE_CAP_SPCIE_CAP_HEADER_REG_0

PCIE_X8_RC_PF0_SPCIE_CAP_SPCIE_CAP_HEADER_REG_0

Reset: 0x16810019 (0b0001,0110,1000,0001,0000,0000,0001,1001)

Bit	Reset	Description
31:20	0x168	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Extended Capability structure or 000h if no other items exist in the linked list of Capabilities. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x19	**EXTENDED_CAP_ID**: Secondary PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. PCI Express Extended Capability ID for the Secondary PCI Express Extended Capability is 0019h. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_SPCIE_CAP_LINK_CONTROL3_REG_0

where <*i*> = 1, 4, 8.

Description: This Register controls equilization and equilization interrupt.

PCIE_X1_RC_PF0_SPCIE_CAP_LINK_CONTROL3_REG_0

PCIE_X4_RC_PF0_SPCIE_CAP_LINK_CONTROL3_REG_0

PCIE_X8_RC_PF0_SPCIE_CAP_LINK_CONTROL3_REG_0

Offset: 0x14c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:2	RO	0x0	**RSVDP_2**: Reserved for future use.
1	RW	0x0	**EQ_REQ_INT_EN**: Link Equalization Request Interrupt Enable. This bit is RW for Downstream Ports and for Upstream Ports when Crosslink Supported is 1b. This bit is not applicable and is RsvdP for Upstream Ports when the Crosslink Supported bit is 0b. If the Port does not support 8.0 GT/s, this bit is permitted to be hardwired to 0b by the controller. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
0	RW	0x0	**PERFORM_EQ**: Perform Equalization. This bit is RW for Downstream Ports and for Upstream Ports when Crosslink Supported is 1b. This bit is not applicable and is RsvdP for Upstream Ports when the Crosslink Supported bit is 0b. If the Port does not support 8.0 GT/s, this bit is permitted to be hardwired to 0b by the controller. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_X<*i*>_RC_PF0_SPCIE_CAP_LANE_ERR_STATUS_REG_0

where <*i*> = 1, 4, 8.

Description: This Register contains Lane Error Status Bits per Lane.

PCIE_X1_RC_PF0_SPCIE_CAP_LANE_ERR_STATUS_REG_0

PCIE_X4_RC_PF0_SPCIE_CAP_LANE_ERR_STATUS_REG_0

PCIE_X8_RC_PF0_SPCIE_CAP_LANE_ERR_STATUS_REG_0

Offset: 0x150
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:1	RO	0x0	**RSVDP_LANE_ERR_STATUS**: Reserved for future use.
0	RW	0x0	**LANE_ERR_STATUS**: Lane Error Status Bits per Lane. Each bit indicates if the corresponding Lane detected a Lane-based error. A value of 1b indicates that a Lane based-error was detected on the corresponding Lane Number. For Ports that are narrower than 32 Lanes, the unused upper bits [31: Maximum Link Width] are RsvdZ. For Ports that do not support 8.0 GT/s and do not set these bits based on 8b/10b errors , this field is permitted to be hardwired to 0 by the controller.

PCIE_X<*i*>_RC_PF0_SPCIE_CAP_SPCIE_CAP_OFF_0CH_REG_0

where <*i*> = 1, 4, 8.

Description: This register provides Transmitter Preset and Receiver Preset Hint for Downstream Port and Upstream Port.

Offset: 0x154
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_SPCIE_CAP_SPCIE_CAP_OFF_0CH_REG_0

Reset: 0x00007575 (0b0xxx,xxxx,0xxx,xxxx,0111,0101,0111,0101)

Bit	R/W	Reset	Description
31	RO	0x0	**RSVDP_31**: Reserved for future use.
23	RO	0x0	**RSVDP_23**: Reserved for future use.
15	RO	0x0	**RSVDP_15**: Reserved for future use.
14:12	RW	0x7	**USP_RX_PRESET_HINT0**: Upstream Port 8.0 GT/s Receiver Preset Hint 0. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Receiver preset hint 0 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: DSP || USP && PCIE_CAP_CROSS_LINK_SUPPORT ? (if (DBI_RO_WR_EN == 1) then R/W (sticky) else R(sticky)) : ROS

Bit	R/W	Reset	Description
11:8	RW	0x5	**USP_TX_PRESET0**: Upstream Port 8.0 GT/s Transmitter Preset 0. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Transmitter preset 0 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: DSP || USP && PCIE_CAP_CROSS_LINK_SUPPORT ? (if (DBI_RO_WR_EN == 1) then R/W (sticky) else R(sticky)) : ROS
7	RO	0x0	**RSVDP_7**: Reserved for future use.
6:4	RW	0x7	**DSP_RX_PRESET_HINT0**: Downstream Port 8.0 GT/s Receiver Preset Hint 0. Receiver preset hint 0 value that may be used as a suggested setting for 8.0 GT/s receiver equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky)
3:0	RW	0x5	**DSP_TX_PRESET0**: Downstream Port 8.0 GT/s Transmitter Preset 0. Transmitter preset 0 value used for 8.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky)

PCIE_X4_RC_PF0_SPCIE_CAP_SPCIE_CAP_OFF_0CH_REG_0

PCIE_X8_RC_PF0_SPCIE_CAP_SPCIE_CAP_OFF_0CH_REG_0

Reset: 0x75757575 (0b0111,0101,0111,0101,0111,0101,0111,0101)

Bit	R/W	Reset	Description
Bit	**R/W**	**Reset**	**Description**
31	RO	0x0	**RSVDP_31**: Reserved for future use.

Bit	R/W	Reset	Description
Bit	**R/W**	**Reset**	**Description**
30:28	RW	0x7	**USP_RX_PRESET_HINT1**: Upstream Port 8.0 GT/s Receiver Preset Hint 1. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Receiver preset hint 1 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: DSP || USP && PCIE_CAP_CROSS_LINK_SUPPORT ? (if (DBI_RO_WR_EN == 1) then R/W (sticky) else R(sticky)) : ROS
27:24	RW	0x5	**USP_TX_PRESET1**: Upstream Port 8.0 GT/s Transmitter Preset 1. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Transmitter preset 1 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: DSP || USP && PCIE_CAP_CROSS_LINK_SUPPORT ? (if (DBI_RO_WR_EN == 1) then R/W (sticky) else R(sticky)) : ROS
23	RO	0x0	**RSVDP_23**: Reserved for future use.
22:20	RW	0x7	**DSP_RX_PRESET_HINT1**: Downstream Port 8.0 GT/s Receiver Preset Hint 1. Receiver preset hint 1 value that may be used as a suggested setting for 8.0 GT/s receiver equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky)
19:16	RW	0x5	**DSP_TX_PRESET1**: Downstream Port 8.0 GT/s Transmitter Preset 1. Transmitter preset 1 value used for 8.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky)

Bit	R/W	Reset	Description
Bit	**R/W**	**Reset**	**Description**
15	RO	0x0	**RSVDP_15**: Reserved for future use.
14:12	RW	0x7	**USP_RX_PRESET_HINT0**: Upstream Port 8.0 GT/s Receiver Preset Hint 0. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Receiver preset hint 0 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: DSP || USP && PCIE_CAP_CROSS_LINK_SUPPORT ? (if (DBI_RO_WR_EN == 1) then R/W (sticky) else R(sticky)) : ROS
11:8	RW	0x5	**USP_TX_PRESET0**: Upstream Port 8.0 GT/s Transmitter Preset 0. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Transmitter preset 0 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: DSP || USP && PCIE_CAP_CROSS_LINK_SUPPORT ? (if (DBI_RO_WR_EN == 1) then R/W (sticky) else R(sticky)) : ROS
7	RO	0x0	**RSVDP_7**: Reserved for future use.
6:4	RW	0x7	**DSP_RX_PRESET_HINT0**: Downstream Port 8.0 GT/s Receiver Preset Hint 0. Receiver preset hint 0 value that may be used as a suggested setting for 8.0 GT/s receiver equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky)

Bit	R/W	Reset	Description
Bit	**R/W**	**Reset**	**Description**
3:0	RW	0x5	**DSP_TX_PRESET0**: Downstream Port 8.0 GT/s Transmitter Preset 0. Transmitter preset 0 value used for 8.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky)

PCIE_X<*i*>_RC_PF0_PL16G_CAP_PL16G_EXT_CAP_HDR_REG_0

where <*i*> = 1, 4, 8.

Description: Physical Layer 16.0 GT/s Extended Capability Header provides information about Capability ID, Version, and next offset.

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_PL16G_CAP_PL16G_EXT_CAP_HDR_REG_0

Offset: 0x158
Reset: 0x17c10026 (0b0001,0111,1100,0001,0000,0000,0010,0110)

Bit	Reset	Description
31:20	0x17c	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. This bit depends on the version of the specification. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

Bit	Reset	Description
15:0	0x26	**EXTENDED_CAP_ID**: PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. The Extended Capability ID for the Physical Layer 16.0 GT/s Capability is 0026h. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X4_RC_PF0_PL16G_CAP_PL16G_EXT_CAP_HDR_REG_0

Offset: 0x168
Reset: 0x18c10026 (0b0001,1000,1100,0001,0000,0000,0010,0110)

Bit	Reset	Description
31:20	0x18c	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. This bit depends on the version of the specification. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x26	**EXTENDED_CAP_ID**: PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. The Extended Capability ID for the Physical Layer 16.0 GT/s Capability is 0026h. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X8_RC_PF0_PL16G_CAP_PL16G_EXT_CAP_HDR_REG_0

Offset: 0x168
Reset: 0x19010026 (0b0001,1001,0000,0001,0000,0000,0010,0110)

Bit	Reset	Description
31:20	0x190	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

Bit	Reset	Description
19:16	0x1	**CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. This bit depends on the version of the specification. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x26	**EXTENDED_CAP_ID**: PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. The Extended Capability ID for the Physical Layer 16.0 GT/s Capability is 0026h. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*j*>_RC_PF0_SPCIE_CAP_SPCIE_CAP_OFF_10H_REG_0

where <*j*> = 4, 8.

PCIE_X4_RC_PF0_SPCIE_CAP_SPCIE_CAP_OFF_10H_REG_0

PCIE_X8_RC_PF0_SPCIE_CAP_SPCIE_CAP_OFF_10H_REG_0

Description: The function of this register is dependent on your actual configuration. - Gen3: LEC or RSVD depending on the value of CX_NL. - Gen4: LEC or LEC2 or RSVD depending on the value of CX_NL. This register provides Transmitter Preset and Receiver Preset Hint for Downstream Port and Upstream Port.

Offset: 0x158
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x75757575 (0b0111,0101,0111,0101,0111,0101,0111,0101)

Bit	R/W	Reset	Description
31	RO	0x0	**RSVDP_31**: Reserved for future use.

Bit	R/W	Reset	Description
30:28	RW	0x7	**USP_RX_PRESET_HINT3**: Upstream Port 8.0 GT/s Receiver Preset Hint3. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Receiver preset hint 3 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: DSP || USP && PCIE_CAP_CROSS_LINK_SUPPORT ? (if (DBI_RO_WR_EN == 1) then R/W (sticky) else R(sticky)) : ROS
27:24	RW	0x5	**USP_TX_PRESET3**: Upstream Port 8.0 GT/s Transmitter Preset3. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Transmitter preset 3 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: DSP || USP && PCIE_CAP_CROSS_LINK_SUPPORT ? (if (DBI_RO_WR_EN == 1) then R/W (sticky) else R(sticky)) : ROS
23	RO	0x0	**RSVDP_23**: Reserved for future use.
22:20	RW	0x7	**DSP_RX_PRESET_HINT3**: Downstream Port 8.0 GT/s Receiver Preset Hint3. Receiver preset hint 3 value that may be used as a suggested setting for 8.0 GT/s receiver equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky)
19:16	RW	0x5	**DSP_TX_PRESET3**: Downstream Port 8.0 GT/s Transmitter Preset3. Transmitter preset 3 value used for 8.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky)
15	RO	0x0	**RSVDP_15**: Reserved for future use.

Bit	R/W	Reset	Description
14:12	RW	0x7	**USP_RX_PRESET_HINT2**: Upstream Port 8.0 GT/s Receiver Preset Hint2. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Receiver preset hint 2 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: DSP || USP && PCIE_CAP_CROSS_LINK_SUPPORT ? (if (DBI_RO_WR_EN == 1) then R/W (sticky) else R(sticky)) : ROS
11:8	RW	0x5	**USP_TX_PRESET2**: Upstream Port 8.0 GT/s Transmitter Preset2. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Transmitter preset 2 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: DSP || USP && PCIE_CAP_CROSS_LINK_SUPPORT ? (if (DBI_RO_WR_EN == 1) then R/W (sticky) else R(sticky)) : ROS
7	RO	0x0	**RSVDP_7**: Reserved for future use.
6:4	RW	0x7	**DSP_RX_PRESET_HINT2**: Downstream Port 8.0 GT/s Receiver Preset Hint2. Receiver preset hint 2 value that may be used as a suggested setting for 8.0 GT/s receiver equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky)
3:0	RW	0x5	**DSP_TX_PRESET2**: Downstream Port 8.0 GT/s Transmitter Preset2. Transmitter preset 2 value used for 8.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky)

PCIE_X8_RC_PF0_SPCIE_CAP_SPCIE_CAP_OFF_14H_REG_0

Description: The function of this register is dependent on your actual configuration. - Gen3: LEC or RSVD depending on the value of CX_NL. - Gen4: LEC or LEC2 or RSVD depending on the value of CX_NL. This register provides Transmitter Preset and Receiver Preset Hint for Downstream Port and Upstream Port.

Offset: 0x15c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x75757575 (0b0111,0101,0111,0101,0111,0101,0111,0101)

Bit	R/W	Reset	Description
31	RO	0x0	**RSVDP_31**: Reserved for future use.
30:28	RW	0x7	**USP_RX_PRESET_HINT5**: Upstream Port 8.0 GT/s Receiver Preset Hint5. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Receiver preset hint 5 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: DSP || USP && PCIE_CAP_CROSS_LINK_SUPPORT ? (if (DBI_RO_WR_EN == 1) then R/W (sticky) else R(sticky)) : ROS
27:24	RW	0x5	**USP_TX_PRESET5**: Upstream Port 8.0 GT/s Transmitter Preset5. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Transmitter preset 5 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: DSP || USP && PCIE_CAP_CROSS_LINK_SUPPORT ? (if (DBI_RO_WR_EN == 1) then R/W (sticky) else R(sticky)) : ROS

Bit	R/W	Reset	Description
23	RO	0x0	**RSVDP_23**: Reserved for future use.
22:20	RW	0x7	**DSP_RX_PRESET_HINT5**: Downstream Port 8.0 GT/s Receiver Preset Hint5. Receiver preset hint 5 value that may be used as a suggested setting for 8.0 GT/s receiver equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky)
19:16	RW	0x5	**DSP_TX_PRESET5**: Downstream Port 8.0 GT/s Transmitter Preset5. Transmitter preset 5 value used for 8.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky)
15	RO	0x0	**RSVDP_15**: Reserved for future use.
14:12	RW	0x7	**USP_RX_PRESET_HINT4**: Upstream Port 8.0 GT/s Receiver Preset Hint4. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Receiver preset hint 4 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: DSP || USP && PCIE_CAP_CROSS_LINK_SUPPORT ? (if (DBI_RO_WR_EN == 1) then R/W (sticky) else R(sticky)) : ROS
11:8	RW	0x5	**USP_TX_PRESET4**: Upstream Port 8.0 GT/s Transmitter Preset4. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Transmitter preset 4 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: DSP || USP && PCIE_CAP_CROSS_LINK_SUPPORT ? (if (DBI_RO_WR_EN == 1) then R/W (sticky) else R(sticky)) : ROS

Bit	R/W	Reset	Description
7	RO	0x0	**RSVDP_7**: Reserved for future use.
6:4	RW	0x7	**DSP_RX_PRESET_HINT4**: Downstream Port 8.0 GT/s Receiver Preset Hint4. Receiver preset hint 4 value that may be used as a suggested setting for 8.0 GT/s receiver equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky)
3:0	RW	0x5	**DSP_TX_PRESET4**: Downstream Port 8.0 GT/s Transmitter Preset4. Transmitter preset 4 value used for 8.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky)

PCIE_X8_RC_PF0_SPCIE_CAP_SPCIE_CAP_OFF_18H_REG_0

Description: The function of this register is dependent on your actual configuration. - Gen3: LEC or RSVD depending on the value of CX_NL. - Gen4: LEC or LEC2 or RSVD depending on the value of CX_NL. This register provides Transmitter Preset and Receiver Preset Hint for Downstream Port and Upstream Port.

Offset: 0x160
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x75757575 (0b0111,0101,0111,0101,0111,0101,0111,0101)

Bit	R/W	Reset	Description
Bit	R/W	Reset	Description
31	RO	0x0	**RSVDP_31**: Reserved for future use.

Bit	R/W	Reset	Description
Bit	**R/W**	**Reset**	**Description**
30:28	RW	0x7	**USP_RX_PRESET_HINT7**: Upstream Port 8.0 GT/s Receiver Preset Hint7. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Receiver preset hint 7 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: DSP || USP && PCIE_CAP_CROSS_LINK_SUPPORT ? (if (DBI_RO_WR_EN == 1) then R/W (sticky) else R(sticky)) : ROS
27:24	RW	0x5	**USP_TX_PRESET7**: Upstream Port 8.0 GT/s Transmitter Preset7. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Transmitter preset 7 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: DSP || USP && PCIE_CAP_CROSS_LINK_SUPPORT ? (if (DBI_RO_WR_EN == 1) then R/W (sticky) else R(sticky)) : ROS
23	RO	0x0	**RSVDP_23**: Reserved for future use.
22:20	RW	0x7	**DSP_RX_PRESET_HINT7**: Downstream Port 8.0 GT/s Receiver Preset Hint7. Receiver preset hint 7 value that may be used as a suggested setting for 8.0 GT/s receiver equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky)
19:16	RW	0x5	**DSP_TX_PRESET7**: Downstream Port 8.0 GT/s Transmitter Preset7. Transmitter preset 7 value used for 8.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky)

Bit	R/W	Reset	Description
Bit	**R/W**	**Reset**	**Description**
15	RO	0x0	**RSVDP_15**: Reserved for future use.
14:12	RW	0x7	**USP_RX_PRESET_HINT6**: Upstream Port 8.0 GT/s Receiver Preset Hint6. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Receiver preset hint 6 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: DSP || USP && PCIE_CAP_CROSS_LINK_SUPPORT ? (if (DBI_RO_WR_EN == 1) then R/W (sticky) else R(sticky)) : ROS
11:8	RW	0x5	**USP_TX_PRESET6**: Upstream Port 8.0 GT/s Transmitter Preset6. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Transmitter preset 6 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: DSP || USP && PCIE_CAP_CROSS_LINK_SUPPORT ? (if (DBI_RO_WR_EN == 1) then R/W (sticky) else R(sticky)) : ROS
7	RO	0x0	**RSVDP_7**: Reserved for future use.
6:4	RW	0x7	**DSP_RX_PRESET_HINT6**: Downstream Port 8.0 GT/s Receiver Preset Hint6. Receiver preset hint 6 value that may be used as a suggested setting for 8.0 GT/s receiver equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky)

Bit	R/W	Reset	Description
Bit	**R/W**	**Reset**	**Description**
3:0	RW	0x5	**DSP_TX_PRESET6**: Downstream Port 8.0 GT/s Transmitter Preset6. Transmitter preset 6 value used for 8.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky)

PCIE_X<*i*>_RC_PF0_PL16G_CAP_PL16G_CAPABILITY_REG_0

where <*i*> = 1, 4, 8.

Description: This register is reserved for the future update.

PCIE_X1_RC_PF0_PL16G_CAP_PL16G_CAPABILITY_REG_0

Offset: 0x15c

PCIE_X4_RC_PF0_PL16G_CAP_PL16G_CAPABILITY_REG_0

Offset: 0x16c

PCIE_X8_RC_PF0_PL16G_CAP_PL16G_CAPABILITY_REG_0

Offset: 0x16c

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**RSVDP_0**: Reserved for future use.

PCIE_X<*i*>_RC_PF0_PL16G_CAP_PL16G_CONTROL_REG_0

where <*i*> = 1, 4, 8.

Description: This register is reserved for the future update.

PCIE_X1_RC_PF0_PL16G_CAP_PL16G_CONTROL_REG_0

Offset: 0x160

PCIE_X4_RC_PF0_PL16G_CAP_PL16G_CONTROL_REG_0

Offset: 0x170

PCIE_X8_RC_PF0_PL16G_CAP_PL16G_CONTROL_REG_0

Offset: 0x170

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**RSVDP_0**: Reserved for future use.

PCIE_X<*i*>_RC_PF0_PL16G_CAP_PL16G_STATUS_REG_0

where <*i*> = 1, 4, 8.

Description: 16.0 GT/s Status Register provides status of equalization of 16.0 GT/s speed.

PCIE_X1_RC_PF0_PL16G_CAP_PL16G_STATUS_REG_0

Offset: 0x164

PCIE_X4_RC_PF0_PL16G_CAP_PL16G_STATUS_REG_0

Offset: 0x174

PCIE_X8_RC_PF0_PL16G_CAP_PL16G_STATUS_REG_0

Offset: 0x174

Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:5	RO	0x0	**RSVDP_5**: Reserved for future use.
4	RW	0x0	**LINK_EQ_16G_REQ**: Link Equalization Request 16.0GT/s. For a Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions.
3	RO	0x0	**EQ_16G_CPL_P3**: Equalization 16.0GT/s Phase 3 Successful. For a Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions. Note: This register field is sticky.
2	RO	0x0	**EQ_16G_CPL_P2**: Equalization 16.0GT/s Phase 2 Successful. For a Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions. Note: This register field is sticky.
1	RO	0x0	**EQ_16G_CPL_P1**: Equalization 16.0GT/s Phase 1 Successful. For a Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions. Note: This register field is sticky.
0	RO	0x0	**EQ_16G_CPL**: Equalization 16.0GT/s Complete. For a Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PL16G_CAP_PL16G_LC_DPAR_STATUS_REG_0

where <*i*> = 1, 4, 8.

Description: The Local Data Parity Mismatch Status register is a 32-bit vector where each bit indicates if the local receiver detected a Data Parity mismatch on the Lane with the corresponding Lane number. This Lane number is the default Lane number which is invariant to Link width and Lane reversal negotiation that occurs during Link training.

PCIE_X1_RC_PF0_PL16G_CAP_PL16G_LC_DPAR_STATUS_REG_0

Offset: 0x168

PCIE_X4_RC_PF0_PL16G_CAP_PL16G_LC_DPAR_STATUS_REG_0

Offset: 0x178

PCIE_X8_RC_PF0_PL16G_CAP_PL16G_LC_DPAR_STATUS_REG_0

Offset: 0x178

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:1	RO	0x0	**RSVDP_LC_DPAR_STATUS**: Reserved for future use.
0	RW	0x0	**LC_DPAR_STATUS**: Local Data Parity Mismatch Status. Each bit indicates if the corresponding Lane detected a Data Parity mismatch. A value of 1b indicates that a mismatch was detected on the corresponding Lane Number. The default value of each bit is 0b. For Ports that are narrower than 32 Lanes, the unused upper bits [31: MaximumLink Width] are RsvdZ.

PCIE_X<*i*>_RC_PF0_PL16G_CAP_PL16G_FIRST_RETIMER_DPAR_STATUS_REG_0

where <*i*> = 1, 4, 8.

Description: The First Retimer Data Parity Status register is a 32-bit vector where each bit indicates if the first Retimer of a Path detected a Data Parity mismatch on the Lane with the corresponding Lane number. This Lane number is the default Lane number which is invariant to Link width and Lane reversal negotiation that occurs during Link training.

PCIE_X1_RC_PF0_PL16G_CAP_PL16G_FIRST_RETIMER_DPAR_STATUS_REG_0

Offset: 0x16c

PCIE_X4_RC_PF0_PL16G_CAP_PL16G_FIRST_RETIMER_DPAR_STATUS_REG_0

Offset: 0x17c

PCIE_X8_RC_PF0_PL16G_CAP_PL16G_FIRST_RETIMER_DPAR_STATUS_REG_0

Offset: 0x17c

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:1	RO	0x0	**RSVDP_FIRST_RETIMER_DPAR_STATUS**: Reserved for future use.
0	RW	0x0	**FIRST_RETIMER_DPAR_STATUS**: First Retimer Data Parity Mismatch Status. Each bit indicates if the corresponding Lane detected a Data Parity mismatch. A value of 1b indicates that a mismatch was detected on the corresponding Lane Number. The default value of each bit is 0b. The value of this field is undefined when no Retimers are present. For Ports that are narrower than 32 Lanes, the unused upper bits [31: Maximum Link Width] are RsvdZ.

PCIE_X<*i*>_RC_PF0_PL16G_CAP_PL16G_SECOND_RETIMER_DPAR_STATUS_REG_0

where <*i*> = 1, 4, 8.

Description: The Second Retimer Data Parity Status register is a 32-bit vector where each bit indicates if the second Retimer of a Path detected a Data Parity mismatch on the Lane with the corresponding Lane number. This Lane number is the default Lane number which is invariant to Link width and Lane reversal negotiation that occurs during Link training.

PCIE_X1_RC_PF0_PL16G_CAP_PL16G_SECOND_RETIMER_DPAR_STATUS_REG_0

Offset: 0x170

PCIE_X4_RC_PF0_PL16G_CAP_PL16G_SECOND_RETIMER_DPAR_STATUS_REG_0

Offset: 0x180

PCIE_X8_RC_PF0_PL16G_CAP_PL16G_SECOND_RETIMER_DPAR_STATUS_REG_0

Offset: 0x180

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:1	RO	0x0	**RSVDP_SECOND_RETIMER_DPAR_STATUS**: Reserved for future use.

Bit	R/W	Reset	Description
0	RW	0x0	**SECOND_RETIMER_DPAR_STATUS**: Second Retimer Data Parity Mismatch Status. Each bit indicates if the corresponding Lane detected a Data Parity mismatch. A value of 1b indicates that a mismatch was detected on the corresponding Lane Number. The default value of each bit is 0b. The value of this field is undefined when no Retimers are present or only one Retimer is present. For Ports that are narrower than 32 Lanes, the unused upper bits [31: Maximum Link Width] are RsvdZ.

PCIE_X<*i*>_RC_PF0_PL16G_CAP_PL16G_CAP_OFF_20H_REG_0

where <*i*> = 1, 4, 8.

Description: This Equalization Control register consists of control fields required for Lane 0-3 16.0 GT/s equalization.

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_PL16G_CAP_PL16G_CAP_OFF_20H_REG_0

Offset: 0x178
Reset: 0x00000055 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0101,0101)

Bit	Reset	Description
7:4	0x5	**USP_16G_TX_PRESET0**: Upstream Port 16.0 GT/s Transmitter Preset0. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field represents the value sent on Lane 0 during 16.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
3:0	0x5	**DSP_16G_TX_PRESET0**: Downstream Port 16.0 GT/s Transmitter Preset0. Transmitter Preset of Lane 0 used for 16.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R

PCIE_X4_RC_PF0_PL16G_CAP_PL16G_CAP_OFF_20H_REG_0

PCIE_X8_RC_PF0_PL16G_CAP_PL16G_CAP_OFF_20H_REG_0

Offset: 0x188
Reset: 0x55555555 (0b0101,0101,0101,0101,0101,0101,0101,0101)

Bit	Reset	Description
31:28	0x5	**USP_16G_TX_PRESET3**: Upstream Port 16.0 GT/s Transmitter Preset3. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field represents the value sent on Lane 3 during 16.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 3 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
27:24	0x5	**DSP_16G_TX_PRESET3**: Downstream Port 16.0 GT/s Transmitter Preset3. Transmitter Preset of Lane 3 used for 16.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
23:20	0x5	**USP_16G_TX_PRESET2**: Upstream Port 16.0 GT/s Transmitter Preset2. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field represents the value sent on Lane 2 during 16.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 2 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
19:16	0x5	**DSP_16G_TX_PRESET2**: Downstream Port 16.0 GT/s Transmitter Preset2. Transmitter Preset of Lane 2 used for 16.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
15:12	0x5	**USP_16G_TX_PRESET1**: Upstream Port 16.0 GT/s Transmitter Preset1. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field represents the value sent on Lane 1 during 16.0 GT/s equalization. - CaseB: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 1 during Link Equalization. Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R

Bit	Reset	Description
11:8	0x5	**DSP_16G_TX_PRESET1**: Downstream Port 16.0 GT/s Transmitter Preset1. Transmitter Preset of Lane 1 used for 16.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
7:4	0x5	**USP_16G_TX_PRESET0**: Upstream Port 16.0 GT/s Transmitter Preset0. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field represents the value sent on Lane 0 during 16.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
3:0	0x5	**DSP_16G_TX_PRESET0**: Downstream Port 16.0 GT/s Transmitter Preset0. Transmitter Preset of Lane 0 used for 16.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R

PCIE_X8_RC_PF0_PL16G_CAP_PL16G_CAP_OFF_24H_REG_0

Description: This Equalization Control register consists of control fields required for Lane 4-7 16.0 GT/s equalization.)

Offset: 0x18c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x55555555 (0b0101,0101,0101,0101,0101,0101,0101,0101)

Bit	Reset	Description
31:28	0x5	**USP_16G_TX_PRESET7**: Upstream Port 16.0 GT/s Transmitter Preset7. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field represents the value sent on Lane 7 during 16.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 7 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
27:24	0x5	**DSP_16G_TX_PRESET7**: Downstream Port 16.0 GT/s Transmitter Preset7. Transmitter Preset of Lane 7 used for 16.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP . Otherwise, this field is HwInit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
23:20	0x5	**USP_16G_TX_PRESET6**: Upstream Port 16.0 GT/s Transmitter Preset6. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field represents the value sent on Lane 6 during 16.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 6 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
19:16	0x5	**DSP_16G_TX_PRESET6**: Downstream Port 16.0 GT/s Transmitter Preset6. Transmitter Preset of Lane 6 used for 16.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
15:12	0x5	**USP_16G_TX_PRESET5**: Upstream Port 16.0 GT/s Transmitter Preset5. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field represents the value sent on Lane 5 during 16.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 5 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
11:8	0x5	**DSP_16G_TX_PRESET5**: Downstream Port 16.0 GT/s Transmitter Preset5. Transmitter Preset of Lane 5 used for 16.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R

Bit	Reset	Description
7:4	0x5	**USP_16G_TX_PRESET4**: Upstream Port 16.0 GT/s Transmitter Preset4. -Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field represents the value sent on Lane 4 during 16.0 GT/s equalization. -Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 4 during Link Equalization. Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
3:0	0x5	**DSP_16G_TX_PRESET4**: Downstream Port 16.0 GT/s Transmitter Preset4. Transmitter Preset of Lane 4 used for 16.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R

PCIE_X<*i*>_RC_PF0_MARGIN_CAP_MARGIN_EXT_CAP_HDR_REG_0

where <*i*> = 1, 4, 8.

Description: This register provides capbility ID, capability version and next offset value for Margining Extended Capability.

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_MARGIN_CAP_MARGIN_EXT_CAP_HDR_REG_0

Offset: 0x17c
Reset: 0x19010027 (0b0001,1001,0000,0001,0000,0000,0010,0111)

Bit	Reset	Description
31:20	0x190	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greaterthan 0FFh. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

Bit	Reset	Description
19:16	0x1	**CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. This field is depends on version of the specification. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x27	**EXTENDED_CAP_ID**: PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. The Extended Capability ID for the Physical Layer 16.0 GT/s Margining Extended Capability is 0027h. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X4_RC_PF0_MARGIN_CAP_MARGIN_EXT_CAP_HDR_REG_0

Offset: 0x18c
Reset: 0x1ac10027 (0b0001,1010,1100,0001,0000,0000,0010,0111)

Bit	Reset	Description
31:20	0x1ac	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greaterthan 0FFh. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. This field is depends on version of the specification. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x27	**EXTENDED_CAP_ID**: PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. The Extended Capability ID for the Physical Layer 16.0 GT/s Margining Extended Capability is 0027h. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X8_RC_PF0_MARGIN_CAP_MARGIN_EXT_CAP_HDR_REG_0

Offset: 0x190
Reset: 0x1c010027 (0b0001,1100,0000,0001,0000,0000,0010,0111)

Bit	Reset	Description
31:20	0x1c0	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greaterthan 0FFh. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. This field is depends on version of the specification. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x27	**EXTENDED_CAP_ID**: PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. The Extended Capability ID for the Physical Layer 16.0 GT/s Margining Extended Capability is 0027h. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_MARGIN_CAP_MARGIN_PORT_CAPABILITIES_STATUS_REG_0

where <*i*> = 1, 4, 8.

Description: This register indicates the status of the Margining feature.

PCIE_X1_RC_PF0_MARGIN_CAP_MARGIN_PORT_CAPABILITIES_STATUS_REG_0

Offset: 0x180

PCIE_X4_RC_PF0_MARGIN_CAP_MARGIN_PORT_CAPABILITIES_STATUS_REG_0

Offset: 0x190

PCIE_X8_RC_PF0_MARGIN_CAP_MARGIN_PORT_CAPABILITIES_STATUS_REG_0

Offset: 0x194

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000001 (0b0000,0000,0000,0000,0000,0000,0000,0001)

Bit	R/W	Reset	Description
31:18	RO	0x0	**RSVDP_18**: Reserved for future use.
17	RO	0x0	**MARGINING_SOFTWARE_READY**: Margining Software Ready. When Margining uses Driver Software is Set, then this bit, when Set, indicates that the required software has performed the required initialization. The value of this bit is undefined if Margining uses Driver Software is Clear. The default value of this bit is implementation specific. If Margining uses Driver Software is Clear, Margining Ready must be Set no later than 100 ms after the Link trains to 16.0 GT/s. Default value is implementation specific.
16	RO	0x0	**MARGINING_READY**: Margining Ready. Indicates when the Margining feature is ready to accept margining commands. Behavior is undefined if this bit is Clear and, for any Lane, any of the Receiver Number , Margin Type , Usage Model , or Margin Payload fields are written. If Margining uses Driver Software is Set, Margining Ready must be Set no later than 100 ms after the later of Margining Software Ready becoming Set or the link training to 16.0 GT/s.
15:1	RO	0x0	**RSVDP_1**: Reserved for future use.
0	RW	0x1	**MARGINING_USES_DRIVER_SOFTWARE**: Margining uses Driver Software. If Set, indicates that Margining is partially implemented using Device Driver software. Margining Software Ready indicates when this software is initialized. If Clear, Margining does not require device driver software. In this case the value read from Margining Software Ready is undefined. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_MARGIN_CAP_MARGIN_LANE_CNTRL_STATUS0_REG_0

where <*i*> = 1, 4, 8.

Description: The Margining Lane Control Register consists of control fields required for per-Lane margining. The number of entries in this register are sized by Maximum Link Width.

PCIE_X1_RC_PF0_MARGIN_CAP_MARGIN_LANE_CNTRL_STATUS0_REG_0

Offset: 0x184

PCIE_X4_RC_PF0_MARGIN_CAP_MARGIN_LANE_CNTRL_STATUS0_REG_0

Offset: 0x194

PCIE_X8_RC_PF0_MARGIN_CAP_MARGIN_LANE_CNTRL_STATUS0_REG_0

Offset: 0x198

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00009c38 (0b0000,0000,0000,0000,1001,1100,0011,1000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**MARGIN_PAYLOAD_STATUS**: Margin Payload(Status) for Lane 0. This field is only meaningful, when the Margin Type is a defined encoding other than 'No Command'. This field must be reset to the default value if the Port goes to DL_Down status.
23	RO	0x0	**RSVDP_23**: Reserved for future use.
22	RO	0x0	**USAGE_MODEL_STATUS**: Usage Model(Status) for Lane 0. This field must be reset to the default value if the Port goes to DL_Down status.
21:19	RO	0x0	**MARGIN_TYPE_STATUS**: Margin Type(Status) for Lane 0. This field must be reset to the default value if the Port goes to DL_Down status.
18:16	RO	0x0	**RECEIVER_NUMBER_STATUS**: Receiver Number(Status) for Lane 0. For Downstream Ports, this field must be reset to the default value if the Port goes to DL_Down status.
15:8	RW	0x9c	**MARGIN_PAYLOAD**: Margin Payload for Lane 0. This field must be reset to the default value if the Port goes to DL_Down status.
7	RO	0x0	**RSVDP_7**: Reserved for future use.
6	RW	0x0	**USAGE_MODEL**: Usage Model for Lane 0. This field must be reset to the default value if the Port goes to DL_Down status.
5:3	RW	0x7	**MARGIN_TYPE**: Margin Type for Lane 0. The default value is 111b. This field must be reset to the default value if the Port goes to DL_Down status.
2:0	RW	0x0	**RECEIVER_NUMBER**: Receiver Number for Lane 0. This field must be reset to the default value if the Port goes to DL_Down status.

PCIE_X<*j*>_RC_PF0_MARGIN_CAP_MARGIN_LANE_CNTRL_STATUS1_REG_0

where <*j*> = 4, 8.

Description: The Margining Lane Control Register consists of control fields required for per-Lane margining. The number of entries in this register are sized by Maximum Link Width.

PCIE_X4_RC_PF0_MARGIN_CAP_MARGIN_LANE_CNTRL_STATUS1_REG_0

Offset: 0x198

PCIE_X8_RC_PF0_MARGIN_CAP_MARGIN_LANE_CNTRL_STATUS1_REG_0

Offset: 0x19c

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00009c38 (0b0000,0000,0000,0000,1001,1100,0011,1000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**MARGIN_PAYLOAD_STATUS**: Margin Payload(Status) for Lane 1. This field is only meaningful, when the Margin Type is a defined encoding other than 'No Command'. This field must be reset to the default value if the Port goes to DL_Down status.
23	RO	0x0	**RSVDP_23**: Reserved for future use.
22	RO	0x0	**USAGE_MODEL_STATUS**: Usage Model(Status) for Lane 1. This field must be reset to the default value if the Port goes to DL_Down status.
21:19	RO	0x0	**MARGIN_TYPE_STATUS**: Margin Type(Status) for Lane 1. This field must be reset to the default value if the Port goes to DL_Down status.
18:16	RO	0x0	**RECEIVER_NUMBER_STATUS**: Receiver Number(Status) for Lane 1. For Downstream Ports, this field must be reset to the default value if the Port goes to DL_Down status.
15:8	RW	0x9c	**MARGIN_PAYLOAD**: Margin Payload for Lane 1. This field must be reset to the default value if the Port goes to DL_Down status.
7	RO	0x0	**RSVDP_7**: Reserved for future use.
6	RW	0x0	**USAGE_MODEL**: Usage Model for Lane 1. This field must be reset to the default value if the Port goes to DL_Down status.
5:3	RW	0x7	**MARGIN_TYPE**: Margin Type for Lane 1. The default value is 111b. This field must be reset to the default value if the Port goes to DL_Down status.
2:0	RW	0x0	**RECEIVER_NUMBER**: Receiver Number for Lane 1. This field must be reset to the default value if the Port goes to DL_Down status.

PCIE_X<*j*>_RC_PF0_MARGIN_CAP_MARGIN_LANE_CNTRL_STATUS2_REG_0

where <*j*> = 4, 8.

Description: The Margining Lane Control Register consists of control fields required for per-Lane margining. The number of entries in this register are sized by Maximum Link Width.

PCIE_X4_RC_PF0_MARGIN_CAP_MARGIN_LANE_CNTRL_STATUS2_REG_0

Offset: 0x19c

PCIE_X8_RC_PF0_MARGIN_CAP_MARGIN_LANE_CNTRL_STATUS2_REG_0

Offset: 0x1a0

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00009c38 (0b0000,0000,0000,0000,1001,1100,0011,1000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**MARGIN_PAYLOAD_STATUS**: Margin Payload(Status) for Lane 2. This field is only meaningful, when the Margin Type is a defined encoding other than 'No Command'. This field must be reset to the default value if the Port goes to DL_Down status.
23	RO	0x0	**RSVDP_23**: Reserved for future use.
22	RO	0x0	**USAGE_MODEL_STATUS**: Usage Model(Status) for Lane 2. This field must be reset to the default value if the Port goes to DL_Down status.
21:19	RO	0x0	**MARGIN_TYPE_STATUS**: Margin Type(Status) for Lane 2. This field must be reset to the default value if the Port goes to DL_Down status.
18:16	RO	0x0	**RECEIVER_NUMBER_STATUS**: Receiver Number(Status) for Lane 2. For Downstream Ports, this field must be reset to the default value if the Port goes to DL_Down status.
15:8	RW	0x9c	**MARGIN_PAYLOAD**: Margin Payload for Lane 2. This field must be reset to the default value if the Port goes to DL_Down status.
7	RO	0x0	**RSVDP_7**: Reserved for future use.

Bit	R/W	Reset	Description
6	RW	0x0	**USAGE_MODEL**: Usage Model for Lane 2. This field must be reset to the default value if the Port goes to DL_Down status.
5:3	RW	0x7	**MARGIN_TYPE**: Margin Type for Lane 2. The default value is 111b. This field must be reset to the default value if the Port goes to DL_Down status.
2:0	RW	0x0	**RECEIVER_NUMBER**: Receiver Number for Lane 2. This field must be reset to the default value if the Port goes to DL_Down status.

PCIE_X<*j*>_RC_PF0_MARGIN_CAP_MARGIN_LANE_CNTRL_STATUS3_REG_0

where <*j*> = 4, 8.

Description: The Margining Lane Control Register consists of control fields required for per-Lane margining. The number of entries in this register are sized by Maximum Link Width.

PCIE_X4_RC_PF0_MARGIN_CAP_MARGIN_LANE_CNTRL_STATUS3_REG_0

Offset: 0x1a0

PCIE_X8_RC_PF0_MARGIN_CAP_MARGIN_LANE_CNTRL_STATUS3_REG_0

Offset: 0x1a4

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00009c38 (0b0000,0000,0000,0000,1001,1100,0011,1000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**MARGIN_PAYLOAD_STATUS**: Margin Payload(Status) for Lane 3. This field is only meaningful, when the Margin Type is a defined encoding other than 'No Command'. This field must be reset to the default value if the Port goes to DL_Down status.
23	RO	0x0	**RSVDP_23**: Reserved for future use.
22	RO	0x0	**USAGE_MODEL_STATUS**: Usage Model(Status) for Lane 3. This field must be reset to the default value if the Port goes to DL_Down status.
21:19	RO	0x0	**MARGIN_TYPE_STATUS**: Margin Type(Status) for Lane 3. This field must be reset to the default value if the Port goes to DL_Down status.

Bit	R/W	Reset	Description
18:16	RO	0x0	**RECEIVER_NUMBER_STATUS**: Receiver Number(Status) for Lane 3. For Downstream Ports, this field must be reset to the default value if the Port goes to DL_Down status.
15:8	RW	0x9c	**MARGIN_PAYLOAD**: Margin Payload for Lane 3. This field must be reset to the default value if the Port goes to DL_Down status.
7	RO	0x0	**RSVDP_7**: Reserved for future use.
6	RW	0x0	**USAGE_MODEL**: Usage Model for Lane 3. This field must be reset to the default value if the Port goes to DL_Down status.
5:3	RW	0x7	**MARGIN_TYPE**: Margin Type for Lane 3. The default value is 111b. This field must be reset to the default value if the Port goes to DL_Down status.
2:0	RW	0x0	**RECEIVER_NUMBER**: Receiver Number for Lane 3. This field must be reset to the default value if the Port goes to DL_Down status.

PCIE_X8_RC_PF0_MARGIN_CAP_MARGIN_LANE_CNTRL_STATUS4_REG_0

Description: The Margining Lane Control Register consists of control fields required for per-Lane margining. The number of entries in this register are sized by Maximum Link Width.

Offset: 0x1a8
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00009c38 (0b0000,0000,0000,0000,1001,1100,0011,1000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**MARGIN_PAYLOAD_STATUS**: Margin Payload(Status) for Lane 4. This field is only meaningful, when the Margin Type is a defined encoding other than 'No Command'. This field must be reset to the default value if the Port goes to DL_Down status.
23	RO	0x0	**RSVDP_23**: Reserved for future use.
22	RO	0x0	**USAGE_MODEL_STATUS**: Usage Model(Status) for Lane 4. This field must be reset to the default value if the Port goes to DL_Down status.
21:19	RO	0x0	**MARGIN_TYPE_STATUS**: Margin Type(Status) for Lane 4. This field must be reset to the default value if the Port goes to DL_Down status.

Bit	R/W	Reset	Description
18:16	RO	0x0	**RECEIVER_NUMBER_STATUS**: Receiver Number(Status) for Lane 4. For Downstream Ports, this field must be reset to the default value if the Port goes to DL_Down status.
15:8	RW	0x9c	**MARGIN_PAYLOAD**: Margin Payload for Lane 4. This field must be reset to the default value if the Port goes to DL_Down status.
7	RO	0x0	**RSVDP_7**: Reserved for future use.
6	RW	0x0	**USAGE_MODEL**: Usage Model for Lane 4. This field must be reset to the default value if the Port goes to DL_Down status.
5:3	RW	0x7	**MARGIN_TYPE**: Margin Type for Lane 4. The default value is 111b. This field must be reset to the default value if the Port goes to DL_Down status.
2:0	RW	0x0	**RECEIVER_NUMBER**: Receiver Number for Lane 4. This field must be reset to the default value if the Port goes to DL_Down status.

PCIE_X8_RC_PF0_MARGIN_CAP_MARGIN_LANE_CNTRL_STATUS5_REG_0

Description: The Margining Lane Control Register consists of control fields required for per-Lane margining. The number of entries in this register are sized by Maximum Link Width.

Offset: 0x1ac
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00009c38 (0b0000,0000,0000,0000,1001,1100,0011,1000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**MARGIN_PAYLOAD_STATUS**: Margin Payload(Status) for Lane 5. This field is only meaningful, when the Margin Type is a defined encoding other than 'No Command'. This field must be reset to the default value if the Port goes to DL_Down status.
23	RO	0x0	**RSVDP_23**: Reserved for future use.
22	RO	0x0	**USAGE_MODEL_STATUS**: Usage Model(Status) for Lane 5. This field must be reset to the default value if the Port goes to DL_Down status.
21:19	RO	0x0	**MARGIN_TYPE_STATUS**: Margin Type(Status) for Lane 5. This field must be reset to the default value if the Port goes to DL_Down status.

Bit	R/W	Reset	Description
18:16	RO	0x0	**RECEIVER_NUMBER_STATUS**: Receiver Number(Status) for Lane 5. For Downstream Ports, this field must be reset to the default value if the Port goes to DL_Down status.
15:8	RW	0x9c	**MARGIN_PAYLOAD**: Margin Payload for Lane 5. This field must be reset to the default value if the Port goes to DL_Down status.
7	RO	0x0	**RSVDP_7**: Reserved for future use.
6	RW	0x0	**USAGE_MODEL**: Usage Model for Lane 5. This field must be reset to the default value if the Port goes to DL_Down status.
5:3	RW	0x7	**MARGIN_TYPE**: Margin Type for Lane 5. The default value is 111b. This field must be reset to the default value if the Port goes to DL_Down status.
2:0	RW	0x0	**RECEIVER_NUMBER**: Receiver Number for Lane 5. This field must be reset to the default value if the Port goes to DL_Down status.

PCIE_X8_RC_PF0_MARGIN_CAP_MARGIN_LANE_CNTRL_STATUS6_REG_0

Description: The Margining Lane Control Register consists of control fields required for per-Lane margining. The number of entries in this register are sized by Maximum Link Width.

Offset: 0x1b0
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00009c38 (0b0000,0000,0000,0000,1001,1100,0011,1000)

Bit	R/W	Reset	Description
Bit	**R/W**	**Reset**	**Description**
31:24	RO	0x0	**MARGIN_PAYLOAD_STATUS**: Margin Payload(Status) for Lane 6. This field is only meaningful, when the Margin Type is a defined encoding other than 'No Command'. This field must be reset to the default value if the Port goes to DL_Down status.
23	RO	0x0	**RSVDP_23**: Reserved for future use.
22	RO	0x0	**USAGE_MODEL_STATUS**: Usage Model(Status) for Lane 6. This field must be reset to the default value if the Port goes to DL_Down status.

Bit	R/W	Reset	Description
Bit	**R/W**	**Reset**	**Description**
21:19	RO	0x0	**MARGIN_TYPE_STATUS**: Margin Type(Status) for Lane 6. This field must be reset to the default value if the Port goes to DL_Down status.
18:16	RO	0x0	**RECEIVER_NUMBER_STATUS**: Receiver Number(Status) for Lane 6. For Downstream Ports, this field must be reset to the default value if the Port goes to DL_Down status.
15:8	RW	0x9c	**MARGIN_PAYLOAD**: Margin Payload for Lane 6. This field must be reset to the default value if the Port goes to DL_Down status.
7	RO	0x0	**RSVDP_7**: Reserved for future use.
6	RW	0x0	**USAGE_MODEL**: Usage Model for Lane 6. This field must be reset to the default value if the Port goes to DL_Down status.
5:3	RW	0x7	**MARGIN_TYPE**: Margin Type for Lane 6. The default value is 111b. This field must be reset to the default value if the Port goes to DL_Down status.
2:0	RW	0x0	**RECEIVER_NUMBER**: Receiver Number for Lane 6. This field must be reset to the default value if the Port goes to DL_Down status.

PCIE_X8_RC_PF0_MARGIN_CAP_MARGIN_LANE_CNTRL_STATUS7_REG_0

Description: The Margining Lane Control Register consists of control fields required for per-Lane margining. The number of entries in this register are sized by Maximum Link Width.

Offset: 0x1b4
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00009c38 (0b0000,0000,0000,0000,1001,1100,0011,1000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**MARGIN_PAYLOAD_STATUS**: Margin Payload(Status) for Lane 7. This field is only meaningful, when the Margin Type is a defined encoding other than 'No Command'. This field must be reset to the default value if the Port goes to DL_Down status.
23	RO	0x0	**RSVDP_23**: Reserved for future use.

Bit	R/W	Reset	Description
22	RO	0x0	**USAGE_MODEL_STATUS**: Usage Model(Status) for Lane 7. This field must be reset to the default value if the Port goes to DL_Down status.
21:19	RO	0x0	**MARGIN_TYPE_STATUS**: Margin Type(Status) for Lane 7. This field must be reset to the default value if the Port goes to DL_Down status.
18:16	RO	0x0	**RECEIVER_NUMBER_STATUS**: Receiver Number(Status) for Lane 7. For Downstream Ports, this field must be reset to the default value if the Port goes to DL_Down status.
15:8	RW	0x9c	**MARGIN_PAYLOAD**: Margin Payload for Lane 7. This field must be reset to the default value if the Port goes to DL_Down status.
7	RO	0x0	**RSVDP_7**: Reserved for future use.
6	RW	0x0	**USAGE_MODEL**: Usage Model for Lane 7. This field must be reset to the default value if the Port goes to DL_Down status.
5:3	RW	0x7	**MARGIN_TYPE**: Margin Type for Lane 7. The default value is 111b. This field must be reset to the default value if the Port goes to DL_Down status.
2:0	RW	0x0	**RECEIVER_NUMBER**: Receiver Number for Lane 7. This field must be reset to the default value if the Port goes to DL_Down status.

PCIE_X<*i*>_RC_PF0_L1SUB_CAP_L1SUB_CAP_HEADER_REG_0

where <*i*> = 1, 4, 8.

Description: L1 Substates Extended Capability Header provides capability ID, capability version and next offset value for L1 Substates.

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_L1SUB_CAP_L1SUB_CAP_HEADER_REG_0

Offset: 0x190
Reset: 0x1a01001e (0b0001,1010,0000,0001,0000,0000,0001,1110)

Bit	Reset	Description
31:20	0x1a0	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h, if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. The bottom 2 bits of this offset are Reserved and must be implemented as 00b although software must mask them to allow for future uses of these bits. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. This field is depends on version of the specification. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x1e	**EXTENDED_CAP_ID**: L1SUB Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. Extended Capability ID for L1 PM Substates is 001Eh. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X4_RC_PF0_L1SUB_CAP_L1SUB_CAP_HEADER_REG_0

Offset: 0x1ac
Reset: 0x1bc1001e (0b0001,1011,1100,0001,0000,0000,0001,1110)

Bit	Reset	Description
31:20	0x1bc	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h, if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. The bottom 2 bits of this offset are Reserved and must be implemented as 00b although software must mask them to allow for future uses of these bits. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. This field is depends on version of the specification. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x1e	**EXTENDED_CAP_ID**: L1SUB Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. Extended Capability ID for L1 PM Substates is 001Eh. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X8_RC_PF0_L1SUB_CAP_L1SUB_CAP_HEADER_REG_0

Offset: 0x1c0
Reset: 0x1d01001e (0b0001,1101,0000,0001,0000,0000,0001,1110)

Bit	Reset	Description
31:20	0x1d0	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h, if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. The bottom 2 bits of this offset are Reserved and must be implemented as 00b although software must mask them to allow for future uses of these bits. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. This field is depends on version of the specification. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x1e	**EXTENDED_CAP_ID**: L1SUB Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. Extended Capability ID for L1 PM Substates is 001Eh. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_L1SUB_CAP_L1SUB_CAPABILITY_REG_0

where <*i*> = 1, 4, 8.

Description: This register provides extended capability of L1 Substates.

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_L1SUB_CAP_L1SUB_CAPABILITY_REG_0

Offset: 0x194
Reset: 0x00280a1f (0b0000,0000,0010,1000,0000,1010,0001,1111)

Bit	R/W	Reset	Description
31:24	RO	0x0	**RSVDP_24**: Reserved for future use.

Bit	R/W	Reset	Description
23:19	RW	0x5	**PWR_ON_VALUE_SUPPORT**: Port T Power On Value. Along with the Port T_POWER_ON_SCALE field in the L1 PM Substates Capabilities register sets the time (in us) that this Port requires the port on the opposite side of Link to wait in L1.2.Exit after sampling CLKREQ# asserted before actively driving the interface. The value of Port T_POWER_ON is calculated by multiplying the value in this field by the scale value in the Port T_POWER_ON_SCALE field in the L1 PM Substates Capabilities register. Required for all Ports for which either the PCI-PM L1.2 Supported bit is Set, ASPM L1.2 Supported bit is Set, or both are Set, otherwise this field is of type RsvdP. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: L1SUB_CAPABILITY_REG.L1_2_ASPM_SUPPORT|| L1SUB_CAPABILITY_REG.L1_2_PCIPM_SUPPORT ? RWS : RSVDP
18	RO	0x0	**RSVDP_18**: Reserved for future use.
17:16	RW	0x0	**PWR_ON_SCALE_SUPPORT**: Port T Power On Scale. Specifies the scale used for the Port T_POWER_ON_VALUE field in the L1 PM Substates Capabilities register. Range of values are given below. Required for all Ports for which either the PCI-PM L1.2 Supported bit is Set, ASPM L1.2 Supported bit is Set, or both are Set, otherwise this field is of type RsvdP. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: L1SUB_CAPABILITY_REG.L1_2_ASPM_SUPPORT|| L1SUB_CAPABILITY_REG.L1_2_PCIPM_SUPPORT ? RWS : RSVDP
15:8	RW	0xa	**COMM_MODE_SUPPORT**: Port Common Mode Restore Time. Time (in us) required for this Port to re-establish common mode. Required for all Ports for which either the PCI-PM L1.2 Supported bit is Set, ASPM L1.2 Supported bit is Set, or both are Set, otherwise this field is of type RsvdP. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: L1SUB_CAPABILITY_REG.L1_2_ASPM_SUPPORT|| L1SUB_CAPABILITY_REG.L1_2_PCIPM_SUPPORT ? RWS : RSVDP
7:5	RO	0x0	**RSVDP_5**: Reserved for future use.
4	RW	0x1	**L1_PMSUB_SUPPORT**: L1 PM Substates ECN Supported. When Set this field indicates that this Port supports L1 PM Substates. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky)
3	RW	0x1	**L1_1_ASPM_SUPPORT**: ASPM L11 Supported. When Set this field indicates that ASPM L1.1 is supported. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky)
2	RW	0x1	**L1_2_ASPM_SUPPORT**: ASPM L12 Supported. When Set this field indicates that ASPM L1.2 is supported. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky)
1	RW	0x1	**L1_1_PCIPM_SUPPORT**: PCI-PM L11 Supported. When Set this field indicates that PCI-PM L1.1 is supported, and must be Set by all Ports implementing L1 PM Substates. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky)

Bit	R/W	Reset	Description
0	RW	0x1	**L1_2_PCIPM_SUPPORT**: PCI-PM L12 Supported. When Set this field indicates that PCI-PM L1.2 is supported. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky)

PCIE_X4_RC_PF0_L1SUB_CAP_L1SUB_CAPABILITY_REG_0

Offset: 0x1b0

PCIE_X8_RC_PF0_L1SUB_CAP_L1SUB_CAPABILITY_REG_0

Offset: 0x1c4
Reset: 0x00a03c1f (0b0000,0000,1010,0000,0011,1100,0001,1111)

Bit	R/W	Reset	Description
31:24	RO	0x0	**RSVDP_24**: Reserved for future use.
23:19	RW	0x14	**PWR_ON_VALUE_SUPPORT**: Port T Power On Value. Along with the Port T_POWER_ON_SCALE field in the L1 PM Substates Capabilities register sets the time (in us) that this Port requires the port on the opposite side of Link to wait in L1.2.Exit after sampling CLKREQ# asserted before actively driving the interface. The value of Port T_POWER_ON is calculated by multiplying the value in this field by the scale value in the Port T_POWER_ON_SCALE field in the L1 PM Substates Capabilities register. Required for all Ports for which either the PCI-PM L1.2 Supported bit is Set, ASPM L1.2 Supported bit is Set, or both are Set, otherwise this field is of type RsvdP. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: L1SUB_CAPABILITY_REG.L1_2_ASPM_SUPPORT|| L1SUB_CAPABILITY_REG.L1_2_PCIPM_SUPPORT ? RWS : RSVDP
18	RO	0x0	**RSVDP_18**: Reserved for future use.
17:16	RW	0x0	**PWR_ON_SCALE_SUPPORT**: Port T Power On Scale. Specifies the scale used for the Port T_POWER_ON_VALUE field in the L1 PM Substates Capabilities register. Range of values are given below. Required for all Ports for which either the PCI-PM L1.2 Supported bit is Set, ASPM L1.2 Supported bit is Set, or both are Set, otherwise this field is of type RsvdP. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: L1SUB_CAPABILITY_REG.L1_2_ASPM_SUPPORT|| L1SUB_CAPABILITY_REG.L1_2_PCIPM_SUPPORT ? RWS : RSVDP
15:8	RW	0x3c	**COMM_MODE_SUPPORT**: Port Common Mode Restore Time. Time (in us) required for this Port to re-establish common mode. Required for all Ports for which either the PCI-PM L1.2 Supported bit is Set, ASPM L1.2 Supported bit is Set, or both are Set, otherwise this field is of type RsvdP. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: L1SUB_CAPABILITY_REG.L1_2_ASPM_SUPPORT|| L1SUB_CAPABILITY_REG.L1_2_PCIPM_SUPPORT ? RWS : RSVDP
7:5	RO	0x0	**RSVDP_5**: Reserved for future use.
4	RW	0x1	**L1_PMSUB_SUPPORT**: L1 PM Substates ECN Supported. When Set this field indicates that this Port supports L1 PM Substates. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky)

Bit	R/W	Reset	Description
3	RW	0x1	**L1_1_ASPM_SUPPORT**: ASPM L11 Supported. When Set this field indicates that ASPM L1.1 is supported. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky)
2	RW	0x1	**L1_2_ASPM_SUPPORT**: ASPM L12 Supported. When Set this field indicates that ASPM L1.2 is supported. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky)
1	RW	0x1	**L1_1_PCIPM_SUPPORT**: PCI-PM L11 Supported. When Set this field indicates that PCI-PM L1.1 is supported, and must be Set by all Ports implementing L1 PM Substates. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky)
0	RW	0x1	**L1_2_PCIPM_SUPPORT**: PCI-PM L12 Supported. When Set this field indicates that PCI-PM L1.2 is supported. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky)

PCIE_X<*i*>_RC_PF0_L1SUB_CAP_L1SUB_CONTROL1_REG_0

where <*i*> = 1, 4, 8.

Description: This register Controls that the individual extended capability is enabled ot not.

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_L1SUB_CAP_L1SUB_CONTROL1_REG_0

Offset: 0x198
Reset: 0x00000a00 (0b0000,0000,0000,0000,0000,1010,0000,0000)

Bit	R/W	Reset	Description
31:29	RW	0x0	**L1_2_TH_SCA**: LTR L12 Threshold Scale. This field provides a scale for the value contained within the LTR_L1.2_THRESHOLD_Value. Hardware operation is undefined if software writes a Not-Permitted value to this field. Required for all Ports Ports for which the ASPM L12 Supported bit is Set, otherwise this field is of type RsvdP. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: L1SUB_CAPABILITY_REG.L1_2_ASPM_SUPPORT ? RW : RSVDP
28:26	RO	0x0	**RSVDP_26**: Reserved for future use.

Bit	R/W	Reset	Description
25:16	RW	0x0	**L1_2_TH_VAL**: LTR L12 Threshold Value. Along with the LTR_L1.2_THRESHOLD_Scale, this field indicates the LTR threshold used to determine if entry into L1 results in L1.1 (if enabled) or L1.2 (if enabled). Required for all Ports for which the ASPM L12 Supported bit is Set, otherwise this field is of type RsvdP. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: L1SUB_CAPABILITY_REG.L1_2_ASPM_SUPPORT ? RW : RSVDP
15:8	RW	0xa	**T_COMMON_MODE**: Common Mode Restore Time. Sets value of TCOMMONMODE (in us), which must be used by the Downstream Ports for timing the re-establishment of common mode. This field is of type RsvdP for Upstream Ports. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: L1SUB_CAPABILITY_REG.L1_2_ASPM_SUPPORT|| L1SUB_CAPABILITY_REG.L1_2_PCIPM_SUPPORT ? RW : RSVDP
7:4	RO	0x0	**RSVDP_4**: Reserved for future use.
3	RW	0x0	**L1_1_ASPM_EN**: ASPM L11 Enable. When Set this field, enables ASPM L1.1. For Ports for which the ASPM L1.1 Supported bit is clear, this bit is permitted to be hardwired to 0. Required for both Upstream and Downstream Ports. For compatibility with possible future extensions, software must not enable L1 PM Substates unless the L1 PM Substates Supported field in the L1 PM Substates Capabilities Register is Set.
2	RW	0x0	**L1_2_ASPM_EN**: ASPM L12 Enable. When Set this field, enables ASPM L1.2. For Ports for which the ASPM L1.2 Supported bit is clear, this bit is permitted to be hardwired to 0. Required for both Upstream and Downstream Ports. For compatibility with possible future extensions, software must not enable L1 PM Substates unless the L1 PM Substates Supported field in the L1 PM Substates Capabilities Register is Set.
1	RW	0x0	**L1_1_PCIPM_EN**: PCI-PM L11 Enable. When Set this field, enables PCI-PM L1.1. Required for both Upstream and Downstream Ports. For compatibility with possible future extensions, software must not enable L1 PM Substates unless the L1 PM Substates Supported field in the L1 PM Substates Capabilities Register is Set.
0	RW	0x0	**L1_2_PCIPM_EN**: PCI-PM L12 Enable. When Set this field, enables PCI-PM L1.2. For Ports for which the PCI-PM L1.2 Supported bit is clear, this bit is permitted to be hardwired to 0. Required for both Upstream and Downstream Ports. For compatibility with possible future extensions, software must not enable L1 PM Substates unless the L1 PM Substates Supported field in the L1 PM Substates Capabilities Register is Set.

PCIE_X4_RC_PF0_L1SUB_CAP_L1SUB_CONTROL1_REG_0

Offset: 0x1b4

PCIE_X8_RC_PF0_L1SUB_CAP_L1SUB_CONTROL1_REG_0

Offset: 0x1c8
Reset: 0x00003c00 (0b0000,0000,0000,0000,0011,1100,0000,0000)

Bit	R/W	Reset	Description
31:29	RW	0x0	**L1_2_TH_SCA**: LTR L12 Threshold Scale. This field provides a scale for the value contained within the LTR_L1.2_THRESHOLD_Value. Hardware operation is undefined if software writes a Not-Permitted value to this field. Required for all Ports Ports for which the ASPM L12 Supported bit is Set, otherwise this field is of type RsvdP. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: L1SUB_CAPABILITY_REG.L1_2_ASPM_SUPPORT ? RW : RSVDP
28:26	RO	0x0	**RSVDP_26**: Reserved for future use.
25:16	RW	0x0	**L1_2_TH_VAL**: LTR L12 Threshold Value. Along with the LTR_L1.2_THRESHOLD_Scale, this field indicates the LTR threshold used to determine if entry into L1 results in L1.1 (if enabled) or L1.2 (if enabled). Required for all Ports for which the ASPM L12 Supported bit is Set, otherwise this field is of type RsvdP. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: L1SUB_CAPABILITY_REG.L1_2_ASPM_SUPPORT ? RW : RSVDP
15:8	RW	0x3c	**T_COMMON_MODE**: Common Mode Restore Time. Sets value of TCOMMONMODE (in us), which must be used by the Downstream Ports for timing the re-establishment of common mode. This field is of type RsvdP for Upstream Ports. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: L1SUB_CAPABILITY_REG.L1_2_ASPM_SUPPORT|| L1SUB_CAPABILITY_REG.L1_2_PCIPM_SUPPORT ? RW : RSVDP
7:4	RO	0x0	**RSVDP_4**: Reserved for future use.
3	RW	0x0	**L1_1_ASPM_EN**: ASPM L11 Enable. When Set this field, enables ASPM L1.1. For Ports for which the ASPM L1.1 Supported bit is clear, this bit is permitted to be hardwired to 0. Required for both Upstream and Downstream Ports. For compatibility with possible future extensions, software must not enable L1 PM Substates unless the L1 PM Substates Supported field in the L1 PM Substates Capabilities Register is Set.
2	RW	0x0	**L1_2_ASPM_EN**: ASPM L12 Enable. When Set this field, enables ASPM L1.2. For Ports for which the ASPM L1.2 Supported bit is clear, this bit is permitted to be hardwired to 0. Required for both Upstream and Downstream Ports. For compatibility with possible future extensions, software must not enable L1 PM Substates unless the L1 PM Substates Supported field in the L1 PM Substates Capabilities Register is Set.
1	RW	0x0	**L1_1_PCIPM_EN**: PCI-PM L11 Enable. When Set this field, enables PCI-PM L1.1. Required for both Upstream and Downstream Ports. For compatibility with possible future extensions, software must not enable L1 PM Substates unless the L1 PM Substates Supported field in the L1 PM Substates Capabilities Register is Set.

Bit	R/W	Reset	Description
0	RW	0x0	**L1_2_PCIPM_EN**: PCI-PM L12 Enable. When Set this field, enables PCI-PM L1.2. For Ports for which the PCI-PM L1.2 Supported bit is clear, this bit is permitted to be hardwired to 0. Required for both Upstream and Downstream Ports. For compatibility with possible future extensions, software must not enable L1 PM Substates unless the L1 PM Substates Supported field in the L1 PM Substates Capabilities Register is Set.

PCIE_X<*i*>_RC_PF0_L1SUB_CAP_L1SUB_CONTROL2_REG_0

where <*i*> = 1, 4, 8.

Description: This register Controls that the individual extended capability is enabled ot not.

PCIE_X1_RC_PF0_L1SUB_CAP_L1SUB_CONTROL2_REG_0

Offset: 0x19c

PCIE_X4_RC_PF0_L1SUB_CAP_L1SUB_CONTROL2_REG_0

Offset: 0x1b8

PCIE_X8_RC_PF0_L1SUB_CAP_L1SUB_CONTROL2_REG_0

Offset: 0x1cc

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000028 (0b0000,0000,0000,0000,0000,0000,0010,1000)

Bit	R/W	Reset	Description
31:8	RO	0x0	**RSVDP_8**: Reserved for future use.
7:3	RW	0x5	**T_POWER_ON_VALUE**: T Power On Value. Along with the T_POWER_ON_SCALE sets the minimum amount of time (in us) that the Port must wait in L1.2.Exit after sampling CLKREQ# asserted before actively driving the interface. T_POWER_ON is calculated by multiplying the value in this field by the value in the T_POWER_ON_SCALE field. Required for all Ports that support L1.2, otherwise this field is of type RsvdP.This field must only be modified when the ASPM L1.2 Enable and PCI-PM L1.2 Enable bits are both Clear. The Port behavior is undefined if this field is modified when either the ASPM L1.2 Enable and/or PCI-PM L1.2 Enable bit(s) are set. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: L1SUB_CAPABILITY_REG.L1_2_ASPM_SUPPORT|| L1SUB_CAPABILITY_REG.L1_2_PCIPM_SUPPORT ? RWS : RSVDP

Bit	R/W	Reset	Description
2	RO	0x0	**RSVDP_2**: Reserved for future use.
1:0	RW	0x0	**T_POWER_ON_SCALE**: T Power On Scale. Specifies the scale used for T_POWER_ON_VALUE. Range of values are given below. Required for all Ports that support L1.2, otherwise this field is of type RsvdP. This field must only be modified when the ASPM L1.2 Enable and PCI-PM L1.2 Enable bits are both Clear. The Port behavior is undefined if this field is modified when either the ASPM L1.2 Enable and/or PCI-PM L1.2 Enable bit(s) are set. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: L1SUB_CAPABILITY_REG.L1_2_ASPM_SUPPORT|| L1SUB_CAPABILITY_REG.L1_2_PCIPM_SUPPORT ? RWS : RSVDP

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_RAS_DES_CAP_HEADER_REG_0

where <*i*> = 1, 4, 8.

Description: The Vendor-Specific Extended Capability (VSEC Capability) is an optional Extended Capability that is permitted to be implemented by any PCI Express Function or RCRB. This Register contains Capability Id, Capability Version and Next Offset value for Vendor-Specific Extended Capability.

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_RAS_DES_CAP_RAS_DES_CAP_HEADER_REG_0

Offset: 0x1a0
Reset: 0x2a01000b (0b0010,1010,0000,0001,0000,0000,0000,1011)

Bit	Reset	Description
31:20	0x2a0	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. Value of this field is depends on the version of the specification. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

Bit	Reset	Description
15:0	0xb	**EXTENDED_CAP_ID**: PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. Extended Capability ID for the Vendor-Specific Extended Capability is 000Bh. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X4_RC_PF0_RAS_DES_CAP_RAS_DES_CAP_HEADER_REG_0

Offset: 0x1bc
Reset: 0x2bc1000b (0b0010,1011,1100,0001,0000,0000,0000,1011)

Bit	Reset	Description
31:20	0x2bc	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. Value of this field is depends on the version of the specification. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0xb	**EXTENDED_CAP_ID**: PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. Extended Capability ID for the Vendor-Specific Extended Capability is 000Bh. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X8_RC_PF0_RAS_DES_CAP_RAS_DES_CAP_HEADER_REG_0

Offset: 0x1d0
Reset: 0x2d01000b (0b0010,1101,0000,0001,0000,0000,0000,1011)

Bit	Reset	Description
31:20	0x2d0	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

Bit	Reset	Description
19:16	0x1	**CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. Value of this field is depends on the version of the specification. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0xb	**EXTENDED_CAP_ID**: PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. Extended Capability ID for the Vendor-Specific Extended Capability is 000Bh. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_VENDOR_SPECIFIC_HEADER_REG_0

where <*i*> = 1, 4, 8.

Description: This Register field provides VSEC Length, VSEC ID and VSEC Rev(Version Number). Vendor-specific software must qualify the associated Vendor ID of the PCI Express Function or RCRB before attempting to interpret the values in the VSEC ID or VSEC Rev fields.

PCIE_X1_RC_PF0_RAS_DES_CAP_VENDOR_SPECIFIC_HEADER_REG_0

Offset: 0x1a4

PCIE_X4_RC_PF0_RAS_DES_CAP_VENDOR_SPECIFIC_HEADER_REG_0

Offset: 0x1c0

PCIE_X8_RC_PF0_RAS_DES_CAP_VENDOR_SPECIFIC_HEADER_REG_0

Offset: 0x1d4

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x10040002 (0b0001,0000,0000,0100,0000,0000,0000,0010)

Bit	Reset	Description
31:20	0x100	**VSEC_LENGTH**: VSEC Length. This field indicates the number of bytes in the entire VSEC structure, including the Vendor-Specific Extended Capability Header, the Vendor-Specific Header, and the vendor-specific registers.

Bit	Reset	Description
19:16	0x4	**VSEC_REV**: VSEC Rev. This field is a vendor-defined version number that indicates the version of the VSEC structure. Software must qualify the Vendor ID and VSEC ID before interpreting this field.
15:0	0x2	**VSEC_ID**: VSEC ID. This field is a vendor-defined ID number that indicates the nature and format of the VSEC structure. Software must qualify the Vendor ID before interpreting this field.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_EVENT_COUNTER_CONTROL_REG_0

where <*i*> = 1, 4, 8.

Description: This is a viewport control register. - Setting the EVENT_COUNTER_EVENT_SELECT and EVENT_COUNTER_LANE_SELECT fields in this register determine the Event Counter data returned by the EVENT_COUNTER_DATA_REG viewport register. - Setting the EVENT_COUNTER_ENABLE field in this register enables the Event Counter selected by the EVENT_COUNTER_EVENT_SELECT and EVENT_COUNTER_LANE_SELECT fields in this register. - Setting the EVENT_COUNTER_CLEAR field in this register clears the Event Counter selected by the EVENT_COUNTER_EVENT_SELECT and EVENT_COUNTER_LANE_SELECT fields in this register. - Reading the EVENT_COUNTER_STATUS field in this register returns the Enable status of the Event Counter selected by the EVENT_COUNTER_EVENT_SELECT and EVENT_COUNTER_LANE_SELECT fields in this register.

PCIE_X1_RC_PF0_RAS_DES_CAP_EVENT_COUNTER_CONTROL_REG_0

Offset: 0x1a8

PCIE_X4_RC_PF0_RAS_DES_CAP_EVENT_COUNTER_CONTROL_REG_0

Offset: 0x1c4

PCIE_X8_RC_PF0_RAS_DES_CAP_EVENT_COUNTER_CONTROL_REG_0

Offset: 0x1d8

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:28	RO	0x0	**RSVDP_28**: Reserved for future use.
27:16	RW	0x0	**EVENT_COUNTER_EVENT_SELECT**: Event Counter Data Select. This field in conjunction with the EVENT_COUNTER_LANE_SELECT field indexes the Event Counter data returned by the EVENT_COUNTER_DATA_REG register. - 27-24: Group number(4-bit: 0..0x7) - 23-16: Event number(8-bit: 0..0x13) within the Group - .. For detailed definitions of Group number and Event number, see the RAS DES chapter in the Databook. Note: This register field is sticky.
15:12	RO	0x0	**RSVDP_12**: Reserved for future use.
11:8	RW	0x0	**EVENT_COUNTER_LANE_SELECT**: Event Counter Lane Select. This field in conjunction with EVENT_COUNTER_EVENT_SELECT indexes the Event Counter data returned by the EVENT_COUNTER_DATA_REG register. Note: This register field is sticky.
7	RO	0x0	**EVENT_COUNTER_STATUS**: Event Counter Status. This register returns the current value of the Event Counter selected by the following fields: - EVENT_COUNTER_EVENT_SELECT - EVENT_COUNTER_LANE_SELECT Note: This register field is sticky.
6:5	RO	0x0	**RSVDP_5**: Reserved for future use.
4:2	WO	0x0	**EVENT_COUNTER_ENABLE**: Event Counter Enable. Enables/disables the Event Counter selected by the EVENT_COUNTER_EVENT_SELECT and EVENT_COUNTER_LANE_SELECT fields in this register. By default, all event counters are disabled. You can enable/disable a specific Event Counter by writing the 'per event off' or 'per event on' codes. You can enable/disable all event counters by writing the 'all on' or 'all off' codes. The read value is always '0'. For other values no change.
1:0	WO	0x0	**EVENT_COUNTER_CLEAR**: Event Counter Clear. Clears the Event Counter selected by the EVENT_COUNTER_EVENT_SELECT and EVENT_COUNTER_LANE_SELECT fields in this register. You can clear the value of a specific Event Counter by writing the 'per clear' code and you can clear all event counters at once by writing the 'all clear' code. The read value is always '0'. Other values are reserved.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_EVENT_COUNTER_DATA_REG_0

where <*i*> = 1, 4, 8.

Description: This viewport register returns the data selected by the following fields: - EVENT_COUNTER_EVENT_SELECT in EVENT_COUNTER_CONTROL_REG - EVENT_COUNTER_LANE_SELECT in EVENT_COUNTER_CONTROL_REG For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X1_RC_PF0_RAS_DES_CAP_EVENT_COUNTER_DATA_REG_0

Offset: 0x1ac

PCIE_X4_RC_PF0_RAS_DES_CAP_EVENT_COUNTER_DATA_REG_0

Offset: 0x1c8

PCIE_X8_RC_PF0_RAS_DES_CAP_EVENT_COUNTER_DATA_REG_0

Offset: 0x1dc

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EVENT_COUNTER_DATA**: Event Counter Data. This register returns the data selected by the following fields: - EVENT_COUNTER_EVENT_SELECT in EVENT_COUNTER_CONTROL_REG Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_TIME_BASED_ANALYSIS_CONTROL_REG_0

where <*i*> = 1, 4, 8.

Description: Used for controlling the measurement of RX/TX data throughput and time spent in each low-power LTSSM state. For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X1_RC_PF0_RAS_DES_CAP_TIME_BASED_ANALYSIS_CONTROL_REG_0

Offset: 0x1b0

PCIE_X4_RC_PF0_RAS_DES_CAP_TIME_BASED_ANALYSIS_CONTROL_REG_0

Offset: 0x1cc

PCIE_X8_RC_PF0_RAS_DES_CAP_TIME_BASED_ANALYSIS_CONTROL_REG_0

Offset: 0x1e0

Read/Write: See table below
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000100 (0b0000,0000,0000,0000,0000,0001,0000,0000)

Bit	R/W	Reset	Description
31:24	RW	0x0	**TIME_BASED_REPORT_SELECT**: Time-based Report Select. Selects what type of data is measured for the selected duration (TIME_BASED_DURATION_SELECT), and returned in TIME_BASED_ANALYSIS_DATA. Each type of data is measured using one of three types of units: - Core_clk Cycles for 2.5GT/s, 5.0GT/s, 8.0GT/s, 16.0GT/s, 32GT/s. Total time in ps is [Value of TIME_BASED_ANALYSIS_DATA returned when TIME_BASED_REPORT_SELECT=0x00] * TIME_BASED_ANALYSIS_DATA. Values 0-4 and 7-8 correspond to Core_clk Cycles for 2.5GT/s, 5.0GT/s, 8.0GT/s, 16.0GT/s, 32GT/s. - Aux_clk Cycles. Total time in ps is [Period of platform specific clock] * TIME_BASED_ANALYSIS_DATA. Values 5, 6, and 9 correspond to aux_clk Cycles. - Core_clk Cycles for 20GT/s, 25GT/s (CCIX ESM data rate). Total time in ps is [Value of TIME_BASED_ANALYSIS_DATA returned when TIME_BASED_REPORT_SELECT=0x10] * TIME_BASED_ANALYSIS_DATA. Values 10-14 and 17-18 correspond to Core_clk Cycles for 2.5GT/s, 5.0GT/s, 8.0GT/s, 16.0GT/s, 32GT/s. - Data Bytes. Actual amount of bytes is 16 * TIME_BASED_ANALYSIS_DATA. Values 20-23 correspond to data bytes. All encodings other than the defined encodings are reserved. Note: This register field is sticky.
23:16	RO	0x0	**RSVDP_16**: Reserved for future use.
15:8	RW	0x1	**TIME_BASED_DURATION_SELECT**: Time-based Duration Select. Selects the duration of time-based analysis. When "manual control" is selected and TIMER_START is set to '1', this analysis never stops until TIMER_STOP is set to '0'. All encodings other than the defined encodings are reserved. Note: This register field is sticky.
7:1	RO	0x0	**RSVDP_1**: Reserved for future use.
0	RW	0x0	**TIMER_START**: Timer Start. This bit will be cleared automatically when the measurement is finished. Note: The app_ras_des_tba_ctrl input also sets the contents of this field and controls the measurement start/stop. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_TIME_BASED_ANALYSIS_DATA_REG_0

where <*i*> = 1, 4, 8.

Description: Contains the measurement results of RX/TX data throughput and time spent in each low-power LTSSM state. This viewport register returns the data selected by the TIME_BASED_REPORT_SELECT field in TIME_BASED_ANALYSIS_CONTROL_REG. For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X1_RC_PF0_RAS_DES_CAP_TIME_BASED_ANALYSIS_DATA_REG_0

Offset: 0x1b4

PCIE_X4_RC_PF0_RAS_DES_CAP_TIME_BASED_ANALYSIS_DATA_REG_0

Offset: 0x1d0

PCIE_X8_RC_PF0_RAS_DES_CAP_TIME_BASED_ANALYSIS_DATA_REG_0

Offset: 0x1e4

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TIME_BASED_ANALYSIS_DATA**: Time Based Analysis Data. This register returns the data selected by the TIME_BASED_REPORT_SELECT field in TIME_BASED_ANALYSIS_CONTROL_REG. The results are cleared when next measurement starts. Note: This register field is sticky.

PCIE_X1_RC_PF0_RAS_DES_CAP_TIME_BASED_ANALYSIS_DATA_63_32_REG_0

where <*i*> = 1, 4, 8.

Description: This viewport register returns the data selected by the TIME_BASED_REPORT_SELECT field in TIME_BASED_ANALYSIS_CONTROL_REG. For more details, see the "Reliability, Availability, and Serviceability (RAS)" section in the "Controller Operations" chapter of the Databook.

PCIE_X1_RC_PF0_RAS_DES_CAP_TIME_BASED_ANALYSIS_DATA_63_32_REG_0

Offset: 0x1b8

PCIE_X4_RC_PF0_RAS_DES_CAP_TIME_BASED_ANALYSIS_DATA_63_32_REG_0

Offset: 0x1d4

PCIE_X8_RC_PF0_RAS_DES_CAP_TIME_BASED_ANALYSIS_DATA_63_32_REG_0

Offset: 0x1e8

Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TIME_BASED_ANALYSIS_DATA_63_32**: Upper 32 bits of Time Based Analysis Data. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_EINJ_ENABLE_REG_0

where <*i*> = 1, 4, 8.

Description: Each type of error insertion is enabled by the corresponding bit in this register. The specific injection controls for each type of error are defined in the following registers: - 0: CRC Error: EINJ0_CRC_REG - 1: Sequence Number Error: EINJ1_SEQNUM_REG - 2: DLLP Error: EINJ2_DLLP_REG - 3: Symbol DataK Mask Error or Sync Header Error: EINJ3_SYMBOL_REG - 4: FC Credit Update Error: EINJ4_FC_REG - 5: TLP Duplicate/Nullify Error: EINJ5_SP_TLP_REG - 6: Specific TLP Error: EINJ6_COMPARE_*_REG/EINJ6_CHANGE_*_REG/EINJ6_TLP_REG After the errors have been inserted by controller, it will clear each bit here. For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X1_RC_PF0_RAS_DES_CAP_EINJ_ENABLE_REG_0

Offset: 0x1d0

PCIE_X4_RC_PF0_RAS_DES_CAP_EINJ_ENABLE_REG_0

Offset: 0x1ec

PCIE_X8_RC_PF0_RAS_DES_CAP_EINJ_ENABLE_REG_0

Offset: 0x200

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:7	RO	0x0	**RSVDP_7**: Reserved for future use.

Bit	R/W	Reset	Description
6	RW	0x0	**ERROR_INJECTION6_ENABLE**: Error Injection6 Enable (Specific TLP Error). Enables insertion of errors into the packets that you select. You can set this bit to '1' when you have disabled RAS datapath protection (DP) by setting CX_RASDP = CX_RASDP_RAM_PROT =0. You can set this bit to '1' when you have disabled the address translation by setting ADDR_TRANSLATION_SUPPORT_EN=0. For more details, see the EINJ6_COMPARE_*_REG/EINJ6_CHANGE_*_REG/EINJ6_TLP_REG registers. Note: This register field is sticky.
5	RW	0x0	**ERROR_INJECTION5_ENABLE**: Error Injection5 Enable (TLP Duplicate/Nullify Error). Enables insertion of duplicate/nullified TLPs. For more details, see the EINJ5_SP_TLP_REG register. Note: This register field is sticky.
4	RW	0x0	**ERROR_INJECTION4_ENABLE**: Error Injection4 Enable (FC Credit Update Error). Enables insertion of errors into UpdateFCs. For more details, see the EINJ4_FC_REG register. Note: This register field is sticky.
3	RW	0x0	**ERROR_INJECTION3_ENABLE**: Error Injection3 Enable (Symbol DataK Mask Error or Sync Header Error). Enables DataK masking of special symbols or the breaking of the sync header. For more details, see the EINJ3_SYMBOL_REG register. Note: This register field is sticky.
2	RW	0x0	**ERROR_INJECTION2_ENABLE**: Error Injection2 Enable (DLLP Error). Enables insertion of DLLP errors. For more details, see the EINJ2_DLLP_REG register. Note: This register field is sticky.
1	RW	0x0	**ERROR_INJECTION1_ENABLE**: Error Injection1 Enable (Sequence Number Error). Enables insertion of errors into sequence numbers. For more details, see the EINJ1_SEQNUM_REG register. Note: This register field is sticky.
0	RW	0x0	**ERROR_INJECTION0_ENABLE**: Error Injection0 Enable (CRC Error). Enables insertion of errors into various CRC. For more details, see the EINJ0_CRC_REG register. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_EINJ0_CRC_REG_0

where <*i*> = 1, 4, 8.

Description: Controls the insertion of errors into the CRC, and parity of ordered sets for the selected type of the packets as follows: - LCRC. Bad TLP will be detected at the receiver side; receiver responds with NAK DLLP; Data Link Retry starts. - 16-bit CRC of ACK/NAK DLLPs. Bad DLLP occurs at the receiver side; Replay NUM Rollover occurs. - 16-bit CRC of UpdateFC DLLPs. Error insertion continues for the specific time; LTSSM transitions to the Recovery state because of the UpdateFC timeout (if the timeout is implemented at the receiver of the UpdateFCs). - ECRC. If ECRC check is enabled, ECRC error is detected at the receiver side. - FCRC. Framing error will be detected, TLP is discarded, and the LTSSM transitions to Recovery state. - Parity of TSOS. Error

insertion continues for the specific time; LTSSM Recovery/Configuration timeout will occur. - Parity of SKPOS. Lane error will be detected at the receiver side.

PCIE_X1_RC_PF0_RAS_DES_CAP_EINJ0_CRC_REG_0

Offset: 0x1d4

PCIE_X4_RC_PF0_RAS_DES_CAP_EINJ0_CRC_REG_0

Offset: 0x1f0

PCIE_X8_RC_PF0_RAS_DES_CAP_EINJ0_CRC_REG_0

Offset: 0x204

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:12	RO	0x0	**RSVDP_12**: Reserved for future use.
11:8	RW	0x0	**EINJ0_CRC_TYPE**: Error injection type. Selects the type of CRC error to be inserted. All encodings other than the defined encodings are reserved. Note: This register field is sticky.
7:0	RW	0x0	**EINJ0_COUNT**: Error injection count. Indicates the number of errors. This register is decremented when the errors have been inserted. - If the counter value is 0x01 and error is inserted, ERROR_INJECTION0_ENABLE in EINJ_ENABLE_REG returns 0b. - If the counter value is 0x00 and ERROR_INJECTION0_ENABLE=1, the errors are inserted until ERROR_INJECTION0_ENABLE is set to '0'. Note: This register field is sticky.

PCIE_X1_RC_PF0_RAS_DES_CAP_EINJ1_SEQNUM_REG_0

where <*i*> = 1, 4, 8.

Description: Controls the sequence number of the specific TLPs and ACK/NAK DLLPs. Data Link Protocol Error will be detected at the Rx side of ACK/NAL DLLPs when one of these conditions is true: - ((NEXT_TRANSMIT_SEQ -1) - AckNak_Seq_Num) mod 4096 > 2048 - (AckNak_Seq_Num - ACKD_SEQ) mod 4096 >= 2048 TLP is treated as Duplicate TLP at the Rx side when all these conditions are true: - Sequence Number != NEXT_RCV_SEQ - (NEXT_RCV_SEQ - Sequence Number) mod 4096 <= 2048 TLP is treated as Bad TLP at the Rx side when all these conditions are true: -

Sequence Number != NEXT_RCV_SEQ and - (NEXT_RCV_SEQ - Sequence Number) mod 4096 > 2048).

PCIE_X1_RC_PF0_RAS_DES_CAP_EINJ1_SEQNUM_REG_0

Offset: 0x1d8

PCIE_X4_RC_PF0_RAS_DES_CAP_EINJ1_SEQNUM_REG_0

Offset: 0x1f4

PCIE_X8_RC_PF0_RAS_DES_CAP_EINJ1_SEQNUM_REG_0

Offset: 0x208

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:29	RO	0x0	**RSVDP_29**: Reserved for future use.
28:16	RW	0x0	**EINJ1_BAD_SEQNUM**: Bad sequence number. Indicates the value to add/subtract from the naturally-assigned sequence numbers. This value is represented by two's complement. For example: - Set Type, SEQ# and Count -- EINJ1_SEQNUM_TYPE =0 (Insert errors into new TLPs) -- EINJ1_BAD_SEQNUM =0x1FFD (represents -3) -- EINJ1_COUNT =1 - Enable Error Injection -- ERROR_INJECTION1_ENABLE =1 - Send a TLP From the Core's Application Interface -- Assume SEQ#5 is given to the TLP. - The SEQ# is Changed to #2 by the Error Injection Function in Layer2. -- 5 + (-3) = 2 - The TLP with SEQ#2 is Transmitted to PCIe Link. Note: This register field is sticky.
15:9	RO	0x0	**RSVDP_9**: Reserved for future use.
8	RW	0x0	**EINJ1_SEQNUM_TYPE**: Sequence number type. Selects the type of sequence number. Note: This register field is sticky.
7:0	RW	0x0	**EINJ1_COUNT**: Error injection count. Indicates the number of errors. This register is decremented as the errors are being inserted. - If the counter value is 0x01 and error is inserted, ERROR_INJECTION1_ENABLE in EINJ_ENABLE_REG returns '0'. - If the counter value is 0x00 and ERROR_INJECTION1_ENABLE=1, the errors are inserted until ERROR_INJECTION1_ENABLE is set to '0'. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_EINJ2_DLLP_REG_0

where <*i*> = 1, 4, 8.

Description: Controls the transmission of DLLPs and inserts the following errors: - If "ACK/NAK DLLP's transmission block" is selected, replay timeout error will occur at the transmitter of the TLPs and then Data Link Retry will occur. - If "Update FC DLLP's transmission block" is selected, LTSSM will transition to the Recovery state because of the UpdateFC timeout (if the timeout is implemented at the receiver of the UpdateFCs). - If "Always Transmission for NAK DLLP" is selected, Data Link Retry will occur at the transmitter of the TLPs. Furthermore, Replay NUM Rollover will occur when the transmitter has been requested four times to send the TLP with the same sequence number.

PCIE_X1_RC_PF0_RAS_DES_CAP_EINJ2_DLLP_REG_0

Offset: 0x1dc

PCIE_X4_RC_PF0_RAS_DES_CAP_EINJ2_DLLP_REG_0

Offset: 0x1f8

PCIE_X8_RC_PF0_RAS_DES_CAP_EINJ2_DLLP_REG_0

Offset: 0x20c

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:10	RO	0x0	**RSVDP_10**: Reserved for future use.
9:8	RW	0x0	**EINJ2_DLLP_TYPE**: DLLP Type. Selects the type of DLLP errors to be inserted. Note: This register field is sticky.
7:0	RW	0x0	**EINJ2_COUNT**: Error injection count. Indicates the number of errors. This register is decremented as the errors are being inserted. - If the counter value is 0x01 and the error is inserted, ERROR_INJECTION2_ENABLE in EINJ_ENABLE_REG returns '0'. - If the counter value is 0x00 and ERROR_INJECTION2_ENABLE =1, the errors are inserted until ERROR_INJECTION2_ENABLE is set to '0'. This register is affected only when EINJ2_DLLP_TYPE =2'10b. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_EINJ3_SYMBOL_REG_0

where <*i*> = 1, 4, 8.

Description: When 8b/10b encoding is used, this register controls error insertion into the special (K code) symbols. - If TS1/TS2/FTS/E-Idle/SKP is selected, it affects whole of the ordered set. It might cause timeout of the LTSSM. - If END/EDB/STP/SDP is selected, TLP/DLLP will be discarded at the receiver side. When 128b/130b encoding is used, this register controls error insertion into the sync-header.

PCIE_X1_RC_PF0_RAS_DES_CAP_EINJ3_SYMBOL_REG_0

Offset: 0x1e0

PCIE_X4_RC_PF0_RAS_DES_CAP_EINJ3_SYMBOL_REG_0

Offset: 0x1fc

PCIE_X8_RC_PF0_RAS_DES_CAP_EINJ3_SYMBOL_REG_0

Offset: 0x210

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:11	RO	0x0	**RSVDP_11**: Reserved for future use.
10:8	RW	0x0	**EINJ3_SYMBOL_TYPE**: Error Type. 8b/10b encoding - Mask K symbol. It is not supported to insert errors into the first ordered-set after exiting from TxElecIdle when CX_FREQ_STEP_EN has been enabled. All encodings other than the defined encodings are reserved. Note: This register field is sticky.
7:0	RW	0x0	**EINJ3_COUNT**: Error injection count. Indicates the number of errors. This register is decremented as the errors are being inserted. - If the counter value is 0x01 and error is inserted, ERROR_INJECTION3_ENABLE in EINJ_ENABLE_REG returns '0'. - If the counter value is 0x00 and ERROR_INJECTION3_ENABLE =1, the errors are inserted until ERROR_INJECTION3_ENABLE is set to '0'. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_EINJ4_FC_REG_0

where <*i*> = 1, 4, 8.

Description: Controls error insertion into the credit value in the UpdateFCs. It is possible to insert errors for any of the following types: - Posted TLP Header credit - Non-Posted TLP Header credit - Completion TLP Header credit - Posted TLP Data credit - Non-Posted TLP Data credit - Completion TLP Data credit These errors are not correctable while error insertion is enabled. Receiver buffer overflow error might occur.

PCIE_X1_RC_PF0_RAS_DES_CAP_EINJ4_FC_REG_0

Offset: 0x1e4

PCIE_X4_RC_PF0_RAS_DES_CAP_EINJ4_FC_REG_0

Offset: 0x200

PCIE_X8_RC_PF0_RAS_DES_CAP_EINJ4_FC_REG_0

Offset: 0x214

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:29	RO	0x0	**RSVDP_29**: Reserved for future use.
28:16	RW	0x0	**EINJ4_BAD_UPDFC_VALUE**: Bad update-FC credit value. Indicates the value to add/subtract from the UpdateFC credit. This value is represented by two's complement. Note: This register field is sticky.
15	RO	0x0	**RSVDP_15**: Reserved for future use.
14:12	RW	0x0	**EINJ4_VC_NUMBER**: VC Number. Indicates target VC Number. Note: This register field is sticky.
11	RO	0x0	**RSVDP_11**: Reserved for future use.
10:8	RW	0x0	**EINJ4_UPDFC_TYPE**: Update-FC type. Selects the credit type. Note: This register field is sticky.

Bit	R/W	Reset	Description
7:0	RW	0x0	**EINJ4_COUNT**: Error injection count. Indicates the number of errors. This register is decremented as the errors are being inserted. - If the counter value is 0x01 and error is inserted, ERROR_INJECTION4_ENABLE in EINJ_ENABLE_REG returns '0'. - If the counter value is 0x00 and ERROR_INJECTION4_ENABLE =1, the errors are inserted until ERROR_INJECTION4_ENABLE is set to '0'. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_EINJ5_SP_TLP_REG_0

where <*i*> = 1, 4, 8.

Description: Controls the generation of specified TLPs. Correctable errors will occur which will be fixed by the PCIe protocol. - For Duplicate TLP, the controller initiates Data Link Retry by handling ACK DLLP as NAK DLLP. These TLPs will be duplicate TLPs at the receiver side. - For Nullified TLP, the TLPs that the controller transmits are changed into nullified TLPs and the original TLPs are stored in the retry buffer. The receiver of these TLPs will detect the lack of seq# and send NAK DLLP at the next TLP. Then the original TLPs are sent from retry buffer and the data controls are recovered. For 128 bit controller or more than 128 bit, the controller inserts errors the number of times of EINJ5_COUNT but doesn't ensure that the errors are continuously inserted into TLPs.

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

PCIE_X1_RC_PF0_RAS_DES_CAP_EINJ5_SP_TLP_REG_0

Offset: 0x1e8

PCIE_X4_RC_PF0_RAS_DES_CAP_EINJ5_SP_TLP_REG_0

Offset: 0x204

PCIE_X8_RC_PF0_RAS_DES_CAP_EINJ5_SP_TLP_REG_0

Offset: 0x218

Bit	R/W	Reset	Description
31:9	RO	0x0	**RSVDP_9**: Reserved for future use.

Bit	R/W	Reset	Description
8	RW	0x0	**EINJ5_SPECIFIED_TLP**: Specified TLP. Selects the specified TLP to be inserted. Note: This register field is sticky.
7:0	RW	0x0	**EINJ5_COUNT**: Error injection count. Indicates the number of errors. This register is decremented as the errors are being inserted. - If the counter value is 0x01 and error is inserted, ERROR_INJECTION5_ENABLE in EINJ_ENABLE_REG returns '0'. - If the counter value is 0x00 and ERROR_INJECTION5_ENABLE =1, the errors are inserted until ERROR_INJECTION5_ENABLE is set to '0'. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_POINT_H0_REG_0

where <*i*> = 1, 4, 8.

Description: Program this register for the 1st DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW0[7:0], TLP_DW0[15:8], TLP_DW0[23:16], TLP_DW0[31:24] The Packet Compare Point registers (EINJ6_COMPARE_POINT*) specify which Tx TLP header bits to compare with the corresponding bits in the Packet Compare Value registers (EINJ6_COMPARE_VALUE*). When all specified bits (in the Tx TLP header and EINJ6_COMPARE_VALUE*) match, the controller inserts errors into the TLP. The type and number of errors are specified by the EINJ6_TLP_REG register.

PCIE_X1_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_POINT_H0_REG_0

Offset: 0x1ec

PCIE_X4_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_POINT_H0_REG_0

Offset: 0x208

PCIE_X8_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_POINT_H0_REG_0

Offset: 0x21c

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_COMPARE_POINT_H0**: Packet Compare Point: 1st DWORD. Specifies which Tx TLP header DWORD#0 bits to compare with the corresponding bits in the Packet Compare Value registers (EINJ6_COMPARE_VALUE*). When all specified bits (in the Tx TLP header and EINJ6_COMPARE_VALUE*) match, the controller inserts errors into the TLP. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_POINT_H1_REG_0

where <*i*> = 1, 4, 8.

Description: Program this register for the 2nd DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW1[7:0], TLP_DW1[15:8], TLP_DW1[23:16], TLP_DW1[31:24] The Packet Compare Point registers (EINJ6_COMPARE_POINT*) specify which Tx TLP header bits to compare with the corresponding bits in the Packet Compare Value registers (EINJ6_COMPARE_VALUE*). When all specified bits (in the Tx TLP header and EINJ6_COMPARE_VALUE*) match, the controller inserts errors into the TLP. The type and number of errors are specified by the EINJ6_TLP_REG register.

PCIE_X1_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_POINT_H1_REG_0

Offset: 0x1f0

PCIE_X4_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_POINT_H1_REG_0

Offset: 0x20c

PCIE_X8_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_POINT_H1_REG_0

Offset: 0x220

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_COMPARE_POINT_H1**: Packet Compare Point: 2nd DWORD. Specifies which Tx TLP header DWORD#1 bits to compare with the corresponding bits in the Packet Compare Value registers (EINJ6_COMPARE_VALUE*). When all specified bits (in the Tx TLP header and EINJ6_COMPARE_VALUE*) match, the controller inserts errors into the TLP. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_POINT_H2_REG_0

where <*i*> = 1, 4, 8.

Description: Program this register for the 3rd DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW2[7:0], TLP_DW2[15:8], TLP_DW2[23:16], TLP_DW2[31:24] The Packet Compare Point registers (EINJ6_COMPARE_POINT*) specify which Tx TLP header bits to compare with the corresponding bits in the Packet Compare Value registers (EINJ6_COMPARE_VALUE*). When all specified bits (in the Tx TLP header and EINJ6_COMPARE_VALUE*) match, the controller inserts errors into the TLP. The type and number of errors are specified by the EINJ6_TLP_REG register.

PCIE_X1_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_POINT_H2_REG_0

Offset: 0x1f4

PCIE_X4_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_POINT_H2_REG_0

Offset: 0x210

PCIE_X8_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_POINT_H2_REG_0

Offset: 0x224

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_COMPARE_POINT_H2**: Packet Compare Point: 3rd DWORD. Specifies which Tx TLP header DWORD#2 bits to compare with the corresponding bits in the Packet Compare Value registers (EINJ6_COMPARE_VALUE*). When all specified bits (in the Tx TLP header and EINJ6_COMPARE_VALUE*) match, the controller inserts errors into the TLP. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_POINT_H3_REG_0

where <*i*> = 1, 4, 8.

Description: Program this register for the 4th DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW3[7:0], TLP_DW3[15:8], TLP_DW3[23:16], TLP_DW3[31:24] The Packet Compare Point registers

(EINJ6_COMPARE_POINT*) specify which Tx TLP header bits to compare with the corresponding bits in the Packet Compare Value registers (EINJ6_COMPARE_VALUE*). When all specified bits (in the Tx TLP header and EINJ6_COMPARE_VALUE*) match, the controller inserts errors into the TLP. The type and number of errors are specified by the EINJ6_TLP_REG register.

PCIE_X1_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_POINT_H3_REG_0

Offset: 0x1f8

PCIE_X4_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_POINT_H3_REG_0

Offset: 0x214

PCIE_X8_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_POINT_H3_REG_0

Offset: 0x228

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_COMPARE_POINT_H3**: Packet Compare Point: 4th DWORD. Specifies which Tx TLP header DWORD#3 bits to compare with the corresponding bits in the Packet Compare Value registers (EINJ6_COMPARE_VALUE*). When all specified bits (in the Tx TLP header and EINJ6_COMPARE_VALUE*) match, the controller inserts errors into the TLP. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_VALUE_H0_REG_0

where <*i*> = 1, 4, 8.

Description: Program this register for the 1st DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW0[7:0], TLP_DW0[15:8], TLP_DW0[23:16], TLP_DW0[31:24] The Packet Compare Point registers (EINJ6_COMPARE_POINT*) specify which Tx TLP header bits to compare with the corresponding bits in the Packet Compare Value registers (EINJ6_COMPARE_VALUE*). When all specified bits (in the Tx TLP header and EINJ6_COMPARE_VALUE*) match, the controller inserts errors into the TLP. The type and number of errors are specified by the EINJ6_TLP_REG register.

PCIE_X1_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_VALUE_H0_REG_0

Offset: 0x1fc

PCIE_X4_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_VALUE_H0_REG_0

Offset: 0x218

PCIE_X8_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_VALUE_H0_REG_0

Offset: 0x22c

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_COMPARE_VALUE_H0**: Packet Compare Value: 1st DWORD. Specifies the value to compare against Tx the TLP header DWORD#0 bits specified in the Packet Compare Point registers (EINJ6_COMPARE_POINT*). Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_VALUE_H1_REG_0

where <*i*> = 1, 4, 8.

Description: Program this register for the 2nd DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW1[7:0], TLP_DW1[15:8], TLP_DW1[23:16], TLP_DW1[31:24] The Packet Compare Point registers (EINJ6_COMPARE_POINT*) specify which Tx TLP header bits to compare with the corresponding bits in the Packet Compare Value registers (EINJ6_COMPARE_VALUE*). When all specified bits (in the Tx TLP header and EINJ6_COMPARE_VALUE*) match, the controller inserts errors into the TLP. The type and number of errors are specified by the EINJ6_TLP_REG register.

PCIE_X1_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_VALUE_H1_REG_0

Offset: 0x200

PCIE_X4_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_VALUE_H1_REG_0

Offset: 0x21c

PCIE_X8_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_VALUE_H1_REG_0

Offset: 0x230

Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_COMPARE_VALUE_H1**: Packet Compare Value: 2nd DWORD. Specifies the value to compare against Tx the TLP header DWORD#1 bits specified in the Packet Compare Point registers (EINJ6_COMPARE_POINT*). Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_VALUE_H2_REG_0

where <*i*> = 1, 4, 8.

Description: Program this register for the 3rd DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW2[7:0], TLP_DW2[15:8], TLP_DW2[23:16], TLP_DW2[31:24] The Packet Compare Point registers (EINJ6_COMPARE_POINT*) specify which Tx TLP header bits to compare with the corresponding bits in the Packet Compare Value registers (EINJ6_COMPARE_VALUE*). When all specified bits (in the Tx TLP header and EINJ6_COMPARE_VALUE*) match, the controller inserts errors into the TLP. The type and number of errors are specified by the EINJ6_TLP_REG register.

PCIE_X1_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_VALUE_H2_REG_0

Offset: 0x204

PCIE_X4_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_VALUE_H2_REG_0

Offset: 0x220

PCIE_X8_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_VALUE_H2_REG_0

Offset: 0x234

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_COMPARE_VALUE_H2**: Packet Compare Value: 3rd DWORD. Specifies the value to compare against Tx the TLP header DWORD#2 bits specified in the Packet Compare Point registers (EINJ6_COMPARE_POINT*). Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_VALUE_H3_REG_0

where <*i*> = 1, 4, 8.

Description: Program this register for the 4th DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW3[7:0], TLP_DW3[15:8], TLP_DW3[23:16], TLP_DW3[31:24] The Packet Compare Point registers (EINJ6_COMPARE_POINT*) specify which Tx TLP header bits to compare with the corresponding bits in the Packet Compare Value registers (EINJ6_COMPARE_VALUE*). When all specified bits (in the Tx TLP header and EINJ6_COMPARE_VALUE*) match, the controller inserts errors into the TLP. The type and number of errors are specified by the EINJ6_TLP_REG register.

PCIE_X1_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_VALUE_H3_REG_0

Offset: 0x208

PCIE_X4_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_VALUE_H3_REG_0

Offset: 0x224

PCIE_X8_RC_PF0_RAS_DES_CAP_EINJ6_COMPARE_VALUE_H3_REG_0

Offset: 0x238

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_COMPARE_VALUE_H3**: Packet Compare Value: 4th DWORD. Specifies the value to compare against Tx the TLP header DWORD#3 bits specified in the Packet Compare Point registers (EINJ6_COMPARE_POINT*). Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_POINT_H0_REG_0

where <*i*> = 1, 4, 8.

Description: Program this register for the 1st DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW0[7:0], TLP_DW0[15:8], TLP_DW0[23:16], TLP_DW0[31:24] The Packet Change Point registers (EINJ6_CHANGE_POINT*) specify which Tx TLP header bits to replace with the corresponding bits

in the Packet Change Value registers (EINJ6_CHANGE_VALUE*). The type and number of errors are specified by the EINJ6_TLP_REG register. Only applies when EINJ6_INVERTED_CONTROL in EINJ6_TLP_REG =0.

PCIE_X1_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_POINT_H0_REG_0

Offset: 0x20c

PCIE_X4_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_POINT_H0_REG_0

Offset: 0x228

PCIE_X8_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_POINT_H0_REG_0

Offset: 0x23c

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_CHANGE_POINT_H0**: Packet Change Point: 1st DWORD. Specifies which Tx TLP header DWORD#0 bits to replace with the corresponding bits in the Packet Change Value registers (EINJ6_CHANGE_VALUE*). Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_POINT_H1_REG_0

where <*i*> = 1, 4, 8.

Description: Program this register for the 2nd DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW1[7:0], TLP_DW1[15:8], TLP_DW1[23:16], TLP_DW1[31:24] The Packet Change Point registers (EINJ6_CHANGE_POINT*) specify which Tx TLP header bits to replace with the corresponding bits in the Packet Change Value registers (EINJ6_CHANGE_VALUE*). The type and number of errors are specified by the EINJ6_TLP_REG register. Only applies when EINJ6_INVERTED_CONTROL in EINJ6_TLP_REG =0.

PCIE_X1_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_POINT_H1_REG_0

Offset: 0x210

PCIE_X4_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_POINT_H1_REG_0

Offset: 0x22c

PCIE_X8_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_POINT_H1_REG_0

Offset: 0x240

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_CHANGE_POINT_H1**: Packet Change Point: 2nd DWORD. Specifies which Tx TLP header DWORD#1 bits to replace with the corresponding bits in the Packet Change Value registers (EINJ6_CHANGE_VALUE*). Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_POINT_H2_REG_0

where <*i*> = 1, 4, 8.

Description: Program this register for the 3rd DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW2[7:0], TLP_DW2[15:8], TLP_DW2[23:16], TLP_DW2[31:24] The Packet Change Point registers (EINJ6_CHANGE_POINT*) specify which Tx TLP header bits to replace with the corresponding bits in the Packet Change Value registers (EINJ6_CHANGE_VALUE*). The type and number of errors are specified by the EINJ6_TLP_REG register. Only applies when EINJ6_INVERTED_CONTROL in EINJ6_TLP_REG =0.

PCIE_X1_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_POINT_H2_REG_0

Offset: 0x214

PCIE_X4_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_POINT_H2_REG_0

Offset: 0x230

PCIE_X8_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_POINT_H2_REG_0

Offset: 0x244

Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_CHANGE_POINT_H2**: Packet Change Point: 3rd DWORD. Specifies which Tx TLP header DWORD#2 bits to replace with the corresponding bits in the Packet Change Value registers (EINJ6_CHANGE_VALUE*). Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_POINT_H3_REG_0

where <*i*> = 1, 4, 8.

Description: Program this register for the 4th DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW3[7:0], TLP_DW3[15:8], TLP_DW3[23:16], TLP_DW3[31:24] The Packet Change Point registers (EINJ6_CHANGE_POINT*) specify which Tx TLP header bits to replace with the corresponding bits in the Packet Change Value registers (EINJ6_CHANGE_VALUE*). The type and number of errors are specified by the EINJ6_TLP_REG register. Only applies when EINJ6_INVERTED_CONTROL in EINJ6_TLP_REG =0.

PCIE_X1_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_POINT_H3_REG_0

Offset: 0x218

PCIE_X4_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_POINT_H3_REG_0

Offset: 0x234

PCIE_X8_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_POINT_H3_REG_0

Offset: 0x248

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_CHANGE_POINT_H3**: Packet Change Point: 4th DWORD. Specifies which Tx TLP header DWORD#3 bits to replace with the corresponding bits in the Packet Change Value registers (EINJ6_CHANGE_VALUE*). Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_VALUE_H0_REG_0

where <*i*> = 1, 4, 8.

Description: Program this register for the 1st DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW0[7:0], TLP_DW0[15:8], TLP_DW0[23:16], TLP_DW0[31:24] The Packet Change Point registers (EINJ6_CHANGE_POINT*) specify which Tx TLP header bits to replace with the corresponding bits in the Packet Change Value registers (EINJ6_CHANGE_VALUE*). The type and number of errors are specified by the EINJ6_TLP_REG register. Only applies when EINJ6_INVERTED_CONTROL in EINJ6_TLP_REG =0.

PCIE_X1_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_VALUE_H0_REG_0

Offset: 0x21c

PCIE_X4_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_VALUE_H0_REG_0

Offset: 0x238

PCIE_X8_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_VALUE_H0_REG_0

Offset: 0x24c

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_CHANGE_VALUE_H0**: Packet Change Value: 1st DWORD. Specifies replacement values for the Tx TLP header DWORD#0 bits defined in the Packet Change Point registers (EINJ6_CHANGE_POINT*). Only applies when the EINJ6_INVERTED_CONTROL field in EINJ6_TLP_REG is '0'. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_VALUE_H1_REG_0

where <*i*> = 1, 4, 8.

Description: Program this register for the 2nd DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW1[7:0], TLP_DW1[15:8], TLP_DW1[23:16], TLP_DW1[31:24] The Packet Change Point registers

(EINJ6_CHANGE_POINT*) specify which Tx TLP header bits to replace with the corresponding bits in the Packet Change Value registers (EINJ6_CHANGE_VALUE*). The type and number of errors are specified by the EINJ6_TLP_REG register. Only applies when EINJ6_INVERTED_CONTROL in EINJ6_TLP_REG =0.

PCIE_X1_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_VALUE_H1_REG_0

Offset: 0x220

PCIE_X4_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_VALUE_H1_REG_0

Offset: 0x23c

PCIE_X8_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_VALUE_H1_REG_0

Offset: 0x250

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_CHANGE_VALUE_H1**: Packet Change Value: 2nd DWORD. Specifies replacement values for the Tx TLP header DWORD#1 bits defined in the Packet Change Point registers (EINJ6_CHANGE_POINT*). Only applies when the EINJ6_INVERTED_CONTROL field in EINJ6_TLP_REG is '0'. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_VALUE_H2_REG_0

where <*i*> = 1, 4, 8.

Description: Program this register for the 3rd DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW2[7:0], TLP_DW2[15:8], TLP_DW2[23:16], TLP_DW2[31:24] The Packet Change Point registers (EINJ6_CHANGE_POINT*) specify which Tx TLP header bits to replace with the corresponding bits in the Packet Change Value registers (EINJ6_CHANGE_VALUE*). The type and number of errors are specified by the EINJ6_TLP_REG register. Only applies when EINJ6_INVERTED_CONTROL in EINJ6_TLP_REG =0.

PCIE_X1_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_VALUE_H2_REG_0

Offset: 0x224

PCIE_X4_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_VALUE_H2_REG_0

Offset: 0x240

PCIE_X8_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_VALUE_H2_REG_0

Offset: 0x254

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_CHANGE_VALUE_H2**: Packet Change Value: 3rd DWORD. Specifies replacement values for the Tx TLP header DWORD#2 bits defined in the Packet Change Point registers (EINJ6_CHANGE_POINT*). Only applies when the EINJ6_INVERTED_CONTROL field in EINJ6_TLP_REG is '0'. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_VALUE_H3_REG_0

where <*i*> = 1, 4, 8.

Description: Program this register for the 4th DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW3[7:0], TLP_DW3[15:8], TLP_DW3[23:16], TLP_DW3[31:24] The Packet Change Point registers (EINJ6_CHANGE_POINT*) specify which Tx TLP header bits to replace with the corresponding bits in the Packet Change Value registers (EINJ6_CHANGE_VALUE*). The type and number of errors are specified by the EINJ6_TLP_REG register. Only applies when EINJ6_INVERTED_CONTROL in EINJ6_TLP_REG =0.

PCIE_X1_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_VALUE_H3_REG_0

Offset: 0x228

PCIE_X4_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_VALUE_H3_REG_0

Offset: 0x244

PCIE_X8_RC_PF0_RAS_DES_CAP_EINJ6_CHANGE_VALUE_H3_REG_0

Offset: 0x258

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_CHANGE_VALUE_H3**: Packet Change Value: 4th DWORD. Specifies replacement values for the Tx TLP header DWORD#3 bits defined in the Packet Change Point registers (EINJ6_CHANGE_POINT*). Only applies when the EINJ6_INVERTED_CONTROL field in EINJ6_TLP_REG is '0'. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_EINJ6_TLP_REG_0

where <*i*> = 1, 4, 8.

Description: The Packet Compare Point registers (EINJ6_COMPARE_POINT*) specify which Tx TLP header bits to compare with the corresponding bits in the Packet Compare Value registers (EINJ6_COMPARE_VALUE*). When all specified bits (in the Tx TLP header and EINJ6_COMPARE_VALUE*) match, the controller inserts errors into the TLP. The type and number of errors are specified by the this register. The Packet Change Point registers (EINJ6_CHANGE_POINT*) specify which Tx TLP header bits to replace with the corresponding bits in the Packet Change Value registers (EINJ6_CHANGE_VALUE*). The type and number of errors are specified by the this register. Only applies when EINJ6_INVERTED_CONTROL in this register =0. The TLP into that errors are injected will not arrive at the transaction layer of the remote device when all of the following conditions are true. - Using 128b/130b encoding - Injecting errors into TLP Length field / TLP digest bit.

PCIE_X1_RC_PF0_RAS_DES_CAP_EINJ6_TLP_REG_0

Offset: 0x22c

PCIE_X4_RC_PF0_RAS_DES_CAP_EINJ6_TLP_REG_0

Offset: 0x248

PCIE_X8_RC_PF0_RAS_DES_CAP_EINJ6_TLP_REG_0

Offset: 0x25c

Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:12	RO	0x0	**RSVDP_12**: Reserved for future use.
11:9	RW	0x0	**EINJ6_PACKET_TYPE**: Packet type. Selects the TLP packets to inject errors into. All encodings other than the specified encodings are reserved. Note: This register field is sticky.
8	RW	0x0	**EINJ6_INVERTED_CONTROL**: Inverted Error Injection Control. Encodded vlues given as above. Note: This register field is sticky.
7:0	RW	0x0	**EINJ6_COUNT**: Error Injection Count. Indicates the number of errors to insert. This counter is decremented while errors are been inserted. - If the counter value is 0x01 and error is inserted, ERROR_INJECTION6_ENABLE in EINJ_ENABLE_REG returns '0'. - If the counter value is 0x00 and ERROR_INJECTION6_ENABLE=1, errors are inserted until ERROR_INJECTION6_ENABLE is set to '0'. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_SD_CONTROL1_REG_0

where <*i*> = 1, 4, 8.

Description: For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X1_RC_PF0_RAS_DES_CAP_SD_CONTROL1_REG_0

Offset: 0x240

PCIE_X4_RC_PF0_RAS_DES_CAP_SD_CONTROL1_REG_0

Offset: 0x25c

PCIE_X8_RC_PF0_RAS_DES_CAP_SD_CONTROL1_REG_0

Offset: 0x270

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**RSVDP_24**: Reserved for future use.
23:22	RW	0x0	**LOW_POWER_INTERVAL**: Low Power Entry Interval Time. Interval Time that the controller starts monitoring RXELECIDLE signal after L0s/L1/L2 entry. You should set the value according to the latency from receiving EIOS to, RXELECIDLE assertion at the PHY. Note: This register field is sticky.
21:20	RW	0x0	**TX_EIOS_NUM**: Number of Tx EIOS. This register sets the number of transmit EIOS for L0s/L1 entry and Disable/Loopback/Hot-reset exit. The controller selects the greater value between this register and the value defined by the PCI-SIG specification. Note: This register field is sticky.
19:17	RO	0x0	**RSVDP_17**: Reserved for future use.
16	RW	0x0	**FORCE_DETECT_LANE_EN**: Force Detect Lane Enable. Note: This register field is sticky.
15:0	RW	0x0	**FORCE_DETECT_LANE**: Force Detect Lane. When the FORCE_DETECT_LANE_EN field is set, the controller ignores receiver detection from PHY during LTSSM Detect state and uses this value instead. Value represents lane number. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_SD_CONTROL2_REG_0

where <*i*> = 1, 4, 8.

Description: For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X1_RC_PF0_RAS_DES_CAP_SD_CONTROL2_REG_0

Offset: 0x244

PCIE_X4_RC_PF0_RAS_DES_CAP_SD_CONTROL2_REG_0

Offset: 0x260

PCIE_X8_RC_PF0_RAS_DES_CAP_SD_CONTROL2_REG_0

Offset: 0x274

Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:17	RO	0x0	**RSVDP_17**: Reserved for future use.
16	RW	0x0	**FRAMING_ERR_RECOVERY_DISABLE**: Framing Error Recovery Disable. This bit disables a transition to Recovery state when a Framing Error is occurred. Note: This register field is sticky.
15:11	RO	0x0	**RSVDP_11**: Reserved for future use.
10	RW	0x0	**DIRECT_LPBKSLV_TO_EXIT**: Direct Loopback Slave To Exit. Note: This register field is sticky.
9	RW	0x0	**DIRECT_POLCOMP_TO_DETECT**: Direct Polling.Compliance to Detect. Note: This register field is sticky.
8	RW	0x0	**DIRECT_RECIDLE_TO_CONFIG**: Direct Recovery.Idle to Configuration. Note: This register field is sticky.
7:3	RO	0x0	**RSVDP_3**: Reserved for future use.
2	RW	0x0	**NOACK_FORCE_LINKDOWN**: Force LinkDown. Note: This register field is sticky.
1	WO	0x0	**RECOVERY_REQUEST**: Recovery Request.
0	RW	0x0	**HOLD_LTSSM**: Hold and Release LTSSM. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_SD_STATUS_L1LANE_REG_0

where <*i*> = 1, 4, 8.

Description: This viewport register returns the data selected by the following field: - LANE_SELECT in SD_STATUS_L1LANE_REG For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X1_RC_PF0_RAS_DES_CAP_SD_CONTROL2_REG_0

Offset: 0x244

PCIE_X4_RC_PF0_RAS_DES_CAP_SD_CONTROL2_REG_0

Offset: 0x260

PCIE_X8_RC_PF0_RAS_DES_CAP_SD_CONTROL2_REG_0

Offset: 0x274

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00180000 (0b0000,0000,0001,1000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**DESKEW_POINTER**: Deskew Pointer. Indicates Deskew pointer of internal Deskew buffer of selected lane number(LANE_SELECT). Note: This register field is sticky.
23:21	RO	0x0	**RSVDP_21**: Reserved for future use.
20	RO	0x1	**PIPE_TXELECIDLE**: PIPE:TxElecIdle. Indicates PIPE TXELECIDLE signal of selected lane number(LANE_SELECT). Note: This register field is sticky.
19	RO	0x1	**PIPE_RXELECIDLE**: PIPE:RxElecIdle. Indicates PIPE RXELECIDLE signal of selected lane number(LANE_SELECT). Note: This register field is sticky.
18	RO	0x0	**PIPE_RXVALID**: PIPE:RxValid. Indicates PIPE RXVALID signal of selected lane number(LANE_SELECT). Note: This register field is sticky.
17	RO	0x0	**PIPE_DETECT_LANE**: PIPE:Detect Lane. Indicates whether PHY indicates receiver detection or not on selected lane number(LANE_SELECT). Note: This register field is sticky.
16	RO	0x0	**PIPE_RXPOLARITY**: PIPE:RxPolarity. Indicates PIPE RXPOLARITY signal of selected lane number(LANE_SELECT). Note: This register field is sticky.
15:4	RO	0x0	**RSVDP_4**: Reserved for future use.
3:0	RW	0x0	**LANE_SELECT**: Lane Select. Lane Select register for Silicon Debug Status Register of Layer1-PerLane. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_SD_STATUS_L1LTSSM_REG_0

where <*i*> = 1, 4, 8.

Description: For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X1_RC_PF0_RAS_DES_CAP_SD_STATUS_L1LTSSM_REG_0

Offset: 0x254

PCIE_X4_RC_PF0_RAS_DES_CAP_SD_STATUS_L1LTSSM_REG_0

Offset: 0x270

PCIE_X8_RC_PF0_RAS_DES_CAP_SD_STATUS_L1LTSSM_REG_0

Offset: 0x284

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000200 (0b0000,0000,0000,0000,0000,0010,0000,0000)

Bit	R/W	Reset	Description
31:16	RO	0x0	**LTSSM_VARIABLE**: LTSSM Variable. Indicates internal LTSSM variables defined in the PCI Express Base Specification. For other value idle_to_rlock_transitioned. Note: This register field is sticky.
15	RO	0x0	**LANE_REVERSAL**: Lane Reversal Operation. Receiver detected lane reversal. This field is only valid in the L0 LTSSM state. Note: This register field is sticky.
14:11	RO	0x0	**RSVDP_11**: Reserved for future use.
10:8	RO	0x2	**PIPE_POWER_DOWN**: PIPE:PowerDown. Indicates PIPE PowerDown signal. Note: This register field is sticky.
7	RW	0x0	**FRAMING_ERR**: Framing Error. Indicates Framing Error detection status.
6:0	RO	0x0	**FRAMING_ERR_PTR**: First Framing Error Pointer. Identifies the first Framing Error using the following encoding. The field contents are only valid value when FRAMING_ERR =1. - Received Unexpected Framing Token (Values 01h to 06h) - Received Unexpected STP Token (Values 11h to 13h) - Received Unexpected Block (Values 21h to 2Eh) All encodings other than the defined encodings are reserved. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_SD_STATUS_PM_REG_0

where <*i*> = 1, 4, 8.

Description: For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X1_RC_PF0_RAS_DES_CAP_SD_STATUS_PM_REG_0

Offset: 0x258

PCIE_X4_RC_PF0_RAS_DES_CAP_SD_STATUS_PM_REG_0

Offset: 0x274

PCIE_X8_RC_PF0_RAS_DES_CAP_SD_STATUS_PM_REG_0

Offset: 0x288

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**RSVDP_24**: Reserved for future use.
23:16	RO	0x0	**LATCHED_NFTS**: Latched N_FTS. Indicates the value of N_FTS in the received TS Ordered Sets from the Link Partner. Note: This register field is sticky.
15:13	RO	0x0	**L1SUB_STATE**: L1Sub State. Indicates internal state machine of L1Sub state. Note: This register field is sticky.
12	RW	0x0	**PME_RESEND_FLAG**
11:8	RO	0x0	**INTERNAL_PM_SSTATE**: Internal PM State(Slave). Indicates internal state machine of Power Management Slave controller. Note: This register field is sticky.
7:5	RO	0x0	**RSVDP_5**: Reserved for future use.
4:0	RO	0x0	**INTERNAL_PM_MSTATE**: Internal PM State(Master). Indicates internal state machine of Power Management Master controller. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_SD_STATUS_L2_REG_0

where <*i*> = 1, 4, 8.

Description: For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X1_RC_PF0_RAS_DES_CAP_SD_STATUS_L2_REG_0

Offset: 0x25c

PCIE_X4_RC_PF0_RAS_DES_CAP_SD_STATUS_L2_REG_0

Offset: 0x278

PCIE_X8_RC_PF0_RAS_DES_CAP_SD_STATUS_L2_REG_0

Offset: 0x28c

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00fff000 (0b0000,0000,1111,1111,1111,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**RSVDP_28**: Reserved for future use.
27	0x0	**FC_INIT2**: FC_INIT2. Indicates the controller is in FC_INIT2(VC0) state. Note: This register field is sticky.
26	0x0	**FC_INIT1**: FC_INIT1. Indicates the controller is in FC_INIT1(VC0) state. Note: This register field is sticky.
25:24	0x0	**DLCMSM**: DLCMSM. Indicates the current DLCMSM. Note: This register field is sticky.
23:12	0xfff	**RX_ACK_SEQ_NO**: Tx Ack Sequence Number. Indicates ACKD_SEQ which is updated by receiving ACK/NAK DLLP. Note: This register field is sticky.
11:0	0x0	**TX_TLP_SEQ_NO**: Tx Tlp Sequence Number. Indicates next transmit sequence number for transmit TLP. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_SD_STATUS_L3FC_REG_0

where <*i*> = 1, 4, 8.

Description: The CREDIT_DATA[0/1] fields in this viewport register return the data for the VC and TLP Type selected by the following fields: - CREDIT_SEL_VC - CREDIT_SEL_CREDIT_TYPE - CREDIT_SEL_TLP_TYPE - CREDIT_SEL_HD For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X1_RC_PF0_RAS_DES_CAP_SD_STATUS_L3FC_REG_0

Offset: 0x260

PCIE_X4_RC_PF0_RAS_DES_CAP_SD_STATUS_L3FC_REG_0

Offset: 0x27c

PCIE_X8_RC_PF0_RAS_DES_CAP_SD_STATUS_L3FC_REG_0

Offset: 0x290

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:20	RO	0x0	**CREDIT_DATA1**: Credit Data1. Current FC credit data selected by the CREDIT_SEL_VC, CREDIT_SEL_CREDIT_TYPE, CREDIT_SEL_TLP_TYPE, and CREDIT_SEL_HD viewport-select fields. - Rx: Credit Allocated Value - Tx: Credit Limit Value. This value is valid when DLCMSM=0x3(DL_ACTIVE). Note: This register field is sticky.
19:8	RO	0x0	**CREDIT_DATA0**: Credit Data0. Current FC credit data selected by the CREDIT_SEL_VC, CREDIT_SEL_CREDIT_TYPE, CREDIT_SEL_TLP_TYPE, and CREDIT_SEL_HD viewport-select fields. - Rx: Credit Received Value - Tx: Credit Consumed Value Note: This register field is sticky.
7	RO	0x0	**RSVDP_7**: Reserved for future use.
6	RW	0x0	**CREDIT_SEL_HD**: Credit Select(HeaderData). This field in conjunction with the CREDIT_SEL_VC, CREDIT_SEL_CREDIT_TYPE, and CREDIT_SEL_TLP_TYPE viewport-select fields determines that data that is returned by the CREDIT_DATA0 and CREDIT_DATA1 data fields. Note: This register field is sticky.
5:4	RW	0x0	**CREDIT_SEL_TLP_TYPE**: Credit Select(TLP Type). This field in conjunction with the CREDIT_SEL_VC, CREDIT_SEL_CREDIT_TYPE, and CREDIT_SEL_HD viewport-select fields determines that data that is returned by the CREDIT_DATA0 and CREDIT_DATA1 data fields. Note: This register field is sticky.

Bit	R/W	Reset	Description
3	RW	0x0	**CREDIT_SEL_CREDIT_TYPE**: Credit Select(Credit Type). This field in conjunction with the CREDIT_SEL_VC, CREDIT_SEL_TLP_TYPE, and CREDIT_SEL_HD viewport-select fields determines that data that is returned by the CREDIT_DATA0 and CREDIT_DATA1 data fields. Note: This register field is sticky.
2:0	RW	0x0	**CREDIT_SEL_VC**: Credit Select(VC). This field in conjunction with the CREDIT_SEL_CREDIT_TYPE, CREDIT_SEL_TLP_TYPE, and CREDIT_SEL_HD viewport-select fields determines that data that is returned by the CREDIT_DATA0 and CREDIT_DATA1 data fields. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_SD_STATUS_L3_REG_0

where <*i*> = 1, 4, 8.

Description: Silicon Debug Status(Layer3). For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X1_RC_PF0_RAS_DES_CAP_SD_STATUS_L3_REG_0

Offset: 0x264

PCIE_X4_RC_PF0_RAS_DES_CAP_SD_STATUS_L3_REG_0

Offset: 0x280

PCIE_X8_RC_PF0_RAS_DES_CAP_SD_STATUS_L3_REG_0

Offset: 0x294

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:8	RO	0x0	**RSVDP_8**: Reserved for future use.
7	RW	0x0	**MFTLP_STATUS**: Malformed TLP Status. Indicates malformed TLP has occurred.

Bit	R/W	Reset	Description
6:0	RO	0x0	**MFTLP_POINTER**: First Malformed TLP Error Pointer. Indicates the element of the received first malformed TLP. This pointer is validated by MFTLP_STATUS. All encodings other than the defined encodings are reserved. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_SD_EQ_CONTROL1_REG_0

where <*i*> = 1, 4, 8.

Description: This is a viewport control register. Setting the EQ_RATE_SEL and EQ_LANE_SEL fields in this register determine the per-lane Silicon Debug EQ Status data returned by the SD_EQ_STATUS[1/2/3] viewport registers. For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X1_RC_PF0_RAS_DES_CAP_SD_EQ_CONTROL1_REG_0

Offset: 0x270

PCIE_X4_RC_PF0_RAS_DES_CAP_SD_EQ_CONTROL1_REG_0

Offset: 0x28c

PCIE_X8_RC_PF0_RAS_DES_CAP_SD_EQ_CONTROL1_REG_0

Offset: 0x2a0

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:24	RW	0x0	**FOM_TARGET**: FOM Target. Indicates figure of merit target criteria value of EQ Master(DSP in EQ Phase3/USP in EQ Phase2). This field is only valid when GEN3_EQ_FB_MODE is 0001b(Figure Of Merit). Note: This register field is sticky.
23	RW	0x0	**FOM_TARGET_ENABLE**: FOM Target Enable. Enables the FOM_TARGET fields. Note: This register field is sticky.
22:18	RO	0x0	**RSVDP_18**: Reserved for future use.

Bit	R/W	Reset	Description
17:16	RW	0x0	**EVAL_INTERVAL_TIME**: Eval Interval Time. Indicates interval time of RxEqEval assertion. This field is used for EQ Master(DSP in EQ Phase3/USP in EQ Phase2). Note: This register field is sticky.
15:10	RO	0x0	**RSVDP_10**: Reserved for future use.
9:8	RW	0x0	**EXT_EQ_TIMEOUT**: Extends EQ Phase2/3 Timeout. This field is used when the LTSSM is in Recovery.EQ2/3. When this field is set, the value of EQ2/3 timeout is extended. Note: This register field is sticky.
7:6	RO	0x0	**RSVDP_6**: Reserved for future use.
5:4	RW	0x0	**EQ_RATE_SEL**: EQ Status Rate Select. Setting this field in conjunction with the EQ_LANE_SEL field determines the per-lane Silicon Debug EQ Status data returned by the SD_EQ_CONTROL[2/3] and SD_EQ_STATUS[1/2/3] viewport registers. Note: This register field is sticky.
3:0	RW	0x0	**EQ_LANE_SEL**: EQ Status Lane Select. Setting this field in conjunction with the EQ_RATE_SEL field determines the per-lane Silicon Debug EQ Status data returned by the SD_EQ_CONTROL[2/3] and SD_EQ_STATUS[1/2/3] viewport registers. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_SD_EQ_CONTROL2_REG_0

where <*i*> = 1, 4, 8.

Description: This viewport register returns the value for the rate and lane selected by the EQ_RATE_SEL and EQ_LANE_SEL fields in the SD_EQ_CONTROL1_REG register. For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X1_RC_PF0_RAS_DES_CAP_SD_EQ_CONTROL2_REG_0

Offset: 0x274

PCIE_X4_RC_PF0_RAS_DES_CAP_SD_EQ_CONTROL2_REG_0

Offset: 0x290

PCIE_X8_RC_PF0_RAS_DES_CAP_SD_EQ_CONTROL2_REG_0

Offset: 0x2a4

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RO	0x0	**RSVDP_31**: Reserved for future use.
30	RW	0x0	**FORCE_LOCAL_TX_PRESET_ENABLE**: Force Local Transmitter Preset Enable. Enables the FORCE_LOCAL_TX_PRESET field. If select rate in the EQ_RATE_SEL field is 32.0GT/s Speed, this feature is not available. Note: This register field is sticky.
29	RW	0x0	**FORCE_LOCAL_RX_HINT_ENABLE**: Force Local Receiver Preset Hint Enable. Enables the FORCE_LOCAL_RX_HINT field. If select rate in the EQ_RATE_SEL field is other than 8.0GT/s Speed, this feature is not available. Note: This register field is sticky.
28	RW	0x0	**FORCE_LOCAL_TX_COEF_ENABLE**: Force Local Transmitter Coefficient Enable. Enables the following fields: - FORCE_LOCAL_TX_PRE_CURSOR - FORCE_LOCAL_TX_CURSOR - FORCE_LOCAL_TX_POST_CURSOR Note: This register field is sticky.
27:24	RW	0x0	**FORCE_LOCAL_TX_PRESET**: Force Local Transmitter Preset. Indicates initial preset value of USP in EQ Slave(EQ Phase2) instead of receiving EQ TS2. If select rate in the EQ_RATE_SEL field is 32.0GT/s Speed, this feature is not available. Note: This register field is sticky.
23:21	RO	0x0	**RSVDP_21**: Reserved for future use.
20:18	RW	0x0	**FORCE_LOCAL_RX_HINT**: Force Local Receiver Preset Hint. Indicates the RxPresetHint value of EQ Slave(DSP in EQ Phase2/USP in EQ Phase3), instead of received or set value. If select rate in the EQ_RATE_SEL field is other than 8.0GT/s Speed, this feature is not available. Note: This register field is sticky.
17:12	RW	0x0	**FORCE_LOCAL_TX_POST_CURSOR**: Force Local Transmitter Post-Cursor. Indicates the coefficient value of EQ Slave(DSP in EQ Phase2/USP in EQ Phase3), instead of the value instructed from link partner. Note: This register field is sticky.
11:6	RW	0x0	**FORCE_LOCAL_TX_CURSOR**: Force Local Transmitter Cursor. Indicates the coefficient value of EQ Slave(DSP in EQ Phase2/USP in EQ Phase3), instead of the value instructed from link partner. Note: This register field is sticky.
5:0	RW	0x0	**FORCE_LOCAL_TX_PRE_CURSOR**: Force Local Transmitter Pre-cursor. Indicates the coefficient value of EQ Slave(DSP in EQ Phase2/USP in EQ Phase3), instead of the value instructed from link partner. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_SD_EQ_CONTROL3_REG_0

where <*i*> = 1, 4, 8.

Description: This viewport register returns the value for the rate and lane selected by the EQ_RATE_SEL and EQ_LANE_SEL fields in the SD_EQ_CONTROL1_REG register. For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X1_RC_PF0_RAS_DES_CAP_SD_EQ_CONTROL3_REG_0

Offset: 0x278

PCIE_X4_RC_PF0_RAS_DES_CAP_SD_EQ_CONTROL3_REG_0

Offset: 0x294

PCIE_X8_RC_PF0_RAS_DES_CAP_SD_EQ_CONTROL3_REG_0

Offset: 0x2a8

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:29	RO	0x0	**RSVDP_29**: Reserved for future use.
28	RW	0x0	**FORCE_REMOTE_TX_COEF_ENABLE**: Force Remote Transmitter Coefficient Enable. Enables the following fields: - FORCE_REMOTE_TX_PRE_CURSOR - FORCE_REMOTE_TX_CURSOR - FORCE_REMOTE_TX_POST_CURSOR This function can only be used when GEN3_EQ_FB_MODE = 0000b(Direction Change) Note: This register field is sticky.
27:18	RO	0x0	**RSVDP_18**: Reserved for future use.
17:12	RW	0x0	**FORCE_REMOTE_TX_POST_CURSOR**: Force Remote Transmitter Post-Cursor. Indicates the coefficient value of EQ Master(DSP in EQ Phase3/USP in EQ Phase2), instead of the value instructed from local phy in dirchange mode. Note: This register field is sticky.
11:6	RW	0x0	**FORCE_REMOTE_TX_CURSOR**: Force Remote Transmitter Cursor. Indicates the coefficient value of EQ Master(DSP in EQ Phase3/USP in EQ Phase2), instead of the value instructed from local phy in dirchange mode. Note: This register field is sticky.

Bit	R/W	Reset	Description
5:0	RW	0x0	**FORCE_REMOTE_TX_PRE_CURSOR**: Force Remote Transmitter Pre-Cursor. Indicates the coefficient value of EQ Master(DSP in EQ Phase3/USP in EQ Phase2), instead of the value instructed from local phy in dirchange mode. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_SD_EQ_STATUS1_REG_0

where <*i*> = 1, 4, 8.

Description: This viewport register returns the first of three words of Silicon Debug EQ Status data for the rate and lane selected by the EQ_RATE_SEL and EQ_LANE_SEL fields in the SD_EQ_CONTROL1_REG register. The following fields are available when Equalization finished unsuccessfully(EQ_CONVERGENCE_INFO=2). - EQ_RULEA_VIOLATION - EQ_RULEB_VIOLATION - EQ_RULEC_VIOLATION - EQ_REJECT_EVENT For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X1_RC_PF0_RAS_DES_CAP_SD_EQ_STATUS1_REG_0

Offset: 0x280

PCIE_X4_RC_PF0_RAS_DES_CAP_SD_EQ_STATUS1_REG_0

Offset: 0x29c

PCIE_X8_RC_PF0_RAS_DES_CAP_SD_EQ_STATUS1_REG_0

Offset: 0x2b0

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:8	0x0	**RSVDP_8**: Reserved for future use.
7	0x0	**EQ_REJECT_EVENT**: EQ Reject Event. Indicates that the controller receives two consecutive TS1 OS w/Reject=1b during EQ Master phase(DSP in EQ Phase3/USP in EQ Phase2). This bit is automatically cleared when the controller starts EQ Master phase again. Note: This register field is sticky.

Bit	Reset	Description
6	0x0	**EQ_RULEC_VIOLATION**: EQ Rule C Violation. Indicates that coefficients rule C violation is detected in the values provided by PHY using direction change method during EQ Master phase(DSP in EQ Phase3/USP in EQ Phase2). The coefficients rule C correspond to the rules c) from section "Rules for Transmitter Coefficents" in the PCI Express Base Specification. This bit is automatically cleared when the controller starts EQ Master phase again. Note: This register field is sticky.
5	0x0	**EQ_RULEB_VIOLATION**: EQ Rule B Violation. Indicates that coefficients rule B violation is detected in the values provided by PHY using direction change method during EQ Master phase(DSP in EQ Phase3/USP in EQ Phase2). The coefficients rules B correspond to the rules b) from section "Rules for Transmitter Coefficents" in the PCI Express Base Specification. This bit is automatically cleared when the controller starts EQ Master phase again. Note: This register field is sticky.
4	0x0	**EQ_RULEA_VIOLATION**: EQ Rule A Violation. Indicates that coefficients rule A violation is detected in the values provided by PHY using direction change method during EQ Master phase(DSP in EQ Phase3/USP in EQ Phase2). The coefficients rules A correspond to the rules a) from section "Rules for Transmitter Coefficents" in the PCI Express Base Specification. This bit is automatically cleared when the controller starts EQ Master phase again. Note: This register field is sticky.
3	0x0	**RSVDP_3**: Reserved for future use.
2:1	0x0	**EQ_CONVERGENCE_INFO**: EQ Convergence Info. Indicates equalization convergence information. This bit is automatically cleared when the controller starts EQ Master phase again. Note: This register field is sticky.
0	0x0	**EQ_SEQUENCE**: EQ Sequence. Indicates that the controller is starting the equalization sequence. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_SD_EQ_STATUS2_REG_0

where <*i*> = 1, 4, 8.

Description: This viewport register returns the second of three words of Silicon Debug EQ Status data for the rate and lane selected by the EQ_RATE_SEL and EQ_LANE_SEL fields in the SD_EQ_CONTROL1_REG register. Each field is available when Equalization finished successfully(EQ_CONVERGENCE_INFO=1). For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X1_RC_PF0_RAS_DES_CAP_SD_EQ_STATUS2_REG_0

Offset: 0x284

PCIE_X4_RC_PF0_RAS_DES_CAP_SD_EQ_STATUS2_REG_0

Offset: 0x2a0

PCIE_X8_RC_PF0_RAS_DES_CAP_SD_EQ_STATUS2_REG_0

Offset: 0x2b4

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**EQ_LOCAL_FOM_VALUE**: EQ Local Figure of Merit. Indicates Local maximum Figure of Merit value. Note: This register field is sticky.
23:21	0x0	**RSVDP_21**: Reserved for future use.
20:18	0x0	**EQ_LOCAL_RX_HINT**: EQ Local Receiver Preset Hint. Indicates Local Receiver Preset Hint value. If select rate in the EQ_RATE_SEL field is other than 8.0GT/s Speed, this feature is not available. Note: This register field is sticky.
17:12	0x0	**EQ_LOCAL_POST_CURSOR**: EQ Local Post-Cursor. Indicates Local post cursor coefficient value. Note: This register field is sticky.
11:6	0x0	**EQ_LOCAL_CURSOR**: EQ Local Cursor. Indicates Local cursor coefficient value. Note: This register field is sticky.
5:0	0x0	**EQ_LOCAL_PRE_CURSOR**: EQ Local Pre-Cursor. Indicates Local pre cursor coefficient value. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_RAS_DES_CAP_SD_EQ_STATUS3_REG_0

where <*i*> = 1, 4, 8.

Description: This viewport register returns the third of three words of Silicon Debug EQ Status data for the rate and lane selected by the EQ_RATE_SEL and EQ_LANE_SEL fields in the SD_EQ_CONTROL1_REG register. Each field is available when Equalization finished successfully(EQ_CONVERGENCE_INFO=1). For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X1_RC_PF0_RAS_DES_CAP_SD_EQ_STATUS3_REG_0

Offset: 0x288

PCIE_X4_RC_PF0_RAS_DES_CAP_SD_EQ_STATUS3_REG_0

Offset: 0x2a4

PCIE_X8_RC_PF0_RAS_DES_CAP_SD_EQ_STATUS3_REG_0

Offset: 0x2b8

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:30	0x0	**RSVDP_30**: Reserved for future use.
29:24	0x0	**EQ_REMOTE_FS**: EQ Remote FS. Indicates Remote FS value. Note: This register field is sticky.
23:18	0x0	**EQ_REMOTE_LF**: EQ Remote LF. Indicates Remote LF value. Note: This register field is sticky.
17:12	0x0	**EQ_REMOTE_POST_CURSOR**: EQ Remote Post-Cursor. Indicates Remote post cursor coefficient value. Note: This register field is sticky.
11:6	0x0	**EQ_REMOTE_CURSOR**: EQ Remote Cursor. Indicates Remote cursor coefficient value. Note: This register field is sticky.
5:0	0x0	**EQ_REMOTE_PRE_CURSOR**: EQ Remote Pre-Cursor. Indicates Remote pre cursor coefficient value. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_VSECRAS_CAP_RASDP_EXT_CAP_HDR_OFF_0

where <*i*> = 1, 4, 8.

Description: This Register provides capability ID, Capability version, and Next capability offset for PCIe Extended capability stucture.

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_VSECRAS_CAP_RASDP_EXT_CAP_HDR_OFF_0

Offset: 0x2a0
Reset: 0x2d81000b (0b0010,1101,1000,0001,0000,0000,0000,1011)

Bit	Reset	Description
31:20	0x2d8	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. The bottom 2 bits of this offset are Reserved and must be implemented as 00b although software must mask them to allow for future uses of these bits. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**CAP**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. A version of the specification that changes the Extended Capability in a way that is not otherwise identifiable (e.g., through a new Capability field) is permitted to increment this field. All such changes to the Capability structure must be software-compatible. Software must check for Capability Version numbers that are greater than or equal to the highest number defined when the software is written, as Functions reporting any such Capability Version numbers will contain a Capability structure that is compatible with that piece of software. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0xb	**ID**: PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X4_RC_PF0_VSECRAS_CAP_RASDP_EXT_CAP_HDR_OFF_0

Offset: 0x2bc
Reset: 0x2f41000b (0b0010,1111,0100,0001,0000,0000,0000,1011)

Bit	Reset	Description
31:20	0x2f4	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. The bottom 2 bits of this offset are Reserved and must be implemented as 00b although software must mask them to allow for future uses of these bits. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

Bit	Reset	Description
19:16	0x1	**CAP**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. A version of the specification that changes the Extended Capability in a way that is not otherwise identifiable (e.g., through a new Capability field) is permitted to increment this field. All such changes to the Capability structure must be software-compatible. Software must check for Capability Version numbers that are greater than or equal to the highest number defined when the software is written, as Functions reporting any such Capability Version numbers will contain a Capability structure that is compatible with that piece of software. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0xb	**ID**: PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X8_RC_PF0_VSECRAS_CAP_RASDP_EXT_CAP_HDR_OFF_0

Offset: 0x2d0
Reset: 0x3081000b (0b0011,0000,1000,0001,0000,0000,0000,1011)

Bit	Reset	Description
31:20	0x308	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. The bottom 2 bits of this offset are Reserved and must be implemented as 00b although software must mask them to allow for future uses of these bits. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**CAP**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. A version of the specification that changes the Extended Capability in a way that is not otherwise identifiable (e.g., through a new Capability field) is permitted to increment this field. All such changes to the Capability structure must be software-compatible. Software must check for Capability Version numbers that are greater than or equal to the highest number defined when the software is written, as Functions reporting any such Capability Version numbers will contain a Capability structure that is compatible with that piece of software. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0xb	**ID**: PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_VSECRAS_CAP_RASDP_VENDOR_SPECIFIC_HDR_OFF_0

where <*i*> = 1, 4, 8.

Description: This Register provides VSEC Length, VSEC ID and VSEC Rev (Version Number). Vendor-specific software must qualify the associated Vendor ID of the PCI Express Function or RCRB before attempting to interpret the values in the VSEC ID or VSEC Rev fields.

PCIE_X1_RC_PF0_VSECRAS_CAP_RASDP_VENDOR_SPECIFIC_HDR_OFF_0

Offset: 0x2a4

PCIE_X4_RC_PF0_VSECRAS_CAP_RASDP_VENDOR_SPECIFIC_HDR_OFF_0

Offset: 0x2c0

PCIE_X8_RC_PF0_VSECRAS_CAP_RASDP_VENDOR_SPECIFIC_HDR_OFF_0

Offset: 0x2d4

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x03810001 (0b0000,0011,1000,0001,0000,0000,0000,0001)

Bit	Reset	Description
31:20	0x38	**VSEC_LENGTH**: VSEC Length. This field indicates the number of bytes in the entire VSEC structure, including the Vendor-Specific Extended Capability Header, the Vendor-Specific Header, and the vendor-specific registers. Note: This register field is sticky.
19:16	0x1	**VSEC_REV**: VSEC Rev. This field is a vendor-defined version number that indicates the version of the VSEC structure. Software must qualify the Vendor ID and VSEC ID before interpreting this field. Note: This register field is sticky.
15:0	0x1	**VSEC_ID**: VSEC ID. This field is a vendor-defined ID number that indicates the nature and format of the VSEC structure. Software must qualify the Vendor ID before interpreting this field. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_VSECRAS_CAP_RASDP_ERROR_PROT_CTRL_OFF_0

where <*i*> = 1, 4, 8.

Description: Allows you to disable ECC error correction for RAMs and datapath. When the AXI Bridge Module is implemented and the master / slave clocks are asynchronous to the PCIe native controller clock (core_clk), you must not write this register while operations are in progress in the AXI master / slave interface.

PCIE_X1_RC_PF0_VSECRAS_CAP_RASDP_ERROR_PROT_CTRL_OFF_0

Offset: 0x2a8

PCIE_X4_RC_PF0_VSECRAS_CAP_RASDP_ERROR_PROT_CTRL_OFF_0

Offset: 0x2c4

PCIE_X8_RC_PF0_VSECRAS_CAP_RASDP_ERROR_PROT_CTRL_OFF_0

Offset: 0x2d8

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**RSVDP_24**: Reserved for future use.
23	RW	0x0	**ERROR_PROT_DISABLE_CXS_RX**: Error correction disable for CXS Tx path (PCIe Rx path). Note: This register field is sticky.
22	RW	0x0	**ERROR_PROT_DISABLE_ADM_RX**: Error correction disable for ADM Rx path. Note: This register field is sticky.
21	RW	0x0	**ERROR_PROT_DISABLE_LAYER3_RX**: Error correction disable for layer 3 Rx path. Note: This register field is sticky.
20	RW	0x0	**ERROR_PROT_DISABLE_LAYER2_RX**: Error correction disable for layer 2 Rx path. Note: This register field is sticky.
19	RW	0x0	**ERROR_PROT_DISABLE_DMA_READ**: Error correction disable for DMA read engine. Note: This register field is sticky.
18	RW	0x0	**ERROR_PROT_DISABLE_AXI_BRIDGE_INBOUND_REQUEST**: Error correction disable for AXI bridge inbound request path. Note: This register field is sticky.
17	RW	0x0	**ERROR_PROT_DISABLE_AXI_BRIDGE_INBOUND_COMPLETION**: Error correction disable for AXI bridge inbound completion composer. Does not disable the error detection reporting for 1-bit and 2-bit ECC errors. Note: This register field is sticky.

Bit	R/W	Reset	Description
16	RW	0x0	**ERROR_PROT_DISABLE_RX**: Global error correction disable for all Rx layers. Note: This register field is sticky.
15:9	RO	0x0	**RSVDP_9**: Reserved for future use.
8	RW	0x0	**ERROR_PROT_DISABLE_DTIM_TX**: Error correction disable for DTIM Tx path. Note: This register field is sticky.
7	RW	0x0	**ERROR_PROT_DISABLE_CXS_TX**: Error correction disable for CXS Rx path (PCIe Tx path). Note: This register field is sticky.
6	RW	0x0	**ERROR_PROT_DISABLE_ADM_TX**: Error correction disable for Adm Tx path. Note: This register field is sticky.
5	RW	0x0	**ERROR_PROT_DISABLE_LAYER3_TX**: Error correction disable for layer 3 Tx path. Note: This register field is sticky.
4	RW	0x0	**ERROR_PROT_DISABLE_LAYER2_TX**: Error correction disable for layer 2 Tx path. Note: This register field is sticky.
3	RW	0x0	**ERROR_PROT_DISABLE_DMA_WRITE**: Error correction disable for DMA write engine. Note: This register field is sticky.
2	RW	0x0	**ERROR_PROT_DISABLE_AXI_BRIDGE_OUTBOUND**: Error correction disable for AXI bridge outbound request path. Note: This register field is sticky.
1	RW	0x0	**ERROR_PROT_DISABLE_AXI_BRIDGE_MASTER**: Error correction disable for AXI bridge master completion buffer. Note: This register field is sticky.
0	RW	0x0	**ERROR_PROT_DISABLE_TX**: Global error correction disable for all Tx layers. Does not disable the error detection reporting for 1-bit and 2-bit ECC errors. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_VSECRAS_CAP_RASDP_CORR_COUNTER_CTRL_OFF_0

where <*i*> = 1, 4, 8.

Description: This is a viewport control register. Setting the CORR_COUNTER_SELECTION_REGION and CORR_COUNTER_SELECTION fields in this register determine the counter data returned by the RASDP_CORR_COUNT_REPORT_OFF viewport data register.

PCIE_X1_RC_PF0_VSECRAS_CAP_RASDP_CORR_COUNTER_CTRL_OFF_0

Offset: 0x2ac

PCIE_X4_RC_PF0_VSECRAS_CAP_RASDP_CORR_COUNTER_CTRL_OFF_0

Offset: 0x2c8

PCIE_X8_RC_PF0_VSECRAS_CAP_RASDP_CORR_COUNTER_CTRL_OFF_0

Offset: 0x2dc

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000010 (0b0000,0000,0000,0000,0000,0000,0001,0000)

Bit	R/W	Reset	Description
31:24	RW	0x0	**CORR_COUNTER_SELECTION**: Counter selection. This field selects the counter ID (within the region defined by CORR_COUNTER_SELECTION_REGION) whose contents can be read from the RASDP_CORR_COUNT_REPORT_OFF register. You can cycle this field value from 0 to 255 to access all counters.
23:20	RW	0x0	**CORR_COUNTER_SELECTION_REGION**: Select correctable counter region. All encodings other than the defined encodings are reserved.
19:5	RO	0x0	**RSVDP_5**: Reserved for future use.
4	RW	0x1	**CORR_EN_COUNTERS**: Enable correctable errors counters. The counters are enabled by default.
3:1	RO	0x0	**RSVDP_1**: Reserved for future use.
0	WO	0x0	**CORR_CLEAR_COUNTERS**: Clear all correctable error counters.

PCIE_X<*i*>_RC_PF0_VSECRAS_CAP_RASDP_CORR_COUNT_REPORT_OFF_0

where <*i*> = 1, 4, 8.

Description: This viewport register returns the counter data selected by the CORR_COUNTER_SELECTION_REGION and CORR_COUNTER_SELECTION fields in the RASDP_CORR_COUNTER_CTRL_OFF register.

PCIE_X1_RC_PF0_VSECRAS_CAP_RASDP_CORR_COUNT_REPORT_OFF_0

Offset: 0x2b0

PCIE_X4_RC_PF0_VSECRAS_CAP_RASDP_CORR_COUNT_REPORT_OFF_0

Offset: 0x2cc

PCIE_X8_RC_PF0_VSECRAS_CAP_RASDP_CORR_COUNTER_CTRL_OFF_0

Offset: 0x2dc

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**CORR_COUNTER_SELECTED**: Counter selection. Returns the value set in the CORR_COUNTER_SELECTION field of the RASDP_CORR_COUNTER_CTRL_OFF register.
23:20	0x0	**CORR_COUNTER_SELECTED_REGION**: Selected correctable counter region. All encodings other than the defined encodings are reserved.
19:8	0x0	**RSVDP_8**: Reserved for future use.
7:0	0x0	**CORR_COUNTER**: Current corrected error count for the selected counter.

PCIE_X<*i*>_RC_PF0_VSECRAS_CAP_RASDP_UNCORR_COUNTER_CTRL_OFF_0

where <*i*> = 1, 4, 8.

Description: This is a viewport control register. Setting the UNCORR_COUNTER_SELECTION_REGION and UNCORR_COUNTER_SELECTION fields in this register determine the counter data returned by the RASDP_UNCORR_COUNT_REPORT_OFF viewport data register.

PCIE_X1_RC_PF0_VSECRAS_CAP_RASDP_UNCORR_COUNTER_CTRL_OFF_0

Offset: 0x2b4

PCIE_X4_RC_PF0_VSECRAS_CAP_RASDP_UNCORR_COUNTER_CTRL_OFF_0

Offset: 0x2d0

PCIE_X8_RC_PF0_VSECRAS_CAP_RASDP_UNCORR_COUNTER_CTRL_OFF_0

Offset: 0x2e4

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000010 (0b0000,0000,0000,0000,0000,0000,0001,0000)

Bit	R/W	Reset	Description
31:24	RW	0x0	**UNCORR_COUNTER_SELECTION**: Counter selection. This field selects the counter ID (within the region defined by UNCORR_COUNTER_SELECTION_REGION) whose contents can be read from the RASDP_UNCORR_COUNT_REPORT_OFF register. You can cycle this field value from 0 to 255 to access all counters.
23:20	RW	0x0	**UNCORR_COUNTER_SELECTION_REGION**: Select uncorrectable counter region. All encodings other than the defined encodings are reserved.
19:5	RO	0x0	**RSVDP_5**: Reserved for future use.
4	RW	0x1	**UNCORR_EN_COUNTERS**: Enable uncorrectable errors counters. The counters are enabled by default.
3:1	RO	0x0	**RSVDP_1**: Reserved for future use.
0	WO	0x0	**UNCORR_CLEAR_COUNTERS**: Clear uncorrectable errors counters. When asserted causes all counters tracking the uncorrectable errors to be cleared.

PCIE_X<*i*>_RC_PF0_VSECRAS_CAP_RASDP_UNCORR_COUNT_REPORT_OFF_0

where <*i*> = 1, 4, 8.

Description: This viewport register returns the counter data selected by the UNCORR_COUNTER_SELECTION_REGION and UNCORR_COUNTER_SELECTION fields in the RASDP_UNCORR_COUNTER_CTRL_OFF register.

PCIE_X1_RC_PF0_VSECRAS_CAP_RASDP_UNCORR_COUNT_REPORT_OFF_0

Offset: 0x2b8

PCIE_X4_RC_PF0_VSECRAS_CAP_RASDP_UNCORR_COUNT_REPORT_OFF_0

Offset: 0x2d4

PCIE_X8_RC_PF0_VSECRAS_CAP_RASDP_UNCORR_COUNT_REPORT_OFF_0

Offset: 0x2e8

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**UNCORR_COUNTER_SELECTED**: Counter selection. Returns the value set in the UNCORR_COUNTER_SELECTION field of the RASDP_UNCORR_COUNTER_CTRL_OFF register.
23:20	0x0	**UNCORR_COUNTER_SELECTED_REGION**: Selected uncorrectable counter region. All encodings other than the defined encodings are reserved.
19:8	0x0	**RSVDP_8**: Reserved for future use.
7:0	0x0	**UNCORR_COUNTER**: Current uncorrected error count for the selected counter

PCIE_X<*i*>_RC_PF0_VSECRAS_CAP_RASDP_ERROR_INJ_CTRL_OFF_0

where <*i*> = 1, 4, 8.

Description: Error injection control for the following features: - 1-bit or 2-bit injection - Continuous or fixed-number (n) injection modes - Global enable/disable - Selectable location where injection occurs.

PCIE_X1_RC_PF0_VSECRAS_CAP_RASDP_ERROR_INJ_CTRL_OFF_0

Offset: 0x2bc

PCIE_X4_RC_PF0_VSECRAS_CAP_RASDP_ERROR_INJ_CTRL_OFF_0

Offset: 0x2d8

PCIE_X8_RC_PF0_VSECRAS_CAP_RASDP_ERROR_INJ_CTRL_OFF_0

Offset: 0x2ec

Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**RSVDP_24**: Reserved for future use.
23:16	RW	0x0	**ERROR_INJ_LOC**: Error injection location. Selects where error injection takes place. You can cycle this field value from 0 to 255 to access all locations.
15:8	RW	0x0	**ERROR_INJ_COUNT**: Error injection count. If value is n, n amount of errors injected.
7:6	RO	0x0	**RSVDP_6**: Reserved for future use.
5:4	RW	0x0	**ERROR_INJ_TYPE**: Error injection type.
3:1	RO	0x0	**RSVDP_1**: Reserved for future use.
0	RW	0x0	**ERROR_INJ_EN**: Error injection global enable. When set, enables the error insertion logic.

PCIE_X<*i*>_RC_PF0_VSECRAS_CAP_RASDP_CORR_ERROR_LOCATION_OFF_0

where <*i*> = 1, 4, 8.

Description: Corrected errors locations. For more details, see the RAS Data Protection (DP) section in the Core Operations chapter of the Databook.

PCIE_X1_RC_PF0_VSECRAS_CAP_RASDP_CORR_ERROR_LOCATION_OFF_0

Offset: 0x2c0

PCIE_X4_RC_PF0_VSECRAS_CAP_RASDP_CORR_ERROR_LOCATION_OFF_0

Offset: 0x2dc

PCIE_X8_RC_PF0_VSECRAS_CAP_RASDP_CORR_ERROR_LOCATION_OFF_0

Offset: 0x2f0

Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00e000e0 (0b0000,0000,1110,0000,0000,0000,1110,0000)

Bit	Reset	Description
31:24	0x0	**LOC_LAST_CORR_ERROR**: Location/ID of the last corrected error within the region defined by REG_LAST_CORR_ERROR. You can cycle this field value from 0 to 255 to access all counters.
23:20	0xe	**REG_LAST_CORR_ERROR**: Region of the last corrected error. All encodings other than the defined encodings are reserved.
19:16	0x0	**RSVDP_16**: Reserved for future use.
15:8	0x0	**LOC_FIRST_CORR_ERROR**: Location/ID of the first corrected error within the region defined by REG_FIRST_CORR_ERROR. You can cycle this field value from 0 to 255 to access all counters.
7:4	0xe	**REG_FIRST_CORR_ERROR**: Region of the first corrected error. All encodings other than the defined encodings are reserved.
3:0	0x0	**RSVDP_0**: Reserved for future use.

PCIE_X<*i*>_RC_PF0_VSECRAS_CAP_RASDP_UNCORR_ERROR_LOCATION_OFF_0

where <*i*> = 1, 4, 8.

Description: Uncorrected errors locations. For more details, see the RAS Data Protection (DP) section in the Core Operations chapter of the Databook.

PCIE_X1_RC_PF0_VSECRAS_CAP_RASDP_UNCORR_ERROR_LOCATION_OFF_0

Offset: 0x2c4

PCIE_X4_RC_PF0_VSECRAS_CAP_RASDP_UNCORR_ERROR_LOCATION_OFF_0

Offset: 0x2e0

PCIE_X8_RC_PF0_VSECRAS_CAP_RASDP_UNCORR_ERROR_LOCATION_OFF_0

Offset: 0x2f4

Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00e000e0 (0b0000,0000,1110,0000,0000,0000,1110,0000)

Bit	Reset	Description
31:24	0x0	**LOC_LAST_UNCORR_ERROR**: Location/ID of the last uncorrected error within the region defined by REG_LAST_UNCORR_ERROR. You can cycle this field value from 0 to 255 to access all counters.
23:20	0xe	**REG_LAST_UNCORR_ERROR**: Region of the last uncorrected error. All encodings other than the defined encodings are reserved.
19:16	0x0	**RSVDP_16**: Reserved for future use.
15:8	0x0	**LOC_FIRST_UNCORR_ERROR**: Location/ID of the first uncorrected error within the region defined by REG_FIRST_UNCORR_ERROR. You can cycle this field value from 0 to 255 to access all counters.
7:4	0xe	**REG_FIRST_UNCORR_ERROR**: Region of the first uncorrected error. All encodings other than the defined encodings are reserved.
3:0	0x0	**RSVDP_0**: Reserved for future use.

PCIE_X<*i*>_RC_PF0_VSECRAS_CAP_RASDP_ERROR_MODE_EN_OFF_0

where <*i*> = 1, 4, 8.

Description: RASDP error mode enable. The controller enters RASDP error mode (if ERROR_MODE_EN =1) upon detection of the first uncorrectable error. During this mode: - Rx TLPs that are forwarded to your application are not guaranteed to be correct; you must discard them. For more details, see the RAS Data Protection (DP) section in the Core Operations chapter of the Databook.

PCIE_X1_RC_PF0_VSECRAS_CAP_RASDP_ERROR_MODE_EN_OFF_0

Offset: 0x2c8

PCIE_X4_RC_PF0_VSECRAS_CAP_RASDP_ERROR_MODE_EN_OFF_0

Offset: 0x2e4

PCIE_X8_RC_PF0_VSECRAS_CAP_RASDP_ERROR_MODE_EN_OFF_0

Offset: 0x2f8

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000001 (0b0000,0000,0000,0000,0000,0000,0000,0001)

Bit	R/W	Reset	Description
31:2	RO	0x0	**RSVDP_2**: Reserved for future use.
1	RW	0x0	**AUTO_LINK_DOWN_EN**: Write '1' to enable the controller to bring the link down when the controller enters RASDP error mode. Note: This register field is sticky.
0	RW	0x1	**ERROR_MODE_EN**: Write '1' to enable the controller enter RASDP error mode when it detects an uncorrectable error. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_VSECRAS_CAP_RASDP_ERROR_MODE_CLEAR_OFF_0

where <*i*> = 1, 4, 8.

Description: Exit RASDP error mode. For more details, see the RAS Data Protection (DP) section in the Core Operations chapter of the Databook.

PCIE_X1_RC_PF0_VSECRAS_CAP_RASDP_ERROR_MODE_CLEAR_OFF_0

Offset: 0x2cc

PCIE_X4_RC_PF0_VSECRAS_CAP_RASDP_ERROR_MODE_CLEAR_OFF_0

Offset: 0x2e8

PCIE_X8_RC_PF0_VSECRAS_CAP_RASDP_ERROR_MODE_CLEAR_OFF_0

Offset: 0x2fc

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:1	RO	0x0	**RSVDP_1**: Reserved for future use.

Bit	R/W	Reset	Description
0	WO	0x0	**ERROR_MODE_CLEAR**: Write '1' to take the controller out of RASDP error mode. The controller will then report uncorrectable errors (through AER internal error reporting) and also stop nullifying/discarding TLPs.

PCIE_X<*i*>_RC_PF0_VSECRAS_CAP_RASDP_RAM_ADDR_CORR_ERROR_OFF_0

where <*i*> = 1, 4, 8.

Description: For more details, see the RAS Data Protection (DP) section in the Core Operations chapter of the Databook.

PCIE_X1_RC_PF0_VSECRAS_CAP_RASDP_RAM_ADDR_CORR_ERROR_OFF_0

Offset: 0x2d0

PCIE_X4_RC_PF0_VSECRAS_CAP_RASDP_RAM_ADDR_CORR_ERROR_OFF_0

Offset: 0x2ec

PCIE_X8_RC_PF0_VSECRAS_CAP_RASDP_RAM_ADDR_CORR_ERROR_OFF_0

Offset: 0x300

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**RAM_INDEX_CORR_ERROR**: RAM index where a corrected error (1-bit ECC) is detected.
27	0x0	**RSVDP_27**: Reserved for future use.
26:0	0x0	**RAM_ADDR_CORR_ERROR**: RAM Address where a corrected error (1-bit ECC) is detected.

PCIE_X<*i*>_RC_PF0_VSECRAS_CAP_RASDP_RAM_ADDR_UNCORR_ERROR_OFF_0

where <*i*> = 1, 4, 8.

Description: For more details, see the RAS Data Protection (DP) section in the Core Operations chapter of the Databook.

PCIE_X1_RC_PF0_VSECRAS_CAP_RASDP_RAM_ADDR_UNCORR_ERROR_OFF_0

Offset: 0x2d4

PCIE_X4_RC_PF0_VSECRAS_CAP_RASDP_RAM_ADDR_UNCORR_ERROR_OFF_0

Offset: 0x2f0

PCIE_X8_RC_PF0_VSECRAS_CAP_RASDP_RAM_ADDR_UNCORR_ERROR_OFF_0

Offset: 0x304

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**RAM_INDEX_UNCORR_ERROR**: RAM index where an uncorrected error (2-bit ECC) is detected.
27	0x0	**RSVDP_27**: Reserved for future use.
26:0	0x0	**RAM_ADDR_UNCORR_ERROR**: RAM Address where an uncorrected error (2-bit ECC) is detected.

PCIE_X<*i*>_RC_PF0_DLINK_CAP_DATA_LINK_FEATURE_EXT_HDR_OFF_0

where <*i*> = 1, 4, 8.

Description: This register provides capability ID, capability version, and next offset value.

PCIE_X1_RC_PF0_DLINK_CAP_DATA_LINK_FEATURE_EXT_HDR_OFF_0

Offset: 0x2d8

PCIE_X4_RC_PF0_DLINK_CAP_DATA_LINK_FEATURE_EXT_HDR_OFF_0

Offset: 0x2f4

PCIE_X8_RC_PF0_DLINK_CAP_DATA_LINK_FEATURE_EXT_HDR_OFF_0

Offset: 0x308

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x2e410025 (0b0010,1110,0100,0001,0000,0000,0010,0101)

Bit	Reset	Description
31:20	0x2e4	**DLINK_NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h, if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. The bottom 2 bits of this offset are Reserved and must be implemented as 00b although software must mask them to allow for future uses of these bits. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**DLINK_CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. This field depends on the version of the specification. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x25	**DLINK_EXT_CAP_ID**: Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. Extended Capability ID for Data Link Feature is 0025h. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_DLINK_CAP_DATA_LINK_FEATURE_CAP_OFF_0

where <*i*> = 1, 4, 8.

Description: This register provides description about extended feature.

PCIE_X1_RC_PF0_DLINK_CAP_DATA_LINK_FEATURE_CAP_OFF_0

Offset: 0x2dc

PCIE_X4_RC_PF0_DLINK_CAP_DATA_LINK_FEATURE_CAP_OFF_0

Offset: 0x2f8

PCIE_X8_RC_PF0_DLINK_CAP_DATA_LINK_FEATURE_CAP_OFF_0

Offset: 0x30c

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x80000001 (0b1000,0000,0000,0000,0000,0000,0000,0001)

Bit	R/W	Reset	Description
31	RW	0x1	**DL_FEATURE_EXCHANGE_EN**: Data Link Feature Exchange Enable. If Set, this bit indicates that this Port will enter the DL_Feature negotiation state. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky)
30:23	RO	0x0	**RSVDP_23**: Reserved for future use.
22:1	RW	0x0	**FUTURE_FEATURE_SUPPORTED**: Local Future Data Link Feature Supported. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
0	RW	0x1	**SCALED_FLOW_CNTL_SUPPORTED**: Local Scaled Flow Control Supported. - Bit 0: Local Scaled Flow Control Supported - Bits [22:1]: RsvdP Bits associated with features that this Port is capable of supporting are HwInit, defaulting to 1b. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky)

PCIE_X<*i*>_RC_PF0_DLINK_CAP_DATA_LINK_FEATURE_STATUS_OFF_0

where <*i*> = 1, 4, 8.

Description: This register provides status of the capability of data link feature.

PCIE_X1_RC_PF0_DLINK_CAP_DATA_LINK_FEATURE_STATUS_OFF_0

Offset: 0x2e0

PCIE_X4_RC_PF0_DLINK_CAP_DATA_LINK_FEATURE_STATUS_OFF_0

Offset: 0x2fc

PCIE_X8_RC_PF0_DLINK_CAP_DATA_LINK_FEATURE_STATUS_OFF_0

Offset: 0x310

Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**DATA_LINK_FEATURE_STATUS_VALID**: Remote Data Link Feature Supported Valid. This field indicates that the Port has received a Data Link Feature DLLP in state DL_Feature and that the Remote Data Link Feature Supported and Remote Data Link Feature Ack fields are meaningful. This field is Cleared on entry to state DL_Inactive.
30:23	0x0	**RSVDP_23**: Reserved for future use.
22:0	0x0	**REMOTE_DATA_LINK_FEATURE_SUPPORTED**: Remote Data Link Feature Supported. - Bit 0: Remote Scaled Flow Control Supported - Bits [22:1]: Undefined

PCIE_X<*i*>_RC_PF0_PTM_CAP_PTM_EXT_CAP_HDR_OFF_0

where <*i*> = 1, 4, 8.

Description: This Register provides Capability ID, Cabability Version, and Next Offset of Precision Time Measurement Capability structure.

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_PTM_CAP_PTM_EXT_CAP_HDR_OFF_0

Offset: 0x2e4
Reset: 0x2f01001f (0b0010,1111,0000,0001,0000,0000,0001,1111)

Bit	Reset	Description
31:20	0x2f0	**PTM_NEXT_OFFSET**: Precision Time Measurement PCI Express Extended Capability Next Offset. This field contains the offset to the next PCI Express Extended Capability structure or 000h if no other items exist in the linked list of Capabilities. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**PTM_CAP_VERSION**: Precision Time Measurement PCI Express Extended Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. This bit depends on the version of the specification. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

Bit	Reset	Description
15:0	0x1f	**PTM_CAP_ID**: Precision Time Measurement PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. PCI Express Extended Capability ID for the Precision Time Measurement Capability is 001Fh. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X4_RC_PF0_PTM_CAP_PTM_EXT_CAP_HDR_OFF_0

Offset: 0x300
Reset: 0x30c1001f (0b0011,0000,1100,0001,0000,0000,0001,1111)

Bit	Reset	Description
31:20	0x30c	**PTM_NEXT_OFFSET**: Precision Time Measurement PCI Express Extended Capability Next Offset. This field contains the offset to the next PCI Express Extended Capability structure or 000h if no other items exist in the linked list of Capabilities. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**PTM_CAP_VERSION**: Precision Time Measurement PCI Express Extended Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. This bit depends on the version of the specification. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x1f	**PTM_CAP_ID**: Precision Time Measurement PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. PCI Express Extended Capability ID for the Precision Time Measurement Capability is 001Fh. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X8_RC_PF0_PTM_CAP_PTM_EXT_CAP_HDR_OFF_0

Offset: 0x314
Reset: 0x3201001f (0b0011,0010,0000,0001,0000,0000,0001,1111)

Bit	Reset	Description
31:20	0x320	**PTM_NEXT_OFFSET**: Precision Time Measurement PCI Express Extended Capability Next Offset. This field contains the offset to the next PCI Express Extended Capability structure or 000h if no other items exist in the linked list of Capabilities. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

Bit	Reset	Description
19:16	0x1	**PTM_CAP_VERSION**: Precision Time Measurement PCI Express Extended Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. This bit depends on the version of the specification. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x1f	**PTM_CAP_ID**: Precision Time Measurement PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. PCI Express Extended Capability ID for the Precision Time Measurement Capability is 001Fh. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PTM_CAP_PTM_CAP_OFF_0

where <*i*> = 1, 4, 8.

Description: This register describes a Function's support for Precision Time Measurement. Not all fields within this register apply to all Functions capable of implementing PTM.

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_PTM_CAP_PTM_CAP_OFF_0

Offset: 0x2e8
Reset: 0x00001006 (0b0000,0000,0000,0000,0001,0000,0000,0110)

Bit	R/W	Reset	Description
31:16	RO	0x0	**RSVDP_16**: Reserved for future use.
15:8	RW	0x10	**PTM_CLK_GRAN**: PTM Local Clock Granularity. For other than this value (between b00000001-b11111110) indicates the period of this Time Source's local clock in ns. If the PTM Root Select bit is Set, this local clock is used to provide PTM Master Time. Otherwise, the Time Source uses this local clock to locally track PTM Master Time received from further Upstream within a PTM Hierarchy. This field is reserved for Functions that do not implement the PTM Time Source role. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky)
7:3	RO	0x0	**RSVDP_3**: Reserved for future use.

Bit	R/W	Reset	Description
2	RW	0x1	**PTM_ROOT_CAPABLE**: PTM Root Capable. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
1	RW	0x1	**PTM_RES_CAPABLE**: PTM Responder Capable. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
0	RW	0x0	**PTM_REQ_CAPABLE**: PTM Requester Capable. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X4_RC_PF0_PTM_CAP_PTM_CAP_OFF_0

Offset: 0x304

PCIE_X8_RC_PF0_PTM_CAP_PTM_CAP_OFF_0

Offset: 0x318
Reset: 0x00001007 (0b0000,0000,0000,0000,0001,0000,0000,0111)

Bit	R/W	Reset	Description
31:16	RO	0x0	**RSVDP_16**: Reserved for future use.
15:8	RW	0x10	**PTM_CLK_GRAN**: PTM Local Clock Granularity. For other than this value (between b00000001-b111111110) indicates the period of this Time Source's local clock in ns. If the PTM Root Select bit is Set, this local clock is used to provide PTM Master Time. Otherwise, the Time Source uses this local clock to locally track PTM Master Time received from further Upstream within a PTM Hierarchy. This field is reserved for Functions that do not implement the PTM Time Source role. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky)
7:3	RO	0x0	**RSVDP_3**: Reserved for future use.
2	RW	0x1	**PTM_ROOT_CAPABLE**: PTM Root Capable. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
1	RW	0x1	**PTM_RES_CAPABLE**: PTM Responder Capable. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
0	RW	0x1	**PTM_REQ_CAPABLE**: PTM Requester Capable. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PTM_CAP_PTM_CONTROL_OFF_0

where <*i*> = 1, 4, 8.

Description: This register controls a Function's participation in the Precision Time Measurement mechanism. Not all fields within this register apply to all Functions capable of implementing PTM.

PCIE_X1_RC_PF0_PTM_CAP_PTM_CONTROL_OFF_0

Offset: 0x2ec

PCIE_X4_RC_PF0_PTM_CAP_PTM_CONTROL_OFF_0

Offset: 0x308

PCIE_X8_RC_PF0_PTM_CAP_PTM_CONTROL_OFF_0

Offset: 0x31c

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:16	RO	0x0	**RSVDP_16**: Reserved for future use.
15:8	RO	0x0	**EFF_GRAN**: PTM Effective Granularity. For Functions implementing the PTM Requester Role, this field provides information relating to the expected accuracy of the PTM clock, but does not otherwise affect the PTM mechanism. For Endpoints, system software must program this field to the value representing the maximum Local Clock Granularity reported by the PTM Root and all intervening PTM Time Sources. For RCiEPs, system software must set this field to the value reported in the Local Clock Granularity field by the associated PTM Time Source. For other than this value (between b00000001-b11111110) indicates the effective PTM granularity in ns. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: HWINIT
7:2	RO	0x0	**RSVDP_2**: Reserved for future use.
1	RO	0x0	**ROOT_SELECT**: PTM Root Select. If the value of the PTM Root Capable bit is 0b, this bit is permitted to be hardwired to 0b by the controller. Within each PTM Hierarchy, it is recommended that system software select only the furthest Upstream Time Source to be the PTM Root. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: HWINIT

Bit	R/W	Reset	Description
0	RW	0x0	**PTM_ENABLE**: PTM Enable.

PCIE_X<*i*>_RC_PF0_PTM_RES_CAP_PTM_RES_CAP_HDR_OFF_0

where <*i*> = 1, 4, 8.

Description: For more details, see the PTM section in the Databook.

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_PTM_RES_CAP_PTM_RES_CAP_HDR_OFF_0

Offset: 0x2f0
Reset: 0x3581000b (0b0011,0101,1000,0001,0000,0000,0000,1011)

Bit	Reset	Description
31:20	0x358	**PTM_RES_EXT_CAP_NEXT_OFFS**: Precision Time Measurement Responder VSEC Next Pointer. For more details, see the PTM section in the Databook. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**PTM_RES_EXT_CAP_VER**: Precision Time Measurement Responder VSEC Version. For more details, see the PTM section in the Databook. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0xb	**PTM_RES_EXT_CAP_ID**: Precision Time Measurement Responder VSEC ID. For more details, see the PTM section in the Databook. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X4_RC_PF0_PTM_RES_CAP_PTM_RES_CAP_HDR_OFF_0

Offset: 0x30c
Reset: 0x3741000b (0b0011,0111,0100,0001,0000,0000,0000,1011)

Bit	Reset	Description
31:20	0x374	**PTM_RES_EXT_CAP_NEXT_OFFS**: Precision Time Measurement Responder VSEC Next Pointer. For more details, see the PTM section in the Databook. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**PTM_RES_EXT_CAP_VER**: Precision Time Measurement Responder VSEC Version. For more details, see the PTM section in the Databook. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0xb	**PTM_RES_EXT_CAP_ID**: Precision Time Measurement Responder VSEC ID. For more details, see the PTM section in the Databook. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X8_RC_PF0_PTM_RES_CAP_PTM_RES_CAP_HDR_OFF_0

Offset: 0x320
Reset: 0x3881000b (0b0011,1000,1000,0001,0000,0000,0000,1011)

Bit	Reset	Description
31:20	0x388	**PTM_RES_EXT_CAP_NEXT_OFFS**: Precision Time Measurement Responder VSEC Next Pointer. For more details, see the PTM section in the Databook. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**PTM_RES_EXT_CAP_VER**: Precision Time Measurement Responder VSEC Version. For more details, see the PTM section in the Databook. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0xb	**PTM_RES_EXT_CAP_ID**: Precision Time Measurement Responder VSEC ID. For more details, see the PTM section in the Databook. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PTM_RES_CAP_PTM_RES_HDR_OFF_0

where <*i*> = 1, 4, 8.

Description: For more details, see the PTM section in the Databook.

PCIE_X1_RC_PF0_PTM_RES_CAP_PTM_RES_HDR_OFF_0

Offset: 0x2f4

PCIE_X4_RC_PF0_PTM_RES_CAP_PTM_RES_HDR_OFF_0

Offset: 0x310

PCIE_X8_RC_PF0_PTM_RES_CAP_PTM_RES_HDR_OFF_0

Offset: 0x324

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x05410004 (0b0000,0101,0100,0001,0000,0000,0000,0100)

Bit	Reset	Description
31:20	0x54	**PTM_RES_VSEC_LENGTH**: PTM Responder VSEC Length. For more details, see the PTM section in the Databook. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**PTM_RES_VSEC_REV**: PTM Responder VSEC Revision. For more details, see the PTM section in the Databook. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x4	**PTM_RES_VSEC_ID**: PTM Responder VSEC ID. For more details, see the PTM section in the Databook. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PTM_RES_CAP_PTM_RES_CONTROL_OFF_0

where <*i*> = 1, 4, 8.

Description: For more details, see the PTM section in the Databook.

PCIE_X1_RC_PF0_PTM_RES_CAP_PTM_RES_CONTROL_OFF_0

Offset: 0x2f8

PCIE_X4_RC_PF0_PTM_RES_CAP_PTM_RES_CONTROL_OFF_0

Offset: 0x314

PCIE_X8_RC_PF0_PTM_RES_CAP_PTM_RES_CONTROL_OFF_0

Offset: 0x328

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:2	RO	0x0	**RSVDP_2**: Reserved for future use.
1	RW	0x0	**PTM_RES_PDEL_BYTE_REV**: PTM Requester Propagation Delay Byte Reversal. For more details, see the PTM section in the Databook. Note: This register field is sticky.
0	RW	0x0	**PTM_RES_CCONTEXT_VALID**: PTM Responder Control Context Valid. PTM Local Timing is valid. This bit is set over the DBI. A speed change or aux_clk_active will set this bit low. For more details, see the PTM section in the Databook. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PTM_RES_CAP_PTM_RES_STATUS_OFF_0

where <*i*> = 1, 4, 8.

Description: For more details, see the PTM section in the Databook.

PCIE_X1_RC_PF0_PTM_RES_CAP_PTM_RES_STATUS_OFF_0

Offset: 0x2fc

PCIE_X4_RC_PF0_PTM_RES_CAP_PTM_RES_STATUS_OFF_0

Offset: 0x318

PCIE_X8_RC_PF0_PTM_RES_CAP_PTM_RES_STATUS_OFF_0

Offset: 0x32c

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:2	0x0	**RSVDP_2**: Reserved for future use.
1	0x0	**PTM_RES_REQUEST_RECEIVED**: PTM 1st Request Received. PTM Responder has received the first PTM Request Message. Upon receipt of a second PTM Request Message a ResponseD message with timing information will be sent from the Responder, if the context is valid. If the context is invalid a Response message will be sent instead. For more details, see the PTM section in the Databook. Note: This register field is sticky.
0	0x0	**PTM_RES_CONTEXT_VALID**: PTM Responder Status Context Valid. PTM Local Timing Context is Valid. Value set from upstream port Requester in a Switch. Shadows the same in the control register in all other products. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PTM_RES_CAP_PTM_RES_LOCAL_LSB_OFF_0

where <*i*> = 1, 4, 8.

Description: For more details, see the PTM section in the Databook.

PCIE_X1_RC_PF0_PTM_RES_CAP_PTM_RES_LOCAL_LSB_OFF_0

Offset: 0x300

PCIE_X4_RC_PF0_PTM_RES_CAP_PTM_RES_LOCAL_LSB_OFF_0

Offset: 0x31c

PCIE_X8_RC_PF0_PTM_RES_CAP_PTM_RES_LOCAL_LSB_OFF_0

Offset: 0x330

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PTM_RES_LOCAL_LSB**: PTM Responder Local Clock LSB. Lower 32 bits of local timer value. For more details, see the PTM section in the Databook. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PTM_RES_CAP_PTM_RES_LOCAL_MSB_OFF_0

where <*i*> = 1, 4, 8.

Description: For more details, see the PTM section in the Databook.

PCIE_X1_RC_PF0_PTM_RES_CAP_PTM_RES_LOCAL_MSB_OFF_0

Offset: 0x304

PCIE_X4_RC_PF0_PTM_RES_CAP_PTM_RES_LOCAL_MSB_OFF_0

Offset: 0x320

PCIE_X8_RC_PF0_PTM_RES_CAP_PTM_RES_LOCAL_MSB_OFF_0

Offset: 0x334

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PTM_RES_LOCAL_MSB**: PTM Responder Local Clock MSB. Upper 32 bits of local timer value. For more details, see the PTM section in the Databook. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PTM_RES_CAP_PTM_RES_T2_LSB_OFF_0

where <*i*> = 1, 4, 8.

Description: For more details, see the PTM section in the Databook.

PCIE_X1_RC_PF0_PTM_RES_CAP_PTM_RES_T2_LSB_OFF_0

Offset: 0x308

PCIE_X4_RC_PF0_PTM_RES_CAP_PTM_RES_T2_LSB_OFF_0

Offset: 0x324

PCIE_X8_RC_PF0_PTM_RES_CAP_PTM_RES_T2_LSB_OFF_0

Offset: 0x338

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PTM_RES_T2_LSB**: PTM Responder T2 Timestamp LSB. Lower 32 bits of the T2 Timestamp value. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PTM_RES_CAP_PTM_RES_T2_MSB_OFF_0

where <*i*> = 1, 4, 8.

Description: For more details, see the PTM section in the Databook.

PCIE_X1_RC_PF0_PTM_RES_CAP_PTM_RES_T2_MSB_OFF_0

Offset: 0x30c

PCIE_X4_RC_PF0_PTM_RES_CAP_PTM_RES_T2_MSB_OFF_0

Offset: 0x328

PCIE_X8_RC_PF0_PTM_RES_CAP_PTM_RES_T2_MSB_OFF_0

Offset: 0x33c

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PTM_RES_T2_MSB**: PTM Responder T2 Timestamp MSB. Upper 32 bits of the T2 Timestamp value. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PTM_RES_CAP_PTM_RES_T2P_LSB_OFF_0

where <*i*> = 1, 4, 8.

Description: For more details, see the PTM section in the Databook.

PCIE_X1_RC_PF0_PTM_RES_CAP_PTM_RES_T2P_LSB_OFF_0

Offset: 0x310

PCIE_X4_RC_PF0_PTM_RES_CAP_PTM_RES_T2P_LSB_OFF_0

Offset: 0x32c

PCIE_X8_RC_PF0_PTM_RES_CAP_PTM_RES_T2P_LSB_OFF_0

Offset: 0x340

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PTM_RES_T2P_LSB**: PTM Responder T2 Previous Timestamp LSB. Lower 32 bits of the previously stored T2 Timestamp value. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PTM_RES_CAP_PTM_RES_T2P_MSB_OFF_0

where <*i*> = 1, 4, 8.

Description: For more details, see the PTM section in the Databook.

PCIE_X1_RC_PF0_PTM_RES_CAP_PTM_RES_T2P_MSB_OFF_0

Offset: 0x314

PCIE_X4_RC_PF0_PTM_RES_CAP_PTM_RES_T2P_MSB_OFF_0

Offset: 0x330

PCIE_X8_RC_PF0_PTM_RES_CAP_PTM_RES_T2P_MSB_OFF_0

Offset: 0x344

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PTM_RES_T2P_MSB**: PTM Responder T2 Previous Timestamp MSB. Upper 32 bits of the previously stored T2 Timestamp value. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PTM_RES_CAP_PTM_RES_T3_LSB_OFF_0

where <*i*> = 1, 4, 8.

Description: For more details, see the PTM section in the Databook.

PCIE_X1_RC_PF0_PTM_RES_CAP_PTM_RES_T3_LSB_OFF_0

Offset: 0x318

PCIE_X4_RC_PF0_PTM_RES_CAP_PTM_RES_T3_LSB_OFF_0

Offset: 0x334

PCIE_X8_RC_PF0_PTM_RES_CAP_PTM_RES_T3_LSB_OFF_0

Offset: 0x348

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PTM_RES_T3_LSB**: PTM Responder T3 Timestamp LSB. Lower 32 bits of the T3 Timestamp value. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PTM_RES_CAP_PTM_RES_T3_MSB_OFF_0

where <*i*> = 1, 4, 8.

Description: For more details, see the PTM section in the Databook.

PCIE_X1_RC_PF0_PTM_RES_CAP_PTM_RES_T3_MSB_OFF_0

Offset: 0x31c

PCIE_X4_RC_PF0_PTM_RES_CAP_PTM_RES_T3_MSB_OFF_0

Offset: 0x338

PCIE_X8_RC_PF0_PTM_RES_CAP_PTM_RES_T3_MSB_OFF_0

Offset: 0x34c

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PTM_RES_T3_MSB**: PTM Responder T3 Timestamp MSB. Upper 32 bits of the T2 Timestamp value. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PTM_RES_CAP_PTM_RES_T3P_LSB_OFF_0

where <*i*> = 1, 4, 8.

Description: For more details, see the PTM section in the Databook.

PCIE_X1_RC_PF0_PTM_RES_CAP_PTM_RES_T3P_LSB_OFF_0

Offset: 0x320

PCIE_X4_RC_PF0_PTM_RES_CAP_PTM_RES_T3P_LSB_OFF_0

Offset: 0x33c

PCIE_X8_RC_PF0_PTM_RES_CAP_PTM_RES_T3P_LSB_OFF_0

Offset: 0x350

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PTM_RES_T3P_LSB**: PTM Responder T3 Previous Timestamp LSB. Lower 32 bits of the previously stored T3 Timestamp value. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PTM_RES_CAP_PTM_RES_T3P_MSB_OFF_0

where <*i*> = 1, 4, 8.

Description: For more details, see the PTM section in the Databook.

PCIE_X1_RC_PF0_PTM_RES_CAP_PTM_RES_T3P_MSB_OFF_0

Offset: 0x324

PCIE_X4_RC_PF0_PTM_RES_CAP_PTM_RES_T3P_MSB_OFF_0

Offset: 0x340

PCIE_X8_RC_PF0_PTM_RES_CAP_PTM_RES_T3P_MSB_OFF_0

Offset: 0x354

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PTM_RES_T3P_MSB**: PTM Responder T3 Previous Timestamp MSB. Upper 32 bits of the previously stored T3 Timestamp value. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PTM_RES_CAP_PTM_RES_TX_LATENCY_OFF_0

where <*i*> = 1, 4, 8.

Description: For more details, see the PTM section in the Databook.

PCIE_X1_RC_PF0_PTM_RES_CAP_PTM_RES_TX_LATENCY_OFF_0

Offset: 0x328

PCIE_X4_RC_PF0_PTM_RES_CAP_PTM_RES_TX_LATENCY_OFF_0

Offset: 0x344

PCIE_X8_RC_PF0_PTM_RES_CAP_PTM_RES_TX_LATENCY_OFF_0

Offset: 0x358

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000018 (0b0000,0000,0000,0000,0000,0000,0001,1000)

Bit	R/W	Reset	Description
31:12	RO	0x0	**RSVDP_12**: Reserved for future use.
11:0	RW	0x18	**PTM_RES_TX_LATENCY**: PTM Responder TX Latency. Responder Transmit path latency viewport register. A register is provided for each supported link speed, and the value used for timestamp adjustment is automatically selected based on the current speed. The PTM Responder Latency Register Select register specifies the register to be written or read. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PTM_RES_CAP_PTM_RES_RX_LATENCY_OFF_0

where <*i*> = 1, 4, 8.

Description: For more details, see the PTM section in the Databook.

PCIE_X1_RC_PF0_PTM_RES_CAP_PTM_RES_RX_LATENCY_OFF_0

Offset: 0x32c

PCIE_X4_RC_PF0_PTM_RES_CAP_PTM_RES_RX_LATENCY_OFF_0

Offset: 0x348

PCIE_X8_RC_PF0_PTM_RES_CAP_PTM_RES_RX_LATENCY_OFF_0

Offset: 0x35c

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000001f (0b0000,0000,0000,0000,0000,0000,0001,1111)

Bit	R/W	Reset	Description
31:12	RO	0x0	**RSVDP_12**: Reserved for future use.
11:0	RW	0x1f	**PTM_RES_RX_LATENCY**: PTM Responder RX Latency. Responder Receive path latency viewport register. A register is provided for each supported link speed, and the value used for timestamp adjustment is automatically selected based on the current speed. The PTM Responder Latency Register Select register specifies the register to be written or read. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PTM_RES_CAP_PTM_RES_NOM_CLOCK_T_OFF_0

where <*i*> = 1, 4, 8.

Description: For more details, see the PTM section in the Databook.

PCIE_X1_RC_PF0_PTM_RES_CAP_PTM_RES_NOM_CLOCK_T_OFF_0

Offset: 0x330

PCIE_X4_RC_PF0_PTM_RES_CAP_PTM_RES_NOM_CLOCK_T_OFF_0

Offset: 0x34c

PCIE_X8_RC_PF0_PTM_RES_CAP_PTM_RES_NOM_CLOCK_T_OFF_0

Offset: 0x360

Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**RSVDP_24**: Reserved for future use.
23:16	0x0	**PTM_RES_NOM_CLOCK_T_INT**: PTM Responder Nominal Clock Period Integral - Integral part of the nominal PTM local clock period, in nanoseconds. For more details, see the PTM section in the Databook. Note: This register field is sticky.
15:0	0x0	**PTM_RES_NOM_CLOCK_T_FRAC**: PTM Responder Nominal Clock Period Fractional. Fractional part of the nominal PTM local clock period. LSB is 1/(2^16) nanoseconds. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PTM_RES_CAP_PTM_RES_SCALED_CLOCK_T_OFF_0

where <*i*> = 1, 4, 8.

Description: For more details, see the PTM section in the Databook.

PCIE_X1_RC_PF0_PTM_RES_CAP_PTM_RES_SCALED_CLOCK_T_OFF_0

Offset: 0x334

PCIE_X4_RC_PF0_PTM_RES_CAP_PTM_RES_SCALED_CLOCK_T_OFF_0

Offset: 0x350

PCIE_X8_RC_PF0_PTM_RES_CAP_PTM_RES_SCALED_CLOCK_T_OFF_0

Offset: 0x364

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PTM_RES_SCALED_CLOCK_T_EN**: PTM Responder Scaled Clock Period Enable. Use the programmed scaled PTM clock period rather than the nominal values. This bit is cleared when the core_clk rate starts to change and can only be set when the clock period change is complete. For more details, see the PTM section in the Databook. Note: This register field is sticky.

Bit	R/W	Reset	Description
30:24	RO	0x0	**RSVDP_24**: Reserved for future use.
23:16	RW	0x0	**PTM_RES_SCALED_CLOCK_T_INT**: PTM Responder Scaled Clock Period Integral. Integral part of the nominal PTM local clock period, in nanoseconds. For more details, see the PTM section in the Databook. Note: This register field is sticky.
15:0	RW	0x0	**PTM_RES_SCALED_CLOCK_T_FRAC**: PTM Responder Scaled Clock Period Fractional. Fractional part of the nominal PTM local clock period. LSB is 1/(2^16) nanoseconds. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PTM_RES_CAP_PTM_RES_LATENCY_REG_SEL_OFF_0

where <*i*> = 1, 4, 8.

Description: For more details, see the PTM section in the Databook.

PCIE_X1_RC_PF0_PTM_RES_CAP_PTM_RES_LATENCY_REG_SEL_OFF_0

Offset: 0x338

PCIE_X4_RC_PF0_PTM_RES_CAP_PTM_RES_LATENCY_REG_SEL_OFF_0

Offset: 0x354

PCIE_X8_RC_PF0_PTM_RES_CAP_PTM_RES_LATENCY_REG_SEL_OFF_0

Offset: 0x368

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:8	RO	0x0	**RSVDP_8**: Reserved for future use.
7	RW	0x0	**PTM_RES_ESM_SEL**: Selection Mode for PTM Responder Tx and Rx Latency Viewport register to be written or read. For more details, see the PTM section in the Databook. Note: This register field is sticky.

Bit	R/W	Reset	Description
6:4	RO	0x0	**RSVDP_4**: Reserved for future use.
3:0	RW	0x0	**PTM_RES_LATENCY_REG_SEL**: Selects the PTM Responder Tx and Rx Latency Viewport register to be written or read, together with the PTM_RES_ESM_SEL field. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_VSECDMA_CAP_VSECDMA_EXT_CAP_HDR_OFF_0

where <*i*> = 1, 4, 8.

Description: This register provides capability ID, capability version, and next capability offset for PCIe extended capability structure.

PCIE_X1_RC_PF0_VSECDMA_CAP_VSECDMA_EXT_CAP_HDR_OFF_0

Offset: 0x358

PCIE_X4_RC_PF0_VSECDMA_CAP_VSECDMA_EXT_CAP_HDR_OFF_0

Offset: 0x374

PCIE_X8_RC_PF0_VSECDMA_CAP_VSECDMA_EXT_CAP_HDR_OFF_0

Offset: 0x388

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0001000b (0b0000,0000,0000,0001,0000,0000,0000,1011)

Bit	Reset	Description
31:20	0x0	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. The bottom 2 bits of this offset are Reserved and must be implemented as 00b although software must mask them to allow for future uses of these bits. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

Bit	Reset	Description
19:16	0x1	**CAP**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. A version of the specification that changes the Extended Capability in a way that is not otherwise identifiable (that is, through a new Capability field) is permitted to increment this field. All such changes to the Capability structure must be software-compatible. Software must check for Capability Version numbers that are greater than or equal to the highest number defined when the software is written, as Functions reporting any such Capability Version numbers will contain a Capability structure that is compatible with that piece of software. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0xb	**ID**: PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_VSECDMA_CAP_VSECDMA_VENDOR_SPECIFIC_HDR_OFF_0

where <*i*> = 1, 4, 8.

Description: This Register provides VSEC Length, VSEC ID, and VSEC Rev(Version Number). Vendor-specific software must qualify the associated Vendor ID of the PCI Express Function or RCRB before attempting to interpret the values in the VSEC ID or VSEC Rev fields.

PCIE_X1_RC_PF0_VSECDMA_CAP_VSECDMA_VENDOR_SPECIFIC_HDR_OFF_0

Offset: 0x35c

PCIE_X4_RC_PF0_VSECDMA_CAP_VSECDMA_VENDOR_SPECIFIC_HDR_OFF_0

Offset: 0x378

PCIE_X8_RC_PF0_VSECDMA_CAP_VSECDMA_VENDOR_SPECIFIC_HDR_OFF_0

Offset: 0x38c

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x01800006 (0b0000,0001,1000,0000,0000,0000,0000,0110)

Bit	Reset	Description
31:20	0x18	**VSEC_LENGTH**: VSEC Length. This field indicates the number of bytes in the entire VSEC structure, including the Vendor-Specific Extended Capability Header, the Vendor-Specific Header, and the vendor-specific registers.
19:16	0x0	**VSEC_REV**: VSEC Rev. This field is a vendor-defined version number that indicates the version of the VSEC structure. Software must qualify the Vendor ID and VSEC ID before interpreting this field.
15:0	0x6	**VSEC_ID**: VSEC ID. This field is a vendor-defined ID number that indicates the nature and format of the VSEC structure. Software must qualify the Vendor ID before interpreting this field.

PCIE_X<*i*>_RC_PF0_VSECDMA_CAP_VSECDMA_DEVICE_INFORMATION_OFF_0

where <*i*> = 1, 4, 8.

Description: This register provides DMA and AXI Bridge implementation-specific information: - DMA Architecture: Legacy DMA or Hyper DMA (HDMA). - Register Location: Port-logic or Mapped to a Function and BAR. - Register Map: Legacy DMA or HDMA. - Channel Separation: Address distance between read and write channels. - AXI Bridge: Used or Not Used - AXI Master Bus Specification: AXI Master Bus Width, Burst Length, and Boundary Pointer Width You must use this information along with the IP version registers: - PCIE_VERSION_NUMBER_OFF - PCIE_VERSION_TYPE_OFF.

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_VSECDMA_CAP_VSECDMA_DEVICE_INFORMATION_OFF_0

Offset: 0x360
Reset: 0x35180000 (0b0011,0101,0001,1000,0000,0000,0000,0000)

Bit	Reset	Description
31:30	0x0	**RSVDP_30**: Reserved for future use.
29:26	0xd	**MASTER_PAGE_BOUNDARY_POINTER_WIDTH**: This field provides address page boundary information. It reports the value of CC_MSTR_PAGE_BOUNDARY_PW configuration parameter.
25:23	0x2	**MASTER_BURST_LENGTH**: Reports the CC_MSTR_BURST_LEN configuration parameter.

Bit	Reset	Description
22:20	0x1	**MASTER_BUS_WIDTH**: This field provides information regarding the AXI master data bus width. It reports the value of MASTER_BUS_DATA_WIDTH configuration parameter.
19	0x1	**AXI**: This field provides information about AXI interface usage. It reports the value of AXI_POPULATED configuration parameter.
18:16	0x0	**CHANNEL_SEPARATION**: If the MAP_FORMAT is set to HDMA_NATIVE, this field specifies the read write channel address separation. Other values are reserved.
15:11	0x0	**PFN**: Physical Function Number. This field provides information regarding the DMA register and physical function mapping.
10:8	0x0	**BARN**: Bar Number. This field provides information regarding the DMA register and BAR number mapping.
7:3	0x0	**RSVDP_3**: Reserved for future use.
2:0	0x0	**MAP_FORMAT**: Defines the register map format and features to be one of the following values: Other values are reserved.

PCIE_X4_RC_PF0_VSECDMA_CAP_VSECDMA_DEVICE_INFORMATION_OFF_0

Offset: 0x37c
Reset: 0x34a80401 (0b0011,0100,1010,1000,0000,0100,0000,0001)

Bit	Reset	Description
31:30	0x0	**RSVDP_30**: Reserved for future use.
29:26	0xd	**MASTER_PAGE_BOUNDARY_POINTER_WIDTH**: This field provides address page boundary information. It reports the value of CC_MSTR_PAGE_BOUNDARY_PW configuration parameter.
25:23	0x1	**MASTER_BURST_LENGTH**: Reports the CC_MSTR_BURST_LEN configuration parameter.
22:20	0x2	**MASTER_BUS_WIDTH**: This field provides information regarding the AXI master data bus width. It reports the value of MASTER_BUS_DATA_WIDTH configuration parameter.
19	0x1	**AXI**: This field provides information about AXI interface usage. It reports the value of AXI_POPULATED configuration parameter.
18:16	0x0	**CHANNEL_SEPARATION**: If the MAP_FORMAT is set to HDMA_NATIVE, this field specifies the read write channel address separation. Other values are reserved.

Bit	Reset	Description
15:11	0x0	**PFN**: Physical Function Number. This field provides information regarding the DMA register and physical function mapping.
10:8	0x4	**BARN**: Bar Number. This field provides information regarding the DMA register and BAR number mapping.
7:3	0x0	**RSVDP_3**: Reserved for future use.
2:0	0x1	**MAP_FORMAT**: Defines the register map format and features to be one of the following values: Other values are reserved.

PCIE_X8_RC_PF0_VSECDMA_CAP_VSECDMA_DEVICE_INFORMATION_OFF_0

Offset: 0x390
Reset: 0x34380401 (0b0011,0100,0011,1000,0000,0100,0000,0001)

Bit	Reset	Description
31:30	0x0	**RSVDP_30**: Reserved for future use.
29:26	0xd	**MASTER_PAGE_BOUNDARY_POINTER_WIDTH**: This field provides address page boundary information. It reports the value of CC_MSTR_PAGE_BOUNDARY_PW configuration parameter.
25:23	0x0	**MASTER_BURST_LENGTH**: Reports the CC_MSTR_BURST_LEN configuration parameter.
22:20	0x3	**MASTER_BUS_WIDTH**: This field provides information regarding the AXI master data bus width. It reports the value of MASTER_BUS_DATA_WIDTH configuration parameter.
19	0x1	**AXI**: This field provides information about AXI interface usage. It reports the value of AXI_POPULATED configuration parameter.
18:16	0x0	**CHANNEL_SEPARATION**: If the MAP_FORMAT is set to HDMA_NATIVE, this field specifies the read write channel address separation. Other values are reserved.
15:11	0x0	**PFN**: Physical Function Number. This field provides information regarding the DMA register and physical function mapping.
10:8	0x4	**BARN**: Bar Number. This field provides information regarding the DMA register and BAR number mapping.
7:3	0x0	**RSVDP_3**: Reserved for future use.
2:0	0x1	**MAP_FORMAT**: Defines the register map format and features to be one of the following values: Other values are reserved.

PCIE_X<*i*>_RC_PF0_VSECDMA_CAP_VSECDMA_NUM_CHAN_OFF_0

where <*i*> = 1, 4, 8.

Description: This register specifies the number of read and write channels implemented.

PCIE_X1_RC_PF0_VSECDMA_CAP_VSECDMA_NUM_CHAN_OFF_0

Offset: 0x364

PCIE_X4_RC_PF0_VSECDMA_CAP_VSECDMA_NUM_CHAN_OFF_0

Offset: 0x380

PCIE_X8_RC_PF0_VSECDMA_CAP_VSECDMA_NUM_CHAN_OFF_0

Offset: 0x394

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00020004 (0b0000,0000,0000,0010,0000,0000,0000,0100)

Bit	Reset	Description
31:26	0x0	**RSVDP_26**: Reserved for future use.
25:16	0x2	**NUM_DMA_RD_CHAN**: This field provides information regarding the number of implemented read channels. It reports the value of CC_NUM_DMA_RD_CHAN parameter.
15:10	0x0	**RSVDP_10**: Reserved for future use.
9:0	0x4	**NUM_DMA_WR_CHAN**: This field provides information regarding the number of implemented write channels. It reports the value of CC_NUM_DMA_WR_CHAN parameter.

PCIE_X<*i*>_RC_PF0_VSECDMA_CAP_VSECDMA_UNROLL_ADDR_OFFSET_LOW_OFF_0

where <*i*> = 1, 4, 8.

Description: This register specifies the lower 32 bits of the offset of the start address of the DMA register map. Applicable only if MAP_FORMAT >0, that is, all map formats other than EDMA_LEGACY_PL.

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_VSECDMA_CAP_VSECDMA_UNROLL_ADDR_OFFSET_LOW_OFF_0

Offset: 0x368
Reset: 0x00001000 (0b0000,0000,0000,0000,0001,0000,0000,0000)

Bit	Reset	Description
31:0	0x1000	**UNROLL_ADDR_OFFSET_LOW**: BAR address offset, 32-bit LSB.

PCIE_X4_RC_PF0_VSECDMA_CAP_VSECDMA_UNROLL_ADDR_OFFSET_LOW_OFF_0

Offset: 0x384

PCIE_X8_RC_PF0_VSECDMA_CAP_VSECDMA_UNROLL_ADDR_OFFSET_LOW_OFF_0

Offset: 0x398
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UNROLL_ADDR_OFFSET_LOW**: BAR address offset, 32-bit LSB.

PCIE_X<*i*>_RC_PF0_VSECDMA_CAP_VSECDMA_UNROLL_ADDR_OFFSET_HIGH_OFF_0

where <*i*> = 1, 4, 8.

Description: This register specifies the upper 32 bits of the offset of the start address of the DMA register map. Applicable only if MAP_FORMAT >0, that is, all map formats other than EDMA_LEGACY_PL.

PCIE_X1_RC_PF0_VSECDMA_CAP_VSECDMA_UNROLL_ADDR_OFFSET_HIGH_OFF_0

Offset: 0x36c

PCIE_X4_RC_PF0_VSECDMA_CAP_VSECDMA_UNROLL_ADDR_OFFSET_HIGH_OFF_0

Offset: 0x388

PCIE_X8_RC_PF0_VSECDMA_CAP_VSECDMA_UNROLL_ADDR_OFFSET_HIGH_OFF_0

Offset: 0x39c

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UNROLL_ADDR_OFFSET_HIGH**: BAR address offset, 32-bit MSB.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_ACK_LATENCY_TIMER_OFF_0

where <*i*> = 1, 4, 8.

Description: This register holds the ack latency timer limit and replay timer limit values.

PCIE_X1_RC_PF0_PORT_LOGIC_ACK_LATENCY_TIMER_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_ACK_LATENCY_TIMER_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_ACK_LATENCY_TIMER_OFF_0

Offset: 0x700
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0c23040b (0b0000,1100,0010,0011,0000,0100,0000,1011)

Bit	Reset	Description
31:16	0xc23	**REPLAY_TIME_LIMIT**: Replay Timer Limit. The replay timer expires when it reaches this limit. The controller initiates a replay upon reception of a NAK or when the replay timer expires. For more details, see "Transmit Replay" in the Databook. - You can modify the effective timer limit through the TIMER_MOD_REPLAY_TIMER field of the TIMER_CTRL_MAX_FUNC_NUM_OFF register. - After reset, the controller updates the default according to the Negotiated Link Width, Max_Payload_Size, and speed. The value is determined from Tables 3-4, 3-5, and 3-6 of the PCI Express Base Specification. - If there is a change in the payload size or link speed, the controller overrides any value that you have written to this register field, and resets the field back to the specification-defined value. The controller does not change the value in the TIMER_MOD_REPLAY_TIMER field of the TIMER_CTRL_MAX_FUNC_NUM_OFF register.

Bit	Reset	Description
15:0	0x40b	**ROUND_TRIP_LATENCY_TIME_LIMIT**: Ack Latency Timer Limit. The Ack latency timer expires when it reaches this limit. For more details, see "ACK/NAK Scheduling" in the Databook. - You can modify the effective timer limit through the TIMER_MOD_ACK_NAK field of the TIMER_CTRL_MAX_FUNC_NUM_OFF register. - After reset, the controller updates the default according to the Negotiated Link Width, Max_Payload_Size, and speed. The value is determined from Tables 3-7, 3-8, and 3-9 of the PCI Express Base Specification. - The limit must reflect the round trip latency from requester to completer. - If there is a change in the payload size or link width, the controller overrides any value that you have written to this register field, and resets the field back to the specification-defined value. The controller does not change the value in the TIMER_MOD_ACK_NAK field of the TIMER_CTRL_MAX_FUNC_NUM_OFF register.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_VENDOR_SPEC_DLLP_OFF_0

where <*i*> = 1, 4, 8.

Description: This register holds the vendor specific DLLP.

PCIE_X1_RC_PF0_PORT_LOGIC_VENDOR_SPEC_DLLP_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_VENDOR_SPEC_DLLP_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_VENDOR_SPEC_DLLP_OFF_0

Offset: 0x704
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffffffff	**VENDOR_SPEC_DLLP**: Vendor Specific DLLP Register. You can use this register to send a specific PCI Express DLLP. Your application can write 8-bit DLLP Type and 24-bit Payload data into this register, and set the VENDOR_SPECIFIC_DLLP_REQ field of the PORT_LINK_CTRL_OFF, to send the DLLP. - Bits[7:0]: DLLP Type - Bits[31:8]: Vendor Defined Payload (24 bits) Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_PORT_FORCE_OFF_0

where <*i*> = 1, 4, 8.

Description: This register can be used for testing and debuggong the link.

PCIE_X1_RC_PF0_PORT_LOGIC_PORT_FORCE_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_PORT_FORCE_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_PORT_FORCE_OFF_0

Offset: 0x708
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00400000 (0b0000,0000,0100,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**RSVDP_24**: Reserved for future use.
23	RW	0x0	**DO_DESKEW_FOR_SRIS**: Use the transitions from TS2 to Logical Idle Symbol, SKP OS to Logical Idle Symbol, EIEOS to Logical Idle Symbol, and FTS Sequence to SKP OS to do deskew instead of using received SKP OS or TS1 to TS2 transition if DO_DESKEW_FOR_SRIS is set to '1'. Note: This register field is sticky.
22	RW	0x1	**SUPPORT_PART_LANES_RXEI_EXIT**: Support LTSSM transition from Polling.Active to Polling.Config based on Rx 8 TSs on any lanes which are Rx EI exit too from base spec after 24ms timeout. This prevents some lanes detected but not Rx EI exit and LTSSM cannot move to Polling.Config. You must set the parameter CX_AUTO_LANE_FLIP_CTRL_EN true for the auto lanes reversal. Note: This register field is sticky.
21:16	RW	0x0	**LINK_STATE**: Forced LTSSM State. The LTSSM state that the controller is forced to when you set the FORCE_EN bit (Force Link). LTSSM state encoding is defined by the lts_state variable in workspace/src/Layer1/smlh_ltssm.v. Note: This register field is sticky.
15	WO	0x0	**FORCE_EN**: Force Link. The controller supports a testing and debug capability to allow your software to force the LTSSM state machine into a specific state, and to force the controller to transmit a specific Link Command. Asserting this bit triggers the following actions: - Forces the LTSSM to the state specified by the Forced LTSSM State field. - Forces the controller to transmit the command specified by the Forced Link Command field. This is a self-clearing register field. Reading from this register field always returns a '0'.
14:12	RO	0x0	**RSVDP_12**: Reserved for future use.
11:8	RW	0x0	**FORCED_LTSSM**: Forced Link Command. The link command that the controller is forced to transmit when you set FORCE_EN bit (Force Link). Link command encoding is defined by the ltssm_cmd variable in workspace/src/Layer1/smlh_ltssm.v. Note: This register field is sticky.
7:0	RW	0x0	**LINK_NUM**: Link Number. Not used for endpoint. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_ACK_F_ASPM_CTRL_OFF_0

where <*i*> = 1, 4, 8.

Description: This register is used to control ack frequency and L0-L1 ASPM behaviour.

Offset: 0x70c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_PORT_LOGIC_ACK_F_ASPM_CTRL_OFF_0

Reset: 0x1b343400 (0b0001,1011,0011,0100,0011,0100,0000,0000)

Bit	R/W	Reset	Description
31	RO	0x0	**RSVDP_31**: Reserved for future use.
30	RW	0x0	**ENTER_ASPM**: ASPM L1 Entry Control. Note: This register field is sticky.
29:27	RW	0x3	**L1_ENTRANCE_LATENCY**: L1 Entrance Latency. Note: Programming this timer with a value greater that 32us has no effect unless extended sync is used, or all of the credits are infinite. Note: This register field is sticky.
26:24	RW	0x3	**L0S_ENTRANCE_LATENCY**: L0s Entrance Latency. Note: This register field is sticky.
23:16	RO	0x34	**COMMON_CLK_N_FTS**: Common Clock N_FTS. This is the N_FTS when common clock is used. The number of Fast Training Sequence ordered sets to be transmitted when transitioning from L0s to L0. This field is only writable (sticky) when all of the following configuration parameter equations are true: - CX_NFTS !=CX_COMM_NFTS - DEFAULT_L0S_EXIT_LATENCY ! =DEFAULT_COMM_L0S_EXIT_LATENCY - DEFAULT_L1_EXIT_LATENCY ! =DEFAULT_COMM_L1_EXIT_LATENCY Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R
15:8	RW	0x34	**ACK_N_FTS**: The number of Fast Training Sequence(N_FTS) ordered sets to be transmitted when transitioning from L0s to L0. Note: This register field is sticky.

Bit	R/W	Reset	Description
7:0	RW	0x0	**ACK_FREQ**: Ack Frequency. The controller accumulates the number of pending ACKs specified here (up to 255) before scheduling an ACK DLLP. - 0: Indicates that this Ack Frequency Counter feature is turned off. The controller generates a low-priority ACK request for every TLP that it receives. The controller waits until the ACK Latency Timer expires, then converts the current low-priority ACK request to a high-priority ACK request and schedules the DLLP for transmission to the remote link partner. - 1-255: Indicates that the controller will schedule a high-priority ACK after receiving this number of TLPs. It might schedule the ACK before receiving this number of TLPs if the ACK Latency Timer expires, but never later. For a typical system, you do not have to modify the default setting. For more details, see "ACK/NAK Scheduling" in the Databook. Note: This register field is sticky.

PCIE_X4_RC_PF0_PORT_LOGIC_ACK_F_ASPM_CTRL_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_ACK_F_ASPM_CTRL_OFF_0

Reset: 0x23343400 (0b0010,0011,0011,0100,0011,0100,0000,0000)

Bit	R/W	Reset	Description
31	RO	0x0	**RSVDP_31**: Reserved for future use.
30	RW	0x0	**ENTER_ASPM**: ASPM L1 Entry Control. Note: This register field is sticky.
29:27	RW	0x4	**L1_ENTRANCE_LATENCY**: L1 Entrance Latency. Note: Programming this timer with a value greater that 32us has no effect unless extended sync is used, or all of the credits are infinite. Note: This register field is sticky.
26:24	RW	0x3	**L0S_ENTRANCE_LATENCY**: L0s Entrance Latency. Note: This register field is sticky.
23:16	RO	0x34	**COMMON_CLK_N_FTS**: Common Clock N_FTS. This is the N_FTS when common clock is used. The number of Fast Training Sequence ordered sets to be transmitted when transitioning from L0s to L0. This field is only writable (sticky) when all of the following configuration parameter equations are true: - CX_NFTS !=CX_COMM_NFTS - DEFAULT_L0S_EXIT_LATENCY ! =DEFAULT_COMM_L0S_EXIT_LATENCY - DEFAULT_L1_EXIT_LATENCY ! =DEFAULT_COMM_L1_EXIT_LATENCY Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R
15:8	RW	0x34	**ACK_N_FTS**: The number of Fast Training Sequence(N_FTS) ordered sets to be transmitted when transitioning from L0s to L0. Note: This register field is sticky.

Bit	R/W	Reset	Description
7:0	RW	0x0	**ACK_FREQ**: Ack Frequency. The controller accumulates the number of pending ACKs specified here (up to 255) before scheduling an ACK DLLP. - 0: Indicates that this Ack Frequency Counter feature is turned off. The controller generates a low-priority ACK request for every TLP that it receives. The controller waits until the ACK Latency Timer expires, then converts the current low-priority ACK request to a high-priority ACK request and schedules the DLLP for transmission to the remote link partner. - 1-255: Indicates that the controller will schedule a high-priority ACK after receiving this number of TLPs. It might schedule the ACK before receiving this number of TLPs if the ACK Latency Timer expires, but never later. For a typical system, you do not have to modify the default setting. For more details, see "ACK/NAK Scheduling" in the Databook. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_PORT_LINK_CTRL_OFF_0

where <*i*> = 1, 4, 8.

Description: Using this register you can control the port link behaviour.

Offset: 0x710
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_PORT_LOGIC_PORT_LINK_CTRL_OFF_0

Reset: 0x00010120 (0b0000,0000,xx00,0001,0000,0001,0010,0000)

Bit	R/W	Reset	Description
31:28	RO	0x0	**RSVDP_28**: Reserved for future use.
27	RW	0x0	**TRANSMIT_LANE_REVERSALE_ENABLE**: TRANSMIT_LANE_REVERSALE_ENABLE is an internally reserved field. Do not use. Note: This register field is sticky.
26	RW	0x0	**EXTENDED_SYNCH**: EXTENDED_SYNCH is an internally reserved field. Do not use. Note: This register field is sticky.
25	RW	0x0	**CORRUPT_LCRC_ENABLE**: CORRUPT_LCRC_ENABLE is an internally reserved field. Do not use. Note: This register field is sticky.
24	RW	0x0	**BEACON_ENABLE**: BEACON_ENABLE is an internally reserved field. Do not use. Note: This register field is sticky.

Bit	R/W	Reset	Description
21:16	RW	0x1	**LINK_CAPABLE**: Link Mode Enable. Sets the number of lanes in the link that you want to connect to the link partner. When you have unused lanes in your system, then you must change the value in this register to reflect the number of lanes. You must also change the value in the "Predetermined Number of Lanes" field of the "Link Width and Speed Change Control Register". For more information, see "How to Tie Off Unused Lanes". For information on upsizing and downsizing the link width, see "Link Establishment". Note: This register field is sticky.
15:12	RO	0x0	**RSVDP_12**: Reserved for future use.
11:8	RW	0x1	**LINK_RATE**: LINK_RATE is an internally reserved field. Do not use. Note: This register field is sticky.
7	RW	0x0	**FAST_LINK_MODE**: Fast Link Mode. Sets all internal LTSSM millisecond timers to Fast Mode for speeding up simulation. Forces the LTSSM training (link initialization) to use shorter time-outs and to link up faster. - The default scaling factor can be changed using the DEFAULT_FAST_LINK_SCALING_FACTOR parameter or through the FAST_LINK_SCALING_FACTOR field in the TIMER_CTRL_MAX_FUNC_NUM_OFF register. - Fast Link Mode can also be activated by setting the diag_ctrl_bus[2] pin to '1'. For more details, see the "Fast Link Simulation Mode" section in the "Integrating the Controller with the PHY or Application RTL or Verification IP" chapter of the User Guide. Note: This register field is sticky.
6	RW	0x0	**LINK_DISABLE**: LINK_DISABLE is an internally reserved field. Do not use. Note: This register field is sticky.
5	RW	0x1	**DLL_LINK_EN**: DLL Link Enable. Note: This register field is sticky.
4	RO	0x0	**RSVDP_4**: Reserved for future use.
3	RW	0x0	**RESET_ASSERT**: Reset Assert. Triggers a recovery and forces the LTSSM to the hot reset state (downstream port only). Note: This register field is sticky.
2	RW	0x0	**LOOPBACK_ENABLE**: Loopback Enable. Turns on loopback. For more details, see "Loopback" in the Databook. Note: This register field is sticky.
1	RW	0x0	**SCRAMBLE_DISABLE**: Scramble Disable. Turns off data scrambling. Note: This register field is sticky.
0	RW	0x0	**VENDOR_SPECIFIC_DLLP_REQ**: Vendor Specific DLLP Request. Reading from this self-clearing register field always returns a '0'.

PCIE_X4_RC_PF0_PORT_LOGIC_PORT_LINK_CTRL_OFF_0

Reset: 0x00070120 (0b0000,0000,xx00,0111,0000,0001,0010,0000)

Bit	R/W	Reset	Description
31:28	RO	0x0	**RSVDP_28**: Reserved for future use.
27	RW	0x0	**TRANSMIT_LANE_REVERSALE_ENABLE**: TRANSMIT_LANE_REVERSALE_ENABLE is an internally reserved field. Do not use. Note: This register field is sticky.
26	RW	0x0	**EXTENDED_SYNCH**: EXTENDED_SYNCH is an internally reserved field. Do not use. Note: This register field is sticky.
25	RW	0x0	**CORRUPT_LCRC_ENABLE**: CORRUPT_LCRC_ENABLE is an internally reserved field. Do not use. Note: This register field is sticky.
24	RW	0x0	**BEACON_ENABLE**: BEACON_ENABLE is an internally reserved field. Do not use. Note: This register field is sticky.
21:16	RW	0x7	**LINK_CAPABLE**: Link Mode Enable. Sets the number of lanes in the link that you want to connect to the link partner. When you have unused lanes in your system, then you must change the value in this register to reflect the number of lanes. You must also change the value in the "Predetermined Number of Lanes" field of the "Link Width and Speed Change Control Register". For more information, see "How to Tie Off Unused Lanes". For information on upsizing and downsizing the link width, see "Link Establishment". Note: This register field is sticky.
15:12	RO	0x0	**RSVDP_12**: Reserved for future use.
11:8	RW	0x1	**LINK_RATE**: LINK_RATE is an internally reserved field. Do not use. Note: This register field is sticky.
7	RW	0x0	**FAST_LINK_MODE**: Fast Link Mode. Sets all internal LTSSM millisecond timers to Fast Mode for speeding up simulation. Forces the LTSSM training (link initialization) to use shorter time-outs and to link up faster. - The default scaling factor can be changed using the DEFAULT_FAST_LINK_SCALING_FACTOR parameter or through the FAST_LINK_SCALING_FACTOR field in the TIMER_CTRL_MAX_FUNC_NUM_OFF register. - Fast Link Mode can also be activated by setting the diag_ctrl_bus[2] pin to '1'. For more details, see the "Fast Link Simulation Mode" section in the "Integrating the Controller with the PHY or Application RTL or Verification IP" chapter of the User Guide. Note: This register field is sticky.
6	RW	0x0	**LINK_DISABLE**: LINK_DISABLE is an internally reserved field. Do not use. Note: This register field is sticky.
5	RW	0x1	**DLL_LINK_EN**: DLL Link Enable. Note: This register field is sticky.
4	RO	0x0	**RSVDP_4**: Reserved for future use.
3	RW	0x0	**RESET_ASSERT**: Reset Assert. Triggers a recovery and forces the LTSSM to the hot reset state (downstream port only). Note: This register field is sticky.

Bit	R/W	Reset	Description
2	RW	0x0	**LOOPBACK_ENABLE**: Loopback Enable. Turns on loopback. For more details, see "Loopback" in the Databook. Note: This register field is sticky.
1	RW	0x0	**SCRAMBLE_DISABLE**: Scramble Disable. Turns off data scrambling. Note: This register field is sticky.
0	RW	0x0	**VENDOR_SPECIFIC_DLLP_REQ**: Vendor Specific DLLP Request. Reading from this self-clearing register field always returns a '0'.

PCIE_X8_RC_PF0_PORT_LOGIC_PORT_LINK_CTRL_OFF_0

Reset: 0x000f0120 (0b0000,0000,xx00,1111,0000,0001,0010,0000)

Bit	R/W	Reset	Description
31:28	RO	0x0	**RSVDP_28**: Reserved for future use.
27	RW	0x0	**TRANSMIT_LANE_REVERSALE_ENABLE**: TRANSMIT_LANE_REVERSALE_ENABLE is an internally reserved field. Do not use. Note: This register field is sticky.
26	RW	0x0	**EXTENDED_SYNCH**: EXTENDED_SYNCH is an internally reserved field. Do not use. Note: This register field is sticky.
25	RW	0x0	**CORRUPT_LCRC_ENABLE**: CORRUPT_LCRC_ENABLE is an internally reserved field. Do not use. Note: This register field is sticky.
24	RW	0x0	**BEACON_ENABLE**: BEACON_ENABLE is an internally reserved field. Do not use. Note: This register field is sticky.
21:16	RW	0xf	**LINK_CAPABLE**: Link Mode Enable. Sets the number of lanes in the link that you want to connect to the link partner. When you have unused lanes in your system, then you must change the value in this register to reflect the number of lanes. You must also change the value in the "Predetermined Number of Lanes" field of the "Link Width and Speed Change Control Register". For more information, see "How to Tie Off Unused Lanes". For information on upsizing and downsizing the link width, see "Link Establishment". Note: This register field is sticky.
15:12	RO	0x0	**RSVDP_12**: Reserved for future use.
11:8	RW	0x1	**LINK_RATE**: LINK_RATE is an internally reserved field. Do not use. Note: This register field is sticky.

Bit	R/W	Reset	Description
7	RW	0x0	**FAST_LINK_MODE**: Fast Link Mode. Sets all internal LTSSM millisecond timers to Fast Mode for speeding up simulation. Forces the LTSSM training (link initialization) to use shorter time-outs and to link up faster. - The default scaling factor can be changed using the DEFAULT_FAST_LINK_SCALING_FACTOR parameter or through the FAST_LINK_SCALING_FACTOR field in the TIMER_CTRL_MAX_FUNC_NUM_OFF register. - Fast Link Mode can also be activated by setting the diag_ctrl_bus[2] pin to '1'. For more details, see the "Fast Link Simulation Mode" section in the "Integrating the Controller with the PHY or Application RTL or Verification IP" chapter of the User Guide. Note: This register field is sticky.
6	RW	0x0	**LINK_DISABLE**: LINK_DISABLE is an internally reserved field. Do not use. Note: This register field is sticky.
5	RW	0x1	**DLL_LINK_EN**: DLL Link Enable. Note: This register field is sticky.
4	RO	0x0	**RSVDP_4**: Reserved for future use.
3	RW	0x0	**RESET_ASSERT**: Reset Assert. Triggers a recovery and forces the LTSSM to the hot reset state (downstream port only). Note: This register field is sticky.
2	RW	0x0	**LOOPBACK_ENABLE**: Loopback Enable. Turns on loopback. For more details, see "Loopback" in the Databook. Note: This register field is sticky.
1	RW	0x0	**SCRAMBLE_DISABLE**: Scramble Disable. Turns off data scrambling. Note: This register field is sticky.
0	RW	0x0	**VENDOR_SPECIFIC_DLLP_REQ**: Vendor Specific DLLP Request. Reading from this self-clearing register field always returns a '0'.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_LANE_SKEW_OFF_0

where <*i*> = 1, 4, 8.

Description: This register is used to control the lane skew behaviour.

Offset: 0x714
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_PORT_LOGIC_LANE_SKEW_OFF_0

Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**DISABLE_LANE_TO_LANE_DESKEW**: Disable Lane-to-Lane Deskew. Causes the controller to disable the internal Lane-to-Lane deskew logic. Note: This register field is sticky.
30:27	0x0	**IMPLEMENT_NUM_LANES**: Implementation-specific Number of Lanes. Set the implementation-specific number of lanes. The number of lanes to be used when in Loopback Master. The number of lanes programmed must be equal to or less than the valid number of lanes set in LINK_CAPABLE field. You must configure this field before initiating Loopback by writing in the LOOPBACK_ENABLE field. The controller will transition from Loopback.Entry to Loopback.Active after receiving two consecutive TS1 Ordered Sets with the Loopback bit asserted on the implementation specific number of lanes configured in this field. Note: This register field is sticky.
26	0x0	**ELASTIC_BUFFER_MODE**: Selects Elasticity Buffer operating mode: Note: This register field is sticky.
25	0x0	**ACK_NAK_DISABLE**: Ack/Nak Disable. Prevents the controller from sending ACK and NAK DLLPs. Note: This register field is sticky.
24	0x0	**FLOW_CTRL_DISABLE**: Flow Control Disable. Prevents the controller from sending FC DLLPs. Note: This register field is sticky.
23:0	0x0	**INSERT_LANE_SKEW**: INSERT_LANE_SKEW is an internally reserved field. Do not use. Note: This register field is sticky.

PCIE_X4_RC_PF0_PORT_LOGIC_LANE_SKEW_OFF_0

Reset: 0x18000000 (0b0001,1000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**DISABLE_LANE_TO_LANE_DESKEW**: Disable Lane-to-Lane Deskew. Causes the controller to disable the internal Lane-to-Lane deskew logic. Note: This register field is sticky.
30:27	0x3	**IMPLEMENT_NUM_LANES**: Implementation-specific Number of Lanes. Set the implementation-specific number of lanes. The number of lanes to be used when in Loopback Master. The number of lanes programmed must be equal to or less than the valid number of lanes set in LINK_CAPABLE field. You must configure this field before initiating Loopback by writing in the LOOPBACK_ENABLE field. The controller will transition from Loopback.Entry to Loopback.Active after receiving two consecutive TS1 Ordered Sets with the Loopback bit asserted on the implementation specific number of lanes configured in this field. Note: This register field is sticky.
26	0x0	**ELASTIC_BUFFER_MODE**: Selects Elasticity Buffer operating mode: Note: This register field is sticky.
25	0x0	**ACK_NAK_DISABLE**: Ack/Nak Disable. Prevents the controller from sending ACK and NAK DLLPs. Note: This register field is sticky.

Bit	Reset	Description
24	0x0	**FLOW_CTRL_DISABLE**: Flow Control Disable. Prevents the controller from sending FC DLLPs. Note: This register field is sticky.
23:0	0x0	**INSERT_LANE_SKEW**: INSERT_LANE_SKEW is an internally reserved field. Do not use. Note: This register field is sticky.

PCIE_X8_RC_PF0_PORT_LOGIC_LANE_SKEW_OFF_0

Reset: 0x38000000 (0b0011,1000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**DISABLE_LANE_TO_LANE_DESKEW**: Disable Lane-to-Lane Deskew. Causes the controller to disable the internal Lane-to-Lane deskew logic. Note: This register field is sticky.
30:27	0x7	**IMPLEMENT_NUM_LANES**: Implementation-specific Number of Lanes. Set the implementation-specific number of lanes. The number of lanes to be used when in Loopback Master. The number of lanes programmed must be equal to or less than the valid number of lanes set in LINK_CAPABLE field. You must configure this field before initiating Loopback by writing in the LOOPBACK_ENABLE field. The controller will transition from Loopback.Entry to Loopback.Active after receiving two consecutive TS1 Ordered Sets with the Loopback bit asserted on the implementation specific number of lanes configured in this field. Note: This register field is sticky.
26	0x0	**ELASTIC_BUFFER_MODE**: Selects Elasticity Buffer operating mode: Note: This register field is sticky.
25	0x0	**ACK_NAK_DISABLE**: Ack/Nak Disable. Prevents the controller from sending ACK and NAK DLLPs. Note: This register field is sticky.
24	0x0	**FLOW_CTRL_DISABLE**: Flow Control Disable. Prevents the controller from sending FC DLLPs. Note: This register field is sticky.
23:0	0x0	**INSERT_LANE_SKEW**: INSERT_LANE_SKEW is an internally reserved field. Do not use. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_TIMER_CTRL_MAX_FUNC_NUM_OFF_0

where <*i*> = 1, 4, 8.

Description: This register holds the ack frequency, latency, replay, fast link scaling timers, and max function number values.

PCIE_X1_RC_PF0_PORT_LOGIC_TIMER_CTRL_MAX_FUNC_NUM_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_TIMER_CTRL_MAX_FUNC_NUM_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_TIMER_CTRL_MAX_FUNC_NUM_OFF_0

Offset: 0x718
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00010000 (0b0000,0000,0000,0001,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RO	0x0	**RSVDP_31**: Reserved for future use.
30:29	RW	0x0	**FAST_LINK_SCALING_FACTOR**: Fast Link Timer Scaling Factor. Sets the scaling factor of LTSSM timer when FAST_LINK_MODE field in PORT_LINK_CTRL_OFF is set to '1'. Default is set by the hidden configuration parameter DEFAULT_FAST_LINK_SCALING_FACTOR which defaults to '0'. Note: This register field is sticky.
28:24	RW	0x0	**UPDATE_FREQ_TIMER**: UPDATE_FREQ_TIMER is an internally reserved field. Do not use. Note: This register field is sticky.
23:19	RW	0x0	**TIMER_MOD_ACK_NAK**: Ack Latency Timer Modifier. Increases the timer value for the Ack latency timer in increments of 64 clock cycles. A value of '0' represents no modification to the timer value. For more details, see the ROUND_TRIP_LATENCY_TIME_LIMIT field of the ACK_LATENCY_TIMER_OFF register. Note: This register field is sticky.
18:14	RW	0x4	**TIMER_MOD_REPLAY_TIMER**: Replay Timer Limit Modifier. Increases the time-out value for the replay timer in increments of 64 clock cycles at Gen1 or Gen2 speed, and in increments of 256 clock cycles at Gen3 speed. A value of '0' represents no modification to the timer limit. For more details, see the REPLAY_TIME_LIMIT field of the ACK_LATENCY_TIMER_OFF register. At Gen3 speed, the controller automatically changes the value of this field to DEFAULT_GEN3_REPLAY_ADJ. Note: This register field is sticky.
13:8	RO	0x0	**RSVDP_8**: Reserved for future use.
7:0	RW	0x0	**MAX_FUNC_NUM**: Maximum function number that can be used in a request. Configuration requests targeted at function numbers above this value are returned with UR (unsupported request). Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_SYMBOL_TIMER_FILTER_1_OFF_0

where <*i*> = 1, 4, 8.

Description: The Filter Mask 1 Register modifies the RADM filtering and error handling rules. For more details, see the "Receive Filtering" section. In each case, '0' applies the associated filtering rule and '1' masks the associated filtering rule.

PCIE_X1_RC_PF0_PORT_LOGIC_SYMBOL_TIMER_FILTER_1_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_SYMBOL_TIMER_FILTER_1_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_SYMBOL_TIMER_FILTER_1_OFF_0

Offset: 0x71c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000140 (0b0000,0000,0000,0000,0000,0001,0100,0000)

Bit	Reset	Description
31:16	0x0	**MASK_RADM_1**: Filter Mask 1. The Filter Mask 1 Register modifies the RADM filtering and error handling rules. For more details, see the "Receive Filtering" section. In each case, '0' applies the associated filtering rule and '1' masks the associated filtering rule. [31]: CX_FLT_MASK_RC_CFG_DISCARD - 0: For RADM RC filter to not allow CFG transaction being received - 1: For RADM RC filter to allow CFG transaction being received [30]: CX_FLT_MASK_RC_IO_DISCARD - 0: For RADM RC filter to not allow IO transaction being received - 1: For RADM RC filter to allow IO transaction being received [29]: CX_FLT_MASK_MSG_DROP - 0: Drop MSG TLP (except for Vendor MSG). Send decoded message on the SII. - 1: Do not Drop MSG (except for Vendor MSG). Send message TLPs to your application on TRGT1 and send decoded message on the SII. - The default for this bit is the inverse of FLT_DROP_MSG. That is, if FLT_DROP_MSG =1, then the default of this bit is '0' (drop message TLPs). This bit only controls message TLPs other than Vendor MSGs. Vendor MSGs are controlled by Filter Mask Register 2, bits [1:0]. The controller never passes ATS Invalidate messages to the SII interface regardless of this filter rule setting. The controller passes all ATS Invalidate messages to TRGT1 (or AXI bridge master), as they are too big for the SII. [28]: CX_FLT_MASK_CPL_ECRC_DISCARD - Only used when completion queue is advertised with infinite credits and is in store-and-forward mode. - 0: Discard completions with ECRC errors - 1: Allow completions with ECRC errors to be passed up - Reserved field for SW. [27]: CX_FLT_MASK_ECRC_DISCARD - 0: Discard TLPs with ECRC errors - 1: Allow TLPs with ECRC errors to be passed up [26]: CX_FLT_MASK_CPL_LEN_MATCH - 0: Enforce length match for completions; a violation results in cpl_abort, and possibly AER of unexp_cpl_err - 1: Mask length match for completions [25]: CX_FLT_MASK_CPL_ATTR_MATCH - 0: Enforce attribute match for completions; a violation results in a malformed TLP error, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca - 1: Mask attribute match for completions [24]: CX_FLT_MASK_CPL_TC_MATCH - 0: Enforce Traffic Class match for completions; a violation results in a malformed TLP error, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca - 1: Mask Traffic Class match for completions [23]: CX_FLT_MASK_CPL_FUNC_MATCH - 0: Enforce function match for completions; a violation results in cpl_abort, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca - 1: Mask function match for completions [22]: CX_FLT_MASK_CPL_REQID_MATCH - 0: Enforce Req. Id match for completions; a violation result in cpl_abort, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca - 1: Mask Req. Id match for completions [21]: CX_FLT_MASK_CPL_TAGERR_MATCH - 0: Enforce Tag Error Rules for completions; a violation result in cpl_abort, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca - 1: Mask Tag Error Rules for completions [20]: CX_FLT_MASK_LOCKED_RD_AS_UR - 0: Treat locked Read TLPs as UR for EP; Supported for RC - 1: Treat locked Read TLPs as Supported for EP; UR for RC [19]: CX_FLT_MASK_CFG_TYPE1_REQ_AS_UR - 0: Treat CFG type1 TLPs as UR for EP; Supported for RC - 1: Treat CFG type1 TLPs as Supported for EP; UR for RC - When CX_SRIOV_ENABLE is set then this bit is set to allow the filter to process Type 1 Config requests if the EP consumes more than one bus number. [18]: CX_FLT_MASK_UR_OUTSIDE_BAR - 0: Treat out-of-bar TLPs as UR - 1: Do not treat out-of-bar TLPs as UR [17]: CX_FLT_MASK_UR_POIS - 0: Treat poisoned request TLPs as UR - 1: Do not treat poisoned request TLPs as UR - The native controller always passes poisoned completions to your application except when you are using the DMA read channel. [16]: CX_FLT_MASK_UR_FUNC_MISMATCH - 0: Treat Function MisMatched TLPs as UR - 1: Do not treat Function MisMatched TLPs as UR Note: This register field is sticky.
15	0x0	**DISABLE_FC_WD_TIMER**: Disable FC Watchdog Timer. Note: This register field is sticky.
14:11	0x0	**EIDLE_TIMER**: EIDLE_TIMER is an internally reserved field. Do not use. Note: This register field is sticky.

Bit	Reset	Description
10:0	0x140	**SKP_INT_VAL**: SKP Interval Value. The number of symbol times to wait between transmitting SKP ordered sets. The controller waits the number of symbol times in this register plus 1, between transmitting SKP ordered sets. Your application must program this register accordingly. For example, if 1536 were programmed into this register (in a 250 MHz controller), then the controller actually transmits SKP ordered sets once every 1537 symbol times. The value programmed to this register is actually clock ticks and not symbol times. In a 125 MHz controller, programming the value programmed to this register should be scaled down by a factor of 2 (because one clock tick =two symbol times in this case). Note: This value is not used at Gen3 speed; the skip interval is hardcoded to 370 blocks. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_FILTER_MASK_2_OFF_0

where <*i*> = 1, 4, 8.

Description: This register modifies the RADM filtering and error handling rules. For more details, see the "Receive Filtering" section. In each case, '0' applies the associated filtering rule and '1' masks the associated filtering rule.

PCIE_X1_RC_PF0_PORT_LOGIC_FILTER_MASK_2_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_FILTER_MASK_2_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_FILTER_MASK_2_OFF_0

Offset: 0x720
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MASK_RADM_2**: Filter Mask 2. This field modifies the RADM filtering and error handling rules. For more details, see the "Receive Filtering" in the Databook. In each case, '0' applies the associated filtering rule and '1' masks the associated filtering rule. [31:10]: Reserved [9]: CX_FLT_MASK_CPL_IN_LUT_CHECK - 0: Disable masking of checking if the tag of CPL is registered in LUT - 1: Enable masking of checking if the tag of CPL is registered in LUT [8]: CX_FLT_MASK_POIS_ERROR_REPORTING - 0: Disable masking of error reporting for Poisoned TLPs - 1: Enable masking of error reporting for Poisoned TLPs [7]: CX_FLT_MASK_PRS_DROP - 0: Allow PRS message to pass through - 1: Drop PRS Messages silently - This bit is ignored when the CX_FLT_MASK_MSG_DROP bit in the MASK_RADM_1 field of the SYMBOL_TIMER_FILTER_1_OFF register is set to '1'. [6]: CX_FLT_UNMASK_TD - 0: Disable unmask TD bit if CX_STRIP_ECRC_ENABLE - 1: Enable unmask TD bit if CX_STRIP_ECRC_ENABLE [5]: CX_FLT_UNMASK_UR_POIS_TRGT0 - 0: Disable unmask CX_FLT_MASK_UR_POIS with TRGT0 destination - 1: Enable unmask CX_FLT_MASK_UR_POIS with TRGT0 destination [4]: CX_FLT_MASK_LN_VENMSG1_DROP - 0: Allow LN message to pass through - 1: Drop LN Messages silently [3]: CX_FLT_MASK_HANDLE_FLUSH - 0: Disable controller Filter to handle flush request - 1: Enable controller Filter to handle flush request [2]: CX_FLT_MASK_DABORT_4UCPL - 0: Enable DLLP abort for unexpected completion - 1: Do not enable DLLP abort for unexpected completion [1]: CX_FLT_MASK_VENMSG1_DROP - 0: Vendor MSG Type 1 dropped silently - 1: Vendor MSG Type 1 not dropped [0]: CX_FLT_MASK_VENMSG0_DROP - 0: Vendor MSG Type 0 dropped with UR error reporting - 1: Vendor MSG Type 0 not dropped Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_AMBA_MUL_OB_DECOMP_NP_SUB_REQ_CTRL_OFF_0

where <*i*> = 1, 4, 8.

Description: This register controls multiple outbound decomposed NP subRequests operation.

PCIE_X1_RC_PF0_PORT_LOGIC_AMBA_MUL_OB_DECOMP_NP_SUB_REQ_CTRL_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_AMBA_MUL_OB_DECOMP_NP_SUB_REQ_CTRL_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_AMBA_MUL_OB_DECOMP_NP_SUB_REQ_CTRL_OFF_0

Offset: 0x724
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000001 (0b0000,0000,0000,0000,0000,0000,0000,0001)

Bit	R/W	Reset	Description
31:1	RO	0x0	**RSVDP_1**: Reserved for future use.

Bit	R/W	Reset	Description
0	RW	0x1	**OB_RD_SPLIT_BURST_EN**: Enable AMBA Multiple Outbound Decomposed NP SubRequests. You should not clear this register unless your application master is requesting an amount of read data greater than Max_Read_Request_Size, and the remote device (or switch) is reordering completions that have different tags. For more details, see "AXI Bridge Ordering" in the AXI chapter of the Databook. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_PL_DEBUG0_OFF_0

where <*i*> = 1, 4, 8.

Description: This register holds cxpl_debug_info[31:0].

PCIE_X1_RC_PF0_PORT_LOGIC_PL_DEBUG0_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_PL_DEBUG0_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_PL_DEBUG0_OFF_0

Offset: 0x728
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DEB_REG_0**: The value on cxpl_debug_info[31:0].

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_PL_DEBUG1_OFF_0

where <*i*> = 1, 4, 8.

Description: This register holds cxpl_debug_info[63:32].

PCIE_X1_RC_PF0_PORT_LOGIC_PL_DEBUG1_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_PL_DEBUG1_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_PL_DEBUG1_OFF_0

Offset: 0x72c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DEB_REG_1**: The value on cxpl_debug_info[63:32].

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_TX_P_FC_CREDIT_STATUS_OFF_0

where <*i*> = 1, 4, 8.

Description: This register provides transmit posted FC credit status.

PCIE_X1_RC_PF0_PORT_LOGIC_TX_P_FC_CREDIT_STATUS_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_TX_P_FC_CREDIT_STATUS_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_TX_P_FC_CREDIT_STATUS_OFF_0

Offset: 0x730
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**RSVDP_TX_P_FC_CREDIT_STATUS**: Reserved for future use.
27:16	0x0	**TX_P_HEADER_FC_CREDIT**: Transmit Posted Header FC Credits. - The posted Header credits advertised by the receiver at the other end of the link, updated with each UpdateFC DLLP. - Default value depends on the number of advertised credits for header and data - Scaled Flow Control: [4'b0, xtlh_xadm_ph_cdts, xtlh_xadm_pd_cdts]; When the number of advertised posted credits (both header and data) are infinite, then the default would be [4'b0, 12'hFFF, 16'hFFFF]. - No Scaling: [12'b0, xtlh_xadm_ph_cdts, xtlh_xadm_pd_cdts]; When the number of advertised posted credits (both header and data) are infinite, then the default would be [12'b0, 8'hFF, 12'hFFF].

Bit	Reset	Description
15:0	0x0	**TX_P_DATA_FC_CREDIT**: Transmit Posted Data FC Credits. - The posted Data credits advertised by the receiver at the other end of the link, updated with each UpdateFC DLLP. - Default value depends on the number of advertised credits for header and data - Scaled Flow Control: [4'b0, xtlh_xadm_ph_cdts, xtlh_xadm_pd_cdts]; When the number of advertised posted credits (both header and data) are infinite, then the default would be [4'b0, 12'hFFF, 16'hFFFF]. - No Scaling: [12'b0, xtlh_xadm_ph_cdts, xtlh_xadm_pd_cdts]; When the number of advertised posted credits (both header and data) are infinite, then the default would be [12'b0, 8'hFF, 12'hFFF].

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_TX_NP_FC_CREDIT_STATUS_OFF_0

where <*i*> = 1, 4, 8.

Description: This register provides the transmit Non-Posted FC credit status.

PCIE_X1_RC_PF0_PORT_LOGIC_TX_NP_FC_CREDIT_STATUS_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_TX_NP_FC_CREDIT_STATUS_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_TX_NP_FC_CREDIT_STATUS_OFF_0

Offset: 0x734
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**RSVDP_TX_NP_FC_CREDIT_STATUS**: Reserved for future use.
27:16	0x0	**TX_NP_HEADER_FC_CREDIT**: Transmit Non-Posted Header FC Credits. - The non-posted Header credits advertised by the receiver at the other end of the link, updated with each UpdateFC DLLP. - Default value depends on the number of advertised credits for header and data - Scaled Flow Control: [4'b0, xtlh_xadm_nph_cdts, xtlh_xadm_npd_cdts]; When the number of advertised non-posted credits (both header and data) are infinite, then the default would be [4'b0, 12'hFFF, 16'hFFFF]. - No Scaling: [12'b0, xtlh_xadm_nph_cdts, xtlh_xadm_npd_cdts]; When the number of advertised non-posted credits (both header and data) are infinite, then the default would be [12'b0, 8'hFF, 12'hFFF].

Bit	Reset	Description
15:0	0x0	**TX_NP_DATA_FC_CREDIT**: Transmit Non-Posted Data FC Credits. - The non-posted Data credits advertised by the receiver at the other end of the link, updated with each UpdateFC DLLP. - Default value depends on the number of advertised credits for header and data - Scaled Flow Control: [4'b0, xtlh_xadm_nph_cdts, xtlh_xadm_npd_cdts]; When the number of advertised non-posted credits (both header and data) are infinite, then the default would be [4'b0, 12'hFFF, 16'hFFFF]. - No Scaling: [12'b0, xtlh_xadm_nph_cdts, xtlh_xadm_npd_cdts]; When the number of advertised non-posted credits (both header and data) are infinite, then the default would be [12'b0, 8'hFF, 12'hFFF].

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_TX_CPL_FC_CREDIT_STATUS_OFF_0

where <*i*> = 1, 4, 8.

Description: This register provides transmit completion FC credit status.

PCIE_X1_RC_PF0_PORT_LOGIC_TX_CPL_FC_CREDIT_STATUS_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_TX_CPL_FC_CREDIT_STATUS_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_TX_CPL_FC_CREDIT_STATUS_OFF_0

Offset: 0x738
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**RSVDP_TX_CPL_FC_CREDIT_STATUS**: Reserved for future use.
27:16	0x0	**TX_CPL_HEADER_FC_CREDIT**: Transmit Completion Header FC Credits. - The Completion Header credits advertised by the receiver at the other end of the link, updated with each UpdateFC DLLP. - Default value depends on the number of advertised credits for header and data - Scaled Flow Control: [4'b0, xtlh_xadm_cplh_cdts, xtlh_xadm_cpld_cdts]; When the number of advertised completion credits (both header and data) are infinite, then the default would be [4'b0, 12'hFFF, 16'hFFFF]. - No Scaling: [12'b0, xtlh_xadm_cplh_cdts, xtlh_xadm_cpld_cdts]; When the number of advertised completion credits (both header and data) are infinite, then the default would be [12'b0, 8'hFF, 12'hFFF].

Bit	Reset	Description
15:0	0x0	**TX_CPL_DATA_FC_CREDIT**: Transmit Completion Data FC Credits. - The Completion Data credits advertised by the receiver at the other end of the link, updated with each UpdateFC DLLP. - Default value depends on the number of advertised credits for header and data - Scaled Flow Control: [4'b0, xtlh_xadm_cplh_cdts, xtlh_xadm_cpld_cdts]; When the number of advertised completion credits (both header and data) are infinite, then the default would be [4'b0, 12'hFFF, 16'hFFFF]. - No Scaling: [12'b0, xtlh_xadm_cplh_cdts, xtlh_xadm_cpld_cdts]; When the number of advertised completion credits (both header and data) are infinite, then the default would be [12'b0, 8'hFF, 12'hFFF].

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_QUEUE_STATUS_OFF_0

where <*i*> = 1, 4, 8.

Description: This register provides the queue status.

PCIE_X1_RC_PF0_PORT_LOGIC_QUEUE_STATUS_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_QUEUE_STATUS_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_QUEUE_STATUS_OFF_0

Offset: 0x73c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,xx00,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**TIMER_MOD_FLOW_CONTROL_EN**: FC Latency Timer Override Enable. Note: This register field is sticky.
30:29	RO	0x0	**RSVDP_29**: Reserved for future use.
28:16	RW	0x0	**TIMER_MOD_FLOW_CONTROL**: FC Latency Timer Override Value. When you set the "FC Latency Timer Override Enable" in this register, the value in this field will override the FC latency timer value that the controller calculates according to the PCIe specification. For more details, see "Flow Control" in the Databook. Note: This register field is sticky.
13	RO	0x0	**RX_SERIALIZATION_Q_NON_EMPTY**: Receive Serialization Queue Not Empty.
12:4	RO	0x0	**RSVDP_4**: Reserved for future use.

Bit	R/W	Reset	Description
3	RW	0x0	**RX_QUEUE_OVERFLOW**: Receive Credit Queue Overflow.
2	RO	0x0	**RX_QUEUE_NON_EMPTY**: Receive Credit Queue Not Empty.
1	RO	0x0	**TX_RETRY_BUFFER_NE**: Transmit Retry Buffer Not Empty.
0	RO	0x0	**RX_TLP_FC_CREDIT_NON_RETURN**: Received TLP FC Credits Not Returned.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_VC_TX_ARBI_1_OFF_0

where <*i*> = 1, 4, 8.

Description: This register is used for setting the WRR weights for VC0 - VC3.

PCIE_X1_RC_PF0_PORT_LOGIC_VC_TX_ARBI_1_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_VC_TX_ARBI_1_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_VC_TX_ARBI_1_OFF_0

Offset: 0x740
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000f (0b0000,0000,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
31:24	0x0	**WRR_WEIGHT_VC_3**: WRR Weight for VC3. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R
23:16	0x0	**WRR_WEIGHT_VC_2**: WRR Weight for VC2. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R
15:8	0x0	**WRR_WEIGHT_VC_1**: WRR Weight for VC1. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R
7:0	0xf	**WRR_WEIGHT_VC_0**: WRR Weight for VC0. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_VC_TX_ARBI_2_OFF_0

where <*i*> = 1, 4, 8.

Description: This register is used for setting the WRR weights for VC4 - VC7.

PCIE_X1_RC_PF0_PORT_LOGIC_VC_TX_ARBI_2_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_VC_TX_ARBI_2_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_VC_TX_ARBI_2_OFF_0

Offset: 0x744
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**WRR_WEIGHT_VC_7**: WRR Weight for VC7. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R
23:16	0x0	**WRR_WEIGHT_VC_6**: WRR Weight for VC6. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R
15:8	0x0	**WRR_WEIGHT_VC_5**: WRR Weight for VC5. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R
7:0	0x0	**WRR_WEIGHT_VC_4**: WRR Weight for VC4. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_VC0_P_RX_Q_CTRL_OFF_0

where <*i*> = 1, 4, 8.

Description: This register controls segmented-buffer VC0 posted receive queue operation.

Offset: 0x748
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_PORT_LOGIC_VC0_P_RX_Q_CTRL_OFF_0

Reset: 0x4523b070 (0b0100,0101,0010,0011,1011,0000,0111,0000)

Bit	Reset	Description
31	0x0	**VC_ORDERING_RX_Q**: VC Ordering for Receive Queues. Determines the VC ordering rule for the receive queues, used only in the segmented-buffer configuration: Note: This register field is sticky.
30	0x1	**TLP_TYPE_ORDERING_VC0**: TLP Type Ordering for VC0. Determines the TLP type ordering rule for VC0 receive queues, used only in the segmented-buffer configuration: Note: This register field is sticky.
29:28	0x0	**RESERVED5**: Reserved. Note: This register field is sticky.
27:26	0x1	**VC0_P_DATA_SCALE**: VC0 Scale Posted Data Credits. Note: This register field is sticky.
25:24	0x1	**VC0_P_HDR_SCALE**: VC0 Scale Posted Header Credits. Note: This register field is sticky.
23:21	0x1	**VC0_P_TLP_Q_MODE**: Reserved. Note: This register field is sticky.
20	0x0	**RESERVED4**: Reserved. Note: This register field is sticky.
19:12	0x3b	**VC0_P_HEADER_CREDIT**: VC0 Posted Header Credits. The number of initial posted header credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R (sticky) Note: This register field is sticky.
11:0	0x70	**VC0_P_DATA_CREDIT**: VC0 Posted Data Credits. The number of initial posted data credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R (sticky) Note: This register field is sticky.

PCIE_X4_RC_PF0_PORT_LOGIC_VC0_P_RX_Q_CTRL_OFF_0

Reset: 0x4622c140 (0b0100,0110,0010,0010,1100,0001,0100,0000)

Bit	Reset	Description
31	0x0	**VC_ORDERING_RX_Q**: VC Ordering for Receive Queues. Determines the VC ordering rule for the receive queues, used only in the segmented-buffer configuration: Note: This register field is sticky.

Bit	Reset	Description
30	0x1	**TLP_TYPE_ORDERING_VC0**: TLP Type Ordering for VC0. Determines the TLP type ordering rule for VC0 receive queues, used only in the segmented-buffer configuration: Note: This register field is sticky.
29:28	0x0	**RESERVED5**: Reserved. Note: This register field is sticky.
27:26	0x1	**VC0_P_DATA_SCALE**: VC0 Scale Posted Data Credits. Note: This register field is sticky.
25:24	0x2	**VC0_P_HDR_SCALE**: VC0 Scale Posted Header Credits. Note: This register field is sticky.
23:21	0x1	**VC0_P_TLP_Q_MODE**: Reserved. Note: This register field is sticky.
20	0x0	**RESERVED4**: Reserved. Note: This register field is sticky.
19:12	0x2c	**VC0_P_HEADER_CREDIT**: VC0 Posted Header Credits. The number of initial posted header credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R (sticky) Note: This register field is sticky.
11:0	0x140	**VC0_P_DATA_CREDIT**: VC0 Posted Data Credits. The number of initial posted data credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R (sticky) Note: This register field is sticky.

PCIE_X8_RC_PF0_PORT_LOGIC_VC0_P_RX_Q_CTRL_OFF_0

Reset: 0x46258260 (0b0100,0110,0010,0101,1000,0010,0110,0000)

Bit	Reset	Description
31	0x0	**VC_ORDERING_RX_Q**: VC Ordering for Receive Queues. Determines the VC ordering rule for the receive queues, used only in the segmented-buffer configuration: Note: This register field is sticky.
30	0x1	**TLP_TYPE_ORDERING_VC0**: TLP Type Ordering for VC0. Determines the TLP type ordering rule for VC0 receive queues, used only in the segmented-buffer configuration: Note: This register field is sticky.
29:28	0x0	**RESERVED5**: Reserved. Note: This register field is sticky.
27:26	0x1	**VC0_P_DATA_SCALE**: VC0 Scale Posted Data Credits. Note: This register field is sticky.
25:24	0x2	**VC0_P_HDR_SCALE**: VC0 Scale Posted Header Credits. Note: This register field is sticky.
23:21	0x1	**VC0_P_TLP_Q_MODE**: Reserved. Note: This register field is sticky.

Bit	Reset	Description
20	0x0	**RESERVED4**: Reserved. Note: This register field is sticky.
19:12	0x58	**VC0_P_HEADER_CREDIT**: VC0 Posted Header Credits. The number of initial posted header credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R (sticky) Note: This register field is sticky.
11:0	0x260	**VC0_P_DATA_CREDIT**: VC0 Posted Data Credits. The number of initial posted data credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R (sticky) Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_VC0_NP_RX_Q_CTRL_OFF_0

where <*i*> = 1, 4, 8.

Description: This register controls the segmented-buffer VC0 non-posted receive queue operation.

Offset: 0x74c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_PORT_LOGIC_VC0_NP_RX_Q_CTRL_OFF_0

Reset: 0x0523b00e (0b0000,0101,0010,0011,1011,0000,0000,1110)

Bit	Reset	Description
31:28	0x0	**RESERVED7**: Reserved. Note: This register field is sticky.
27:26	0x1	**VC0_NP_DATA_SCALE**: VC0 Scale Non-Posted Data Credits. Note: This register field is sticky.
25:24	0x1	**VC0_NP_HDR_SCALE**: VC0 Scale Non-Posted Header Credits. Note: This register field is sticky.
23:21	0x1	**VC0_NP_TLP_Q_MODE**: Reserved. Note: This register field is sticky.
20	0x0	**RESERVED6**: Reserved. Note: This register field is sticky.

Bit	Reset	Description
19:12	0x3b	**VC0_NP_HEADER_CREDIT**: VC0 Non-Posted Header Credits. The number of initial non-posted header credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R (sticky) Note: This register field is sticky.
11:0	0xe	**VC0_NP_DATA_CREDIT**: VC0 Non-Posted Data Credits. The number of initial non-posted data credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R (sticky) Note: This register field is sticky.

PCIE_X4_RC_PF0_PORT_LOGIC_VC0_NP_RX_Q_CTRL_OFF_0

Reset: 0x0622c02c (0b0000,0110,0010,0010,1100,0000,0010,1100)

Bit	Reset	Description
31:28	0x0	**RESERVED7**: Reserved. Note: This register field is sticky.
27:26	0x1	**VC0_NP_DATA_SCALE**: VC0 Scale Non-Posted Data Credits. Note: This register field is sticky.
25:24	0x2	**VC0_NP_HDR_SCALE**: VC0 Scale Non-Posted Header Credits. Note: This register field is sticky.
23:21	0x1	**VC0_NP_TLP_Q_MODE**: Reserved. Note: This register field is sticky.
20	0x0	**RESERVED6**: Reserved. Note: This register field is sticky.
19:12	0x2c	**VC0_NP_HEADER_CREDIT**: VC0 Non-Posted Header Credits. The number of initial non-posted header credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R (sticky) Note: This register field is sticky.
11:0	0x2c	**VC0_NP_DATA_CREDIT**: VC0 Non-Posted Data Credits. The number of initial non-posted data credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R (sticky) Note: This register field is sticky.

PCIE_X8_RC_PF0_PORT_LOGIC_VC0_NP_RX_Q_CTRL_OFF_0

Reset: 0x06258058 (0b0000,0110,0010,0101,1000,0000,0101,1000)

Bit	Reset	Description
31:28	0x0	**RESERVED7**: Reserved. Note: This register field is sticky.
27:26	0x1	**VC0_NP_DATA_SCALE**: VC0 Scale Non-Posted Data Credits. Note: This register field is sticky.

Bit	Reset	Description
25:24	0x2	**VC0_NP_HDR_SCALE**: VC0 Scale Non-Posted Header Credits. Note: This register field is sticky.
23:21	0x1	**VC0_NP_TLP_Q_MODE**: Reserved. Note: This register field is sticky.
20	0x0	**RESERVED6**: Reserved. Note: This register field is sticky.
19:12	0x58	**VC0_NP_HEADER_CREDIT**: VC0 Non-Posted Header Credits. The number of initial non-posted header credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R (sticky) Note: This register field is sticky.
11:0	0x58	**VC0_NP_DATA_CREDIT**: VC0 Non-Posted Data Credits. The number of initial non-posted data credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R (sticky) Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_VC0_CPL_RX_Q_CTRL_OFF_0

where <*i*> = 1, 4, 8.

Description: This register controls the segmented-buffer VC0 completion receive queue operation.

Offset: 0x750
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_PORT_LOGIC_VC0_CPL_RX_Q_CTRL_OFF_0

Reset: 0x05200000 (0b0000,0101,0010,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**RESERVED9**: Reserved. Note: This register field is sticky.
27:26	0x1	**VC0_CPL_DATA_SCALE**: VC0 Scale CPL Data Credits. Note: This register field is sticky.
25:24	0x1	**VC0_CPL_HDR_SCALE**: VC0 Scale CPL Header Credits. Note: This register field is sticky.
23:21	0x1	**VC0_CPL_TLP_Q_MODE**: Reserved. Note: This register field is sticky.
20	0x0	**RESERVED8**: Reserved. Note: This register field is sticky.

Bit	Reset	Description
19:12	0x0	**VC0_CPL_HEADER_CREDIT**: VC0 Completion Header Credits. The number of initial Completion header credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R (sticky) Note: This register field is sticky.
11:0	0x0	**VC0_CPL_DATA_CREDIT**: VC0 Completion Data Credits. The number of initial Completion data credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R (sticky) Note: This register field is sticky.

PCIE_X4_RC_PF0_PORT_LOGIC_VC0_CPL_RX_Q_CTRL_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_VC0_CPL_RX_Q_CTRL_OFF_0

Reset: 0x06200000 (0b0000,0110,0010,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**RESERVED9**: Reserved. Note: This register field is sticky.
27:26	0x1	**VC0_CPL_DATA_SCALE**: VC0 Scale CPL Data Credits. Note: This register field is sticky.
25:24	0x2	**VC0_CPL_HDR_SCALE**: VC0 Scale CPL Header Credits. Note: This register field is sticky.
23:21	0x1	**VC0_CPL_TLP_Q_MODE**: Reserved. Note: This register field is sticky.
20	0x0	**RESERVED8**: Reserved. Note: This register field is sticky.
19:12	0x0	**VC0_CPL_HEADER_CREDIT**: VC0 Completion Header Credits. The number of initial Completion header credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R (sticky) Note: This register field is sticky.
11:0	0x0	**VC0_CPL_DATA_CREDIT**: VC0 Completion Data Credits. The number of initial Completion data credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R (sticky) Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_GEN2_CTRL_OFF_0

where <*i*> = 1, 4, 8.

Description: This register controls various functions of the controller related to link training, lane reversal, and equalization.

Offset: 0x80c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_PORT_LOGIC_GEN2_CTRL_OFF_0

Reset: 0x00020134 (0b0xxx,xxxx,0000,0010,0000,0001,0011,0100)

Bit	R/W	Reset	Description
31	RO	0x0	**RSVDP_31**: Reserved for future use.
23	RW	0x0	**SELECTABLE_DEEMPH_BIT_MUX**: The selectable deemphasis bit (Symbol 4 bit 6) of the transmitted TS2 Ordered Sets for DSP in Recovery.RcvrCfg state is muxed between the Selectable De-emphasis field in the Link Control 2 register and the value requested by USP in Recovery.RcvrLock state through Tx TS1s from USP. Note: This register field is sticky.
22	RW	0x0	**SELECT_DEEMPH_VAR_MUX**: The select_deemphasis variable for DSP on entry to Recovery.RcvrCfg state is muxed between the Selectable De-emphasis field in the Link Control 2 register and the value requested by the Upstream Port in the eight consecutive TS1 Ordered Sets it received. Note: This register field is sticky.
21	RW	0x0	**GEN1_EI_INFERENCE**: Electrical Idle Inference Mode at Gen1 Rate. Programmable mode to determine inferred electrical idle (EI) in Recovery.Speed or Loopback.Active (as slave) state at Gen1 speed by looking for a '1' value on RxElecIdle instead of looking for a '0' on RxValid. If the PHY fails to deassert the RxValid signal in Recovery.Speed or Loopback.Active (because of corrupted EIOS for example), then EI cannot be inferred successfully in the controller by just detecting the condition RxValid=0. Note: This register field is sticky.
20	RW	0x0	**SEL_DEEMPHASIS**: Used to set the de-emphasis level for upstream ports. This bit selects the level of de-emphasis the link operates at. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.
19	RW	0x0	**CONFIG_TX_COMP_RX**: Config Tx Compliance Receive Bit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.
18	RW	0x0	**CONFIG_PHY_TX_CHANGE**: Config PHY Tx Swing. Controls the PHY transmitter voltage swing level. The controller drives the mac_phy_txswing output from this register bit field. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.

Bit	R/W	Reset	Description
17	RW	0x1	**DIRECT_SPEED_CHANGE**: Directed Speed Change. When the speed change occurs, the controller will clear the contents of this field; and a read to this field by your software will return a '0'. To manually initiate the speed change: - Write to LINK_CONTROL2_LINK_STATUS2_REG.PCIE_CAP_TARGET_LINK_SPEED in the local device - Deassert this field - Assert this field If you set the default of this field using the DEFAULT_GEN2_SPEED_CHANGE configuration parameter to '1', then the speed change is initiated automatically after link up, and the controller clears the contents of this field. If you want to prevent this automatic speed change, then write a lower speed value to the Target Link Speed field of the Link Control 2 register (LINK_CONTROL2_LINK_STATUS2_OFF.PCIE_CAP_TARGET_LINK_SPEED) through the DBI before link up. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
16	RW	0x0	**AUTO_LANE_FLIP_CTRL_EN**: Enable Auto flipping of the lanes. You must set the CX_AUTO_LANE_FLIP_CTRL_EN configuration parameter to include the hardware for this feature in the controller. For more details, see the 'Lane Reversal' appendix in the Databook. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.
15:13	RW	0x0	**PRE_DET_LANE**: Predetermined Lane for Auto Flip. This field defines which physical lane is connected to logical Lane0 by the flip operation performed in Detect. This field is used to restrict the receiver detect procedure to a particular lane when the default detect and polling procedure performed on all lanes cannot be successful. A notable example of when it is useful to program this field to a value different from the default, is when a lane is asymmetrically broken, that is, it is detected in Detect LTSSM state but it cannot exit Electrical Idle in Polling LTSSM state. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.
12:8	RW	0x1	**NUM_OF_LANES**: Predetermined Number of Lanes. Defines the number of lanes which are connected and not bad. Used to limit the effective link width to ignore 'broken" or "unused" lanes that detect a receiver. Indicates the number of lanes to check for exit from Electrical Idle in Polling.Active and L2.Idle. It is possible that the LTSSM might detect a receiver on a bad or broken lane during the Detect Substate. However, it is also possible that such a lane might also fail to exit Electrical Idle and therefore prevent a valid link from being configured. This value is referred to as the "Predetermined Number of Lanes" in section 4.2.6.2.1 of the PCI Express Base Specification. When you have unused lanes in your system, then you must change the value in this register to reflect the number of lanes. You must also change the value in the "Link Mode Enable" field of PORT_LINK_CTRL_OFF. The value in this register is normally the same as the encoded value in PORT_LINK_CTRL_OFF. If you find that one of your used lanes is bad then you must reduce the value in this register. For more information, see "How to Tie Off Unused Lanes" in the Databook. For information on upsizing and downsizing the link width, see "Link Establishment" in the Databook. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.

Bit	R/W	Reset	Description
7:0	RW	0x34	**FAST_TRAINING_SEQ**: Sets the Number of Fast Training Sequences (N_FTS) that the controller advertises as its N_FTS during Gen2 or Gen3 link training. This value is used to inform the link partner about the PHY's ability to recover synchronization after a low power state. The number should be provided by the PHY vendor. Do not set N_FTS to zero; doing so can cause the LTSSM to go into the recovery state when exiting from L0s. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.

PCIE_X4_RC_PF0_PORT_LOGIC_GEN2_CTRL_OFF_0

Reset: 0x00030434 (0b00xx,0000,0000,0011,0000,0100,0011,0100)

Bit	R/W	Reset	Description
Bit	R/W	Reset	Description
31	RO	0x0	**RSVDP_31**: Reserved for future use.
30	RW	0x0	**FORCE_LANE_FLIP**: Enable to force the LANE_UNDER_TEST physical lane flips to logical lane 0. All the other physical lanes are turned off. The LINK_CAPABLE register must be set to 1 and only x1 link can be formed if the FORCE_LANE_FLIP register is set to 1. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.
27:24	RW	0x0	**LANE_UNDER_TEST**: The Lane Under Test is the lane for Forced Lane Flip or for Loopback Eq. Only one lane is configured each time. The default of this field is the CX_DEFAULT_LANE_UNDER_TEST configuration parameter. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.
23	RW	0x0	**SELECTABLE_DEEMPH_BIT_MUX**: The selectable deemphasis bit (Symbol 4 bit 6) of the transmitted TS2 Ordered Sets for DSP in Recovery.RcvrCfg state is muxed between the Selectable De-emphasis field in the Link Control 2 register and the value requested by USP in Recovery.RcvrLock state through Tx TS1s from USP. Note: This register field is sticky.
22	RW	0x0	**SELECT_DEEMPH_VAR_MUX**: The select_deemphasis variable for DSP on entry to Recovery.RcvrCfg state is muxed between the Selectable De-emphasis field in the Link Control 2 register and the value requested by the Upstream Port in the eight consecutive TS1 Ordered Sets it received. Note: This register field is sticky.
21	RW	0x0	**GEN1_EI_INFERENCE**: Electrical Idle Inference Mode at Gen1 Rate. Programmable mode to determine inferred electrical idle (EI) in Recovery.Speed or Loopback.Active (as slave) state at Gen1 speed by looking for a '1' value on RxElecIdle instead of looking for a '0' on RxValid. If the PHY fails to deassert the RxValid signal in Recovery.Speed or Loopback.Active (because of corrupted EIOS for example), then EI cannot be inferred successfully in the controller by just detecting the condition RxValid=0. Note: This register field is sticky.

Bit	R/W	Reset	Description
Bit	**R/W**	**Reset**	**Description**
20	RW	0x0	**SEL_DEEMPHASIS**: Used to set the de-emphasis level for upstream ports. This bit selects the level of de-emphasis the link operates at. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.
19	RW	0x0	**CONFIG_TX_COMP_RX**: Config Tx Compliance Receive Bit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.
18	RW	0x0	**CONFIG_PHY_TX_CHANGE**: Config PHY Tx Swing. Controls the PHY transmitter voltage swing level. The controller drives the mac_phy_txswing output from this register bit field. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.
17	RW	0x1	**DIRECT_SPEED_CHANGE**: Directed Speed Change. When the speed change occurs, the controller will clear the contents of this field; and a read to this field by your software will return a '0'. To manually initiate the speed change: - Write to LINK_CONTROL2_LINK_STATUS2_REG.PCIE_CAP_TARGET_LINK_SPEED in the local device - Deassert this field - Assert this field If you set the default of this field using the DEFAULT_GEN2_SPEED_CHANGE configuration parameter to '1', then the speed change is initiated automatically after link up, and the controller clears the contents of this field. If you want to prevent this automatic speed change, then write a lower speed value to the Target Link Speed field of the Link Control 2 register (LINK_CONTROL2_LINK_STATUS2_OFF.PCIE_CAP_TARGET_LINK_SPEED) through the DBI before link up. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
16	RW	0x1	**AUTO_LANE_FLIP_CTRL_EN**: Enable Auto flipping of the lanes. You must set the CX_AUTO_LANE_FLIP_CTRL_EN configuration parameter to include the hardware for this feature in the controller. For more details, see the 'Lane Reversal' appendix in the Databook. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.
15:13	RW	0x0	**PRE_DET_LANE**: Predetermined Lane for Auto Flip. This field defines which physical lane is connected to logical Lane0 by the flip operation performed in Detect. This field is used to restrict the receiver detect procedure to a particular lane when the default detect and polling procedure performed on all lanes cannot be successful. A notable example of when it is useful to program this field to a value different from the default, is when a lane is asymmetrically broken, that is, it is detected in Detect LTSSM state but it cannot exit Electrical Idle in Polling LTSSM state. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.

Bit	R/W	Reset	Description
Bit	**R/W**	**Reset**	**Description**
12:8	RW	0x4	**NUM_OF_LANES**: Predetermined Number of Lanes. Defines the number of lanes which are connected and not bad. Used to limit the effective link width to ignore 'broken" or "unused" lanes that detect a receiver. Indicates the number of lanes to check for exit from Electrical Idle in Polling.Active and L2.Idle. It is possible that the LTSSM might detect a receiver on a bad or broken lane during the Detect Substate. However, it is also possible that such a lane might also fail to exit Electrical Idle and therefore prevent a valid link from being configured. This value is referred to as the "Predetermined Number of Lanes" in section 4.2.6.2.1 of the PCI Express Base Specification. When you have unused lanes in your system, then you must change the value in this register to reflect the number of lanes. You must also change the value in the "Link Mode Enable" field of PORT_LINK_CTRL_OFF. The value in this register is normally the same as the encoded value in PORT_LINK_CTRL_OFF. If you find that one of your used lanes is bad then you must reduce the value in this register. For more information, see "How to Tie Off Unused Lanes" in the Databook. For information on upsizing and downsizing the link width, see "Link Establishment" in the Databook. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.
7:0	RW	0x34	**FAST_TRAINING_SEQ**: Sets the Number of Fast Training Sequences (N_FTS) that the controller advertises as its N_FTS during Gen2 or Gen3 link training. This value is used to inform the link partner about the PHY's ability to recover synchronization after a low power state. The number should be provided by the PHY vendor. Do not set N_FTS to zero; doing so can cause the LTSSM to go into the recovery state when exiting from L0s. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.

PCIE_X8_RC_PF0_PORT_LOGIC_GEN2_CTRL_OFF_0

Reset: 0x00030834 (0b00xx,0000,0000,0011,0000,1000,0011,0100)

Bit	R/W	Reset	Description
Bit	**R/W**	**Reset**	**Description**
31	RO	0x0	**RSVDP_31**: Reserved for future use.
30	RW	0x0	**FORCE_LANE_FLIP**: Enable to force the LANE_UNDER_TEST physical lane flips to logical lane 0. All the other physical lanes are turned off. The LINK_CAPABLE register must be set to 1 and only x1 link can be formed if the FORCE_LANE_FLIP register is set to 1. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.
27:24	RW	0x0	**LANE_UNDER_TEST**: The Lane Under Test is the lane for Forced Lane Flip or for Loopback Eq. Only one lane is configured each time. The default of this field is the CX_DEFAULT_LANE_UNDER_TEST configuration parameter. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.

Bit	R/W	Reset	Description
Bit	**R/W**	**Reset**	**Description**
23	RW	0x0	**SELECTABLE_DEEMPH_BIT_MUX**: The selectable deemphasis bit (Symbol 4 bit 6) of the transmitted TS2 Ordered Sets for DSP in Recovery.RcvrCfg state is muxed between the Selectable De-emphasis field in the Link Control 2 register and the value requested by USP in Recovery.RcvrLock state through Tx TS1s from USP. Note: This register field is sticky.
22	RW	0x0	**SELECT_DEEMPH_VAR_MUX**: The select_deemphasis variable for DSP on entry to Recovery.RcvrCfg state is muxed between the Selectable De-emphasis field in the Link Control 2 register and the value requested by the Upstream Port in the eight consecutive TS1 Ordered Sets it received. Note: This register field is sticky.
21	RW	0x0	**GEN1_EI_INFERENCE**: Electrical Idle Inference Mode at Gen1 Rate. Programmable mode to determine inferred electrical idle (EI) in Recovery.Speed or Loopback.Active (as slave) state at Gen1 speed by looking for a '1' value on RxElecIdle instead of looking for a '0' on RxValid. If the PHY fails to deassert the RxValid signal in Recovery.Speed or Loopback.Active (because of corrupted EIOS for example), then EI cannot be inferred successfully in the controller by just detecting the condition RxValid=0. Note: This register field is sticky.
20	RW	0x0	**SEL_DEEMPHASIS**: Used to set the de-emphasis level for upstream ports. This bit selects the level of de-emphasis the link operates at. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.
19	RW	0x0	**CONFIG_TX_COMP_RX**: Config Tx Compliance Receive Bit. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.
18	RW	0x0	**CONFIG_PHY_TX_CHANGE**: Config PHY Tx Swing. Controls the PHY transmitter voltage swing level. The controller drives the mac_phy_txswing output from this register bit field. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.
17	RW	0x1	**DIRECT_SPEED_CHANGE**: Directed Speed Change. When the speed change occurs, the controller will clear the contents of this field; and a read to this field by your software will return a '0'. To manually initiate the speed change: - Write to LINK_CONTROL2_LINK_STATUS2_REG.PCIE_CAP_TARGET_LINK_SPEED in the local device - Deassert this field - Assert this field If you set the default of this field using the DEFAULT_GEN2_SPEED_CHANGE configuration parameter to '1', then the speed change is initiated automatically after link up, and the controller clears the contents of this field. If you want to prevent this automatic speed change, then write a lower speed value to the Target Link Speed field of the Link Control 2 register (LINK_CONTROL2_LINK_STATUS2_OFF.PCIE_CAP_TARGET_LINK_SPEED) through the DBI before link up. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

Bit	R/W	Reset	Description
Bit	**R/W**	**Reset**	**Description**
16	RW	0x1	**AUTO_LANE_FLIP_CTRL_EN**: Enable Auto flipping of the lanes. You must set the CX_AUTO_LANE_FLIP_CTRL_EN configuration parameter to include the hardware for this feature in the controller. For more details, see the 'Lane Reversal' appendix in the Databook. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.
15:13	RW	0x0	**PRE_DET_LANE**: Predetermined Lane for Auto Flip. This field defines which physical lane is connected to logical Lane0 by the flip operation performed in Detect. This field is used to restrict the receiver detect procedure to a particular lane when the default detect and polling procedure performed on all lanes cannot be successful. A notable example of when it is useful to program this field to a value different from the default, is when a lane is asymmetrically broken, that is, it is detected in Detect LTSSM state but it cannot exit Electrical Idle in Polling LTSSM state. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.
12:8	RW	0x8	**NUM_OF_LANES**: Predetermined Number of Lanes. Defines the number of lanes which are connected and not bad. Used to limit the effective link width to ignore 'broken" or "unused" lanes that detect a receiver. Indicates the number of lanes to check for exit from Electrical Idle in Polling.Active and L2.Idle. It is possible that the LTSSM might detect a receiver on a bad or broken lane during the Detect Substate. However, it is also possible that such a lane might also fail to exit Electrical Idle and therefore prevent a valid link from being configured. This value is referred to as the "Predetermined Number of Lanes" in section 4.2.6.2.1 of the PCI Express Base Specification. When you have unused lanes in your system, then you must change the value in this register to reflect the number of lanes. You must also change the value in the "Link Mode Enable" field of PORT_LINK_CTRL_OFF. The value in this register is normally the same as the encoded value in PORT_LINK_CTRL_OFF. If you find that one of your used lanes is bad then you must reduce the value in this register. For more information, see "How to Tie Off Unused Lanes" in the Databook. For information on upsizing and downsizing the link width, see "Link Establishment" in the Databook. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.
7:0	RW	0x34	**FAST_TRAINING_SEQ**: Sets the Number of Fast Training Sequences (N_FTS) that the controller advertises as its N_FTS during Gen2 or Gen3 link training. This value is used to inform the link partner about the PHY's ability to recover synchronization after a low power state. The number should be provided by the PHY vendor. Do not set N_FTS to zero; doing so can cause the LTSSM to go into the recovery state when exiting from L0s. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_PHY_STATUS_OFF_0

where <*i*> = 1, 4, 8.

Description: Memory mapped register from phy_cfg_status GPIO input pins.

PCIE_X1_RC_PF0_PORT_LOGIC_PHY_STATUS_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_PHY_STATUS_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_PHY_STATUS_OFF_0

Offset: 0x810
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PHY_STATUS**: PHY Status. Data received directly from the phy_cfg_status bus. These is a GPIO register reflecting the values on the static phy_cfg_status input signals. The usage is left completely to the user and does not in any way influence controller functionality. You can use it for any static sideband status signalling requirements that you have for your PHY. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_PHY_CONTROL_OFF_0

where <*i*> = 1, 4, 8.

Description: Memory mapped register to cfg_phy_control GPIO output pins.

PCIE_X1_RC_PF0_PORT_LOGIC_PHY_CONTROL_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_PHY_CONTROL_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_PHY_CONTROL_OFF_0

Offset: 0x814
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PHY_CONTROL**: PHY Control. Data sent directly to the cfg_phy_control bus. This is a GPIO register driving the values on the static cfg_phy_control output signals, and does not in any way influence controller functionality. It can be used for any static sideband control signaling requirements that you have for your PHY. This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_TRGT_MAP_CTRL_OFF_0

where <*i*> = 1, 4, 8.

Description: This register controls the target map.

Offset: 0x81c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_PORT_LOGIC_TRGT_MAP_CTRL_OFF_0

Reset: 0x0000007f (0b0000,0000,0000,0000,000x,xxxx,x111,1111)

Bit	R/W	Reset	Description
31:21	RO	0x0	**TARGET_MAP_RESERVED_21_31**: Reserved. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R (sticky)
20:16	RW	0x0	**TARGET_MAP_INDEX**: The number of the PF Function on which the Target Values are set. This register does not respect the Byte Enable setting, any write will affect all register bits.
15:13	RO	0x0	**TARGET_MAP_RESERVED_13_15**: Reserved. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R (sticky)
6	RW	0x1	**TARGET_MAP_ROM**: Target Value for the ROM page of the PF Function selected by the index number. This register does not respect the Byte Enable setting, any write will affect all register bits.
5:0	RW	0x3f	**TARGET_MAP_PF**: Target Values for each BAR on the PF Function selected by the index number. This register does not respect the Byte Enable setting, any write will affect all register bits.

PCIE_X4_RC_PF0_PORT_LOGIC_TRGT_MAP_CTRL_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_TRGT_MAP_CTRL_OFF_0

Reset: 0x0000006b (0b0000,0000,0000,0000,000x,xxxx,x110,1011)

Bit	R/W	Reset	Description
31:21	RO	0x0	**TARGET_MAP_RESERVED_21_31**: Reserved. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R (sticky)

Bit	R/W	Reset	Description
20:16	RW	0x0	**TARGET_MAP_INDEX**: The number of the PF Function on which the Target Values are set. This register does not respect the Byte Enable setting, any write will affect all register bits.
15:13	RO	0x0	**TARGET_MAP_RESERVED_13_15**: Reserved. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R (sticky)
6	RW	0x1	**TARGET_MAP_ROM**: Target Value for the ROM page of the PF Function selected by the index number. This register does not respect the Byte Enable setting, any write will affect all register bits.
5:0	RW	0x2b	**TARGET_MAP_PF**: Target Values for each BAR on the PF Function selected by the index number. This register does not respect the Byte Enable setting, any write will affect all register bits.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_MSI_CTRL_ADDR_OFF_0

where <*i*> = 1, 4, 8.

Description: This register holds the integrated MSI reception module address.

PCIE_X1_RC_PF0_PORT_LOGIC_MSI_CTRL_ADDR_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_MSI_CTRL_ADDR_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSI_CTRL_ADDR_OFF_0

Offset: 0x820
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_CTRL_ADDR**: Integrated MSI Reception Module Address. System specified address for MSI memory write transaction termination. Within the AXI Bridge, every received Memory Write request is examined to see if it targets the MSI Address that has been specified in this register; and also to see if it satisfies the definition of an MSI interrupt request. When these conditions are satisfied the Memory Write request is marked as an MSI request. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_MSI_CTRL_UPPER_ADDR_OFF_0

where <*i*> = 1, 4, 8.

Description: This register holds the integrated MSI reception module upper address.

PCIE_X1_RC_PF0_PORT_LOGIC_MSI_CTRL_UPPER_ADDR_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_MSI_CTRL_UPPER_ADDR_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSI_CTRL_UPPER_ADDR_OFF_0

Offset: 0x824
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_CTRL_UPPER_ADDR**: Integrated MSI Reception Module Upper Address. System specified upper address for MSI memory write transaction termination. Allows functions to support a 64-bit MSI address. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_0_EN_OFF_0

where <*i*> = 1, 4, 8.

Description: This register enables integrated MSI reception module interrupt0.

PCIE_X1_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_0_EN_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_0_EN_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_0_EN_OFF_0

Offset: 0x828
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_CTRL_INT_0_EN**: MSI Interrupt0 Enable. Specifies which interrupts are enabled. When an MSI is received from a disabled interrupt, no status bit gets set in MSI controller interrupt status register. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_0_MASK_OFF_0

where <*i*> = 1, 4, 8.

Description: This register provides information regarding the MSI Interrupt0 mask.

PCIE_X1_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_0_MASK_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_0_MASK_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_0_MASK_OFF_0

Offset: 0x82c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_CTRL_INT_0_MASK**: MSI Interrupt0 Mask. Allows enabled interrupts to be masked. When an MSI is received for a masked interrupt, the corresponding status bit gets set in the interrupt status register but the msi_ctrl_int output is not set HIGH. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_0_STATUS_OFF_0

where <*i*> = 1, 4, 8.

Description: This register provides the MSI Interrupt0 status.

PCIE_X1_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_0_STATUS_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_0_STATUS_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_0_STATUS_OFF_0

Offset: 0x830
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_CTRL_INT_0_STATUS**: MSI Interrupt0 Status. When an MSI is detected for EP#i, one bit in this register is set. The decoding of the data payload of the MSI Memory Write request determines which bit gets set. A status is bit is cleared by writing a 1 to the bit. Each bit corresponds to a single MSI Interrupt Vector.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_1_EN_OFF_0

where <*i*> = 1, 4, 8.

Description: This register enables integrated MSI reception module interrupt1.

PCIE_X1_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_1_EN_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_1_EN_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_1_EN_OFF_0

Offset: 0x834
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_CTRL_INT_1_EN**: MSI Interrupt1 Enable. Specifies which interrupts are enabled. When an MSI is received from a disabled interrupt, no status bit gets set in MSI controller interrupt status register. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_1_MASK_OFF_0

where <*i*> = 1, 4, 8.

Description: This register provides information regarding the MSI Interrupt1 mask.

PCIE_X1_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_1_MASK_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_1_MASK_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_1_MASK_OFF_0

Offset: 0x838
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_CTRL_INT_1_MASK**: MSI Interrupt1 Mask. Allows enabled interrupts to be masked. When an MSI is received for a masked interrupt, the corresponding status bit gets set in the interrupt status register but the msi_ctrl_int output is not set HIGH. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_1_STATUS_OFF_0

where <*i*> = 1, 4, 8.

Description: This register provides the MSI Interrupt1 status.

PCIE_X1_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_1_STATUS_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_1_STATUS_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_1_STATUS_OFF_0

Offset: 0x83c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_CTRL_INT_1_STATUS**: MSI Interrupt1 Status. When an MSI is detected for EP#i, one bit in this register is set. The decoding of the data payload of the MSI Memory Write request determines which bit gets set. A status is bit is cleared by writing a 1 to the bit. Each bit corresponds to a single MSI Interrupt Vector.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_2_EN_OFF_0

where <*i*> = 1, 4, 8.

Description: This register enables integrated MSI reception module interrupt2.

PCIE_X1_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_2_EN_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_2_EN_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_2_EN_OFF_0

Offset: 0x840
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_CTRL_INT_2_EN**: MSI Interrupt2 Enable. Specifies which interrupts are enabled. When an MSI is received from a disabled interrupt, no status bit gets set in MSI controller interrupt status register. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_2_MASK_OFF_0

where <*i*> = 1, 4, 8.

Description: This register provides information regarding the MSI Interrupt2 mask.

PCIE_X1_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_2_MASK_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_2_MASK_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_2_MASK_OFF_0

Offset: 0x844
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_CTRL_INT_2_MASK**: MSI Interrupt2 Mask. Allows enabled interrupts to be masked. When an MSI is received for a masked interrupt, the corresponding status bit gets set in the interrupt status register but the msi_ctrl_int output is not set HIGH. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_2_STATUS_OFF_0

where <*i*> = 1, 4, 8.

Description: This register provides the MSI Interrupt2 status.

PCIE_X1_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_2_STATUS_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_2_STATUS_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_2_STATUS_OFF_0

Offset: 0x848
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_CTRL_INT_2_STATUS**: MSI Interrupt2 Status. When an MSI is detected for EP#i, one bit in this register is set. The decoding of the data payload of the MSI Memory Write request determines which bit gets set. A status is bit is cleared by writing a 1 to the bit. Each bit corresponds to a single MSI Interrupt Vector.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_3_EN_OFF_0

where <*i*> = 1, 4, 8.

Description: This register enables integrated MSI reception module interrupt3.

PCIE_X1_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_3_EN_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_3_EN_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_3_EN_OFF_0

Offset: 0x84c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_CTRL_INT_3_EN**: MSI Interrupt3 Enable. Specifies which interrupts are enabled. When an MSI is received from a disabled interrupt, no status bit gets set in MSI controller interrupt status register. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_3_MASK_OFF_0

where <*i*> = 1, 4, 8.

Description: This register provides information regarding the MSI Interrupt3 mask.

PCIE_X1_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_3_MASK_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_3_MASK_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_3_MASK_OFF_0

Offset: 0x850
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_CTRL_INT_3_MASK**: MSI Interrupt3 Mask. Allows enabled interrupts to be masked. When an MSI is received for a masked interrupt, the corresponding status bit gets set in the interrupt status register but the msi_ctrl_int output is not set HIGH. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_3_STATUS_OFF_0

where <*i*> = 1, 4, 8.

Description: This register provides the MSI Interrupt3 status.

PCIE_X1_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_3_STATUS_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_3_STATUS_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_3_STATUS_OFF_0

Offset: 0x854
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_CTRL_INT_3_STATUS**: MSI Interrupt3 Status. When an MSI is detected for EP#i, one bit in this register is set. The decoding of the data payload of the MSI Memory Write request determines which bit gets set. A status is bit is cleared by writing a 1 to the bit. Each bit corresponds to a single MSI Interrupt Vector.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_4_EN_OFF_0

where <*i*> = 1, 4, 8.

Description: This register enables integrated MSI reception module interrupt4.

PCIE_X1_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_4_EN_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_4_EN_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_4_EN_OFF_0

Offset: 0x858
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_CTRL_INT_4_EN**: MSI Interrupt4 Enable. Specifies which interrupts are enabled. When an MSI is received from a disabled interrupt, no status bit gets set in MSI controller interrupt status register. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_4_MASK_OFF_0

where <*i*> = 1, 4, 8.

Description: This register provides information regarding the MSI Interrupt4 mask.

PCIE_X1_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_4_MASK_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_4_MASK_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_4_MASK_OFF_0

Offset: 0x85c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_CTRL_INT_4_MASK**: MSI Interrupt4 Mask. Allows enabled interrupts to be masked. When an MSI is received for a masked interrupt, the corresponding status bit gets set in the interrupt status register but the msi_ctrl_int output is not set HIGH. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_4_STATUS_OFF_0

where <*i*> = 1, 4, 8.

Description: This register provides the MSI Interrupt4 status.

PCIE_X1_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_4_STATUS_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_4_STATUS_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_4_STATUS_OFF_0

Offset: 0x860
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_CTRL_INT_4_STATUS**: MSI Interrupt4 Status. When an MSI is detected for EP#i, one bit in this register is set. The decoding of the data payload of the MSI Memory Write request determines which bit gets set. A status is bit is cleared by writing a 1 to the bit. Each bit corresponds to a single MSI Interrupt Vector.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_5_EN_OFF_0

where <*i*> = 1, 4, 8.

Description: This register enables integrated MSI reception module interrupt5.

PCIE_X1_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_5_EN_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_5_EN_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_5_EN_OFF_0

Offset: 0x864
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_CTRL_INT_5_EN**: MSI Interrupt5 Enable. Specifies which interrupts are enabled. When an MSI is received from a disabled interrupt, no status bit gets set in MSI controller interrupt status register. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_5_MASK_OFF_0

where <*i*> = 1, 4, 8.

Description: This register provides information regarding the MSI Interrupt5 mask.

PCIE_X1_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_5_MASK_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_5_MASK_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_5_MASK_OFF_0

Offset: 0x868
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_CTRL_INT_5_MASK**: MSI Interrupt5 Mask. Allows enabled interrupts to be masked. When an MSI is received for a masked interrupt, the corresponding status bit gets set in the interrupt status register but the msi_ctrl_int output is not set HIGH. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_5_STATUS_OFF_0

where <*i*> = 1, 4, 8.

Description: This register provides the MSI Interrupt5 status.

PCIE_X1_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_5_STATUS_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_5_STATUS_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_5_STATUS_OFF_0

Offset: 0x86c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_CTRL_INT_5_STATUS**: MSI Interrupt5 Status. When an MSI is detected for EP#i, one bit in this register is set. The decoding of the data payload of the MSI Memory Write request determines which bit gets set. A status is bit is cleared by writing a 1 to the bit. Each bit corresponds to a single MSI Interrupt Vector.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_6_EN_OFF_0

where <*i*> = 1, 4, 8.

Description: This register enables integrated MSI reception module interrupt6.

PCIE_X1_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_6_EN_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_6_EN_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_6_EN_OFF_0

Offset: 0x870
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_CTRL_INT_6_EN**: MSI Interrupt6 Enable. Specifies which interrupts are enabled. When an MSI is received from a disabled interrupt, no status bit gets set in MSI controller interrupt status register. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_6_MASK_OFF_0

where <*i*> = 1, 4, 8.

Description: This register provides information regarding the MSI Interrupt6 mask.

PCIE_X1_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_6_MASK_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_6_MASK_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_6_MASK_OFF_0

Offset: 0x874
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_CTRL_INT_6_MASK**: MSI Interrupt6 Mask. Allows enabled interrupts to be masked. When an MSI is received for a masked interrupt, the corresponding status bit gets set in the interrupt status register but the msi_ctrl_int output is not set HIGH. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_6_STATUS_OFF_0

where <*i*> = 1, 4, 8.

Description: This register provides the MSI Interrupt6 status.

PCIE_X1_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_6_STATUS_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_6_STATUS_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_6_STATUS_OFF_0

Offset: 0x878
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_CTRL_INT_6_STATUS**: MSI Interrupt6 Status. When an MSI is detected for EP#i, one bit in this register is set. The decoding of the data payload of the MSI Memory Write request determines which bit gets set. A status is bit is cleared by writing a 1 to the bit. Each bit corresponds to a single MSI Interrupt Vector.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_7_EN_OFF_0

where <*i*> = 1, 4, 8.

Description: This register enables integrated MSI reception module interrupt7.

PCIE_X1_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_7_EN_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_7_EN_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_7_EN_OFF_0

Offset: 0x87c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_CTRL_INT_7_EN**: MSI Interrupt7 Enable. Specifies which interrupts are enabled. When an MSI is received from a disabled interrupt, no status bit gets set in MSI controller interrupt status register. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_7_MASK_OFF_0

where <*i*> = 1, 4, 8.

Description: This register provides information regarding the MSI Interrupt7 mask.

PCIE_X1_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_7_MASK_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_7_MASK_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_7_MASK_OFF_0

Offset: 0x880
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_CTRL_INT_7_MASK**: MSI Interrupt7 Mask. Allows enabled interrupts to be masked. When an MSI is received for a masked interrupt, the corresponding status bit gets set in the interrupt status register but the msi_ctrl_int output is not set HIGH. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_7_STATUS_OFF_0

where <*i*> = 1, 4, 8.

Description: This register provides the MSI Interrupt7 status.

PCIE_X1_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_7_STATUS_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_7_STATUS_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSI_CTRL_INT_7_STATUS_OFF_0

Offset: 0x884
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_CTRL_INT_7_STATUS**: MSI Interrupt7 Status. When an MSI is detected for EP#i, one bit in this register is set. The decoding of the data payload of the MSI Memory Write request determines which bit gets set. A status is bit is cleared by writing a 1 to the bit. Each bit corresponds to a single MSI Interrupt Vector.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_MSI_GPIO_IO_OFF_0

where <*i*> = 1, 4, 8.

Description: The contents of this register drive the top-level GPIO msi_ctrl_io[31:0].

PCIE_X1_RC_PF0_PORT_LOGIC_MSI_GPIO_IO_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_MSI_GPIO_IO_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSI_GPIO_IO_OFF_0

Offset: 0x888
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSI_GPIO_REG**: MSI GPIO Register. The contents of this register drive the top-level GPIO msi_ctrl_io[31:0]. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_CLOCK_GATING_CTRL_OFF_0

where <*i*> = 1, 4, 8.

Description: This register enables you to disable dynamic clock gating. By default dynamic clock gating is on, allowing the controller to autonomously enable and disable its clocks. The clock gating is performed in the clock and reset module, DWC_pcie_clk_rst.v, and is initiated by the controllers clock enable signals. The following modules support dynamic clock gating: - AXI Bridge - RADM.

PCIE_X1_RC_PF0_PORT_LOGIC_CLOCK_GATING_CTRL_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_CLOCK_GATING_CTRL_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_CLOCK_GATING_CTRL_OFF_0

Offset: 0x88c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000003 (0b0000,0000,0000,0000,0000,0000,0000,0011)

Bit	R/W	Reset	Description
31:2	RO	0x0	**RSVDP_2**: Reserved for future use.
1	RW	0x1	**AXI_CLK_GATING_EN**: AXI Clock Gating Enable. This register enables the AXI Bridge to autonomously enable and disable the AXI Master clock, the AXI Slave clock and the AXI DBI slave clock. The DWC_pcie_clk_rst.v module provides the gated clock, mstr_axi_aclk_gated, to the AXI Bridge and is enabled when the controllers clock enable signal, mstr_aclk_active, is asserted. For the AXI Slave this module provides the gated clock, slv_axi_aclk_gated, to the AXI Bridge and is enabled when the controllers clock enable signal, slv_aclk_active, is asserted. If the AXI DBI Slave is enabled (DBI_4SLAVE_POPULATED=1) the module provides the gated clock, dbi_axi_aclk_gated, to the AXI Bridge and is enabled when the controllers clock enable signal, dbi_aclk_active, is asserted. The controller de-asserts the clock enable signals when the respective AXI Master/Slave interfaces are idle. Note: This register field is sticky.
0	RW	0x1	**RADM_CLK_GATING_EN**: RADM Clock Gating Enable. This register, if set, enables the RADM to autonomously enable and disable its clock. The DWC_pcie_clk_rst.v module provides the gated clock, radm_clk_g, to the RADM and is enabled when the controllers clock enable signal, en_radm_clk_g, is asserted. The RADM clock is a gated version of the controller clock, core_clk. The controller de-asserts en_radm_clk_g when there is no Rx traffic, Rx queues and pre/post-queue pipelines are empty, RADM completion LUT is empty, and there are no FLR actions pending. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_GEN3_RELATED_OFF_0

where <*i*> = 1, 4, 8.

Description: There is no Gen3-specific N_FTS field. The N_FTS field in the "Link Width and Speed Change Control Register" is used for both Gen2 and Gen3 speed modes. There is no Gen3-specific "Directed Speed Change" field. The "Directed Speed Change" field in the "Link Width and Speed Change Control Register" is used to change to Gen2 or Gen3 speed. A speed change to Gen3 occurs if (1) the "Directed Speed Change" field is set to '1' and (2) the "Target Link Speed" field in the Link Control 2 Register is set to Gen3. Gen3 support is advertised by both sides of the link during link training.

Offset: 0x890
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_PORT_LOGIC_GEN3_RELATED_OFF_0

Reset: 0x00400000 (0b0000,0000,0100,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:26	RO	0x0	**RSVDP_26**: Reserved for future use.
25:24	RW	0x0	**RATE_SHADOW_SEL**: Rate Shadow Select. This register value decide the Data Rate of shadow register. The following shadow registers are controlled by this register. - GEN3_RELATED_OFF[9] EQ_PHASE_2_3 - GEN3_RELATED_OFF[12] RXEQ_PH01_EN - GEN3_RELATED_OFF[19] RE_EQ_REQUEST_ENABLE - GEN3_RELATED_OFF[21] AUTO_EQ_DISABLE - GEN3_RELATED_OFF[22] USP_SEND_8GT_EQ_TS2_DISABLE - GEN3_EQ_LOCAL_FS_LF_OFF[5:0] GEN3_EQ_LOCAL_LF - GEN3_EQ_LOCAL_FS_LF_OFF[11:6] GEN3_EQ_LOCAL_FS - GEN3_EQ_PSET_COEFF_MAP_0[5:0] GEN3_EQ_PRE_CURSOR_PSET - GEN3_EQ_PSET_COEFF_MAP_0[11:6] GEN3_EQ_CURSOR_PSET - GEN3_EQ_PSET_COEFF_MAP_0[17:12] GEN3_EQ_POSET_CURSOR_PSET - GEN3_EQ_CONTROL_OFF[3:0] GEN3_EQ_FB_MODE - GEN3_EQ_CONTROL_OFF[4] GEN3_EQ_PHASE23_EXIT_MODE - GEN3_EQ_CONTROL_OFF[5] GEN3_EQ_EVAL_2MS_DISABLE - GEN3_EQ_CONTROL_OFF[23:8] GEN3_EQ_PSET_REQ_VEC - GEN3_EQ_CONTROL_OFF[24] GEN3_EQ_FOM_INC_INITIAL_EVAL - GEN3_EQ_CONTROL_OFF[25] GEN3_EQ_PSET_REQ_AS_COEF - GEN3_EQ_FB_MODE_DIR_CHANGE_OFF[4:0] GEN3_EQ_FMDC_T_MIN_PHASE23 - GEN3_EQ_FB_MODE_DIR_CHANGE_OFF[9:5] GEN3_EQ_FMDC_N_EVALS - GEN3_EQ_FB_MODE_DIR_CHANGE_OFF[13:10] GEN3_EQ_FMDC_MAX_PRE_CUSROR_DELTA - GEN3_EQ_FB_MODE_DIR_CHANGE_OFF[17:14] GEN3_EQ_FMDC_MAX_POST_CUSROR_DELTA Note: This register field is sticky.

Bit	R/W	Reset	Description
23	RW	0x0	**GEN3_EQ_INVREQ_EVAL_DIFF_DISABLE**: Eq InvalidRequest and RxEqEval Different Time Assertion Disable. Disable the assertion of Eq InvalidRequest and RxEqEval at different time. Note: When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shared for Gen3 and Gen4/Gen5 data rate. Note: This register field is sticky.
22	RW	0x1	**USP_SEND_8GT_EQ_TS2_DISABLE**: Upstream Port Send 8GT/s or 16GT/s EQ TS2 Disable. The base spec defines that USP can optionally send 8GT or 16GT EQ TS2, which implies that USP can set DSP TxPreset value in Gen4 or Gen5 Data Rate. This applies to upstream ports only; It does not apply to downstream ports. Note: - When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shadow register for Gen3 and Gen4/Gen5 data rate. - If RATE_SHADOW_SEL==00b, this register is RSVD. - If RATE_SHADOW_SEL==01b, this register is for Gen4 data rate. - If RATE_SHADOW_SEL==10b, this register is for Gen5 data rate. Value after reset in Gen4/Gen5 is 0x1. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: see description Note: This register field is sticky.
21	RW	0x0	**AUTO_EQ_DISABLE**: Autonomous Equalization Disable. Note: - When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shadow register for Gen3 and Gen4/Gen5 data rate. - If RATE_SHADOW_SEL==00b, this register is RSVD. - If RATE_SHADOW_SEL==01b, this register is for Gen4 data rate. - If RATE_SHADOW_SEL==10b, this register is for Gen5 data rate. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: see description Note: This register field is sticky.
20:19	RO	0x0	**RSVDP_19**: Reserved for future use.
18	RW	0x0	**GEN3_DC_BALANCE_DISABLE**: DC Balance Disable. Disable DC Balance feature. Note: When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shared for Gen3 and Gen4/Gen5 data rate. Note: This register field is sticky.
17	RW	0x0	**GEN3_DLLP_XMT_DELAY_DISABLE**: DLLP Transmission Delay Disable. Disable delay transmission of DLLPs before equalization. Note: When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shared for Gen3 and Gen4/Gen5 data rate. Note: This register field is sticky.
16	RW	0x0	**GEN3_EQUALIZATION_DISABLE**: Equalization Disable. Disable equalization feature. This bit cannot be changed once the LTSSM starts link training. Note: When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shared for Gen3 and Gen4/Gen5 data rate. Note: This register field is sticky.
15:14	RO	0x0	**RSVDP_14**: Reserved for future use.
13	RW	0x0	**RXEQ_RGRDLESS_RXTS**: When set to '1', the controller as Gen3 EQ master asserts RxEqEval to instruct the PHY to do Rx adaptation and evaluation after a 500ns timeout from a new preset request. Note: When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shadow register for Gen3 and Gen4/Gen5 data rate. If RATE_SHADOW_SEL==00b, this register is for Gen3 data rate. If RATE_SHADOW_SEL==01b, this register is for Gen4 data rate. If RATE_SHADOW_SEL==10b, this register is for Gen5 data rate. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: see description Note: This register field is sticky.

Bit	R/W	Reset	Description
12	RW	0x0	**RXEQ_PH01_EN**: Rx Equalization Phase 0/Phase 1 Hold Enable. When this bit is set the upstream port holds phase 0 (the downstream port holds phase 1) for 10ms. Holding phase 0 or phase 1 can be used to allow sufficient time for Rx Equalization to be performed by the PHY. This bit is used during Virtex-7 Gen3 equalization. The programmable bits [RXEQ_PH01_EN, EQ_PHASE_2_3] can be used to obtain the following variations of the equalization procedure: Note: When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shadow register for Gen3 and Gen4/Gen5 data rate. If RATE_SHADOW_SEL==00b, this register is for Gen3 data rate. If RATE_SHADOW_SEL==01b, this register is for Gen4 data rate. If RATE_SHADOW_SEL==10b, this register is for Gen5 data rate. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: see description Note: This register field is sticky.
11	RW	0x0	**EQ_REDO**: Equalization Redo Disable. Disable autonomous mechanism for requesting to redo the equalization process. The received presets or coefficients mismatch in Recovery.RcvrLock after Recovery EQ phases causes the EQ redo requests. If the EQ redo is infinite or you do not want eq requests and redo, setting this bit to 1 will stop the EQ requests and EQ redo so that the link can go ahead to L0 state for packet trasmissions. Note: When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shared for Gen3 and Gen4/Gen5 data rate. Note: This register field is sticky.
10	RW	0x0	**EQ_EIEOS_CNT**: Equalization EIEOS Count Reset Disable. Disable requesting reset of EIEOS count during equalization. Note: When CX_GEN4_SPEED/ CX_GEN5_SPEED, this register is shared for Gen3 and Gen4/Gen5 data rate. Note: This register field is sticky.
9	RW	0x0	**EQ_PHASE_2_3**: Equalization Phase 2 and Phase 3 Disable. This applies to downstream ports only. Note: When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shadow register for Gen3 and Gen4/Gen5 data rate. If RATE_SHADOW_SEL==00b, this register is for Gen3 data rate. If RATE_SHADOW_SEL==01b, this register is for Gen4 data rate. If RATE_SHADOW_SEL==10b, this register is for Gen5 data rate. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: see description Note: This register field is sticky.
8	RW	0x0	**DISABLE_SCRAMBLER_GEN_3**: Disable Scrambler for Gen3 and Gen4 Data Rate. The Gen3 and Gen4 scrambler/descrambler within the controller needs to be disabled when the scrambling function is implemented outside of the controller (for example within the PHY). Note: When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shared for Gen3 and Gen4/Gen5 data rate. Note: This register field is sticky.
7:2	RO	0x0	**RSVDP_2**: Reserved for future use.
1	RW	0x0	**NO_SEED_VALUE_CHANGE**: If this bit is set to 1, the seed value of LFSR for scrambler at Gen3 rate does not change after LinkUp = 1. This bit takes effect only when CX_AUTO_LANE_FLIP_CTRL_EN is supported. This feature requires both sides of the link support it. Note: this register is shared for Gen3 and Gen4/Gen5 data rates. Note: This register field is sticky.

Bit	R/W	Reset	Description
0	RW	0x0	**GEN3_ZRXDC_NONCOMPL**: Gen3 Receiver Impedance ZRX-DC Not Compliant. Receivers that operate at 8.0 GT/s with an impedance other than the range defined by the ZRX-DC parameter for 2.5 GT/s (40-60 Ohms) must meet additional behavior requirements in the following LTSSM states: Polling, Rx_L0s, L1, L2, and Disabled. Note: When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shared for Gen3 and Gen4/Gen5 data rates. Note: This register field is sticky.

PCIE_X4_RC_PF0_PORT_LOGIC_GEN3_RELATED_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_GEN3_RELATED_OFF_0

Reset: 0x00400002 (0b0000,0000,0100,0000,0000,0000,0000,0010)

Bit	R/W	Reset	Description
Bit	**R/W**	**Reset**	**Description**
31:26	RO	0x0	**RSVDP_26**: Reserved for future use.
25:24	RW	0x0	**RATE_SHADOW_SEL**: Rate Shadow Select. This register value decide the Data Rate of shadow register. The following shadow registers are controlled by this register. - GEN3_RELATED_OFF[9] EQ_PHASE_2_3 - GEN3_RELATED_OFF[12] RXEQ_PH01_EN - GEN3_RELATED_OFF[19] RE_EQ_REQUEST_ENABLE - GEN3_RELATED_OFF[21] AUTO_EQ_DISABLE - GEN3_RELATED_OFF[22] USP_SEND_8GT_EQ_TS2_DISABLE - GEN3_EQ_LOCAL_FS_LF_OFF[5:0] GEN3_EQ_LOCAL_LF - GEN3_EQ_LOCAL_FS_LF_OFF[11:6] GEN3_EQ_LOCAL_FS - GEN3_EQ_PSET_COEFF_MAP_0[5:0] GEN3_EQ_PRE_CURSOR_PSET - GEN3_EQ_PSET_COEFF_MAP_0[11:6] GEN3_EQ_CURSOR_PSET - GEN3_EQ_PSET_COEFF_MAP_0[17:12] GEN3_EQ_POSET_CURSOR_PSET - GEN3_EQ_CONTROL_OFF[3:0] GEN3_EQ_FB_MODE - GEN3_EQ_CONTROL_OFF[4] GEN3_EQ_PHASE23_EXIT_MODE - GEN3_EQ_CONTROL_OFF[5] GEN3_EQ_EVAL_2MS_DISABLE - GEN3_EQ_CONTROL_OFF[23:8] GEN3_EQ_PSET_REQ_VEC - GEN3_EQ_CONTROL_OFF[24] GEN3_EQ_FOM_INC_INITIAL_EVAL - GEN3_EQ_CONTROL_OFF[25] GEN3_EQ_PSET_REQ_AS_COEF - GEN3_EQ_FB_MODE_DIR_CHANGE_OFF[4:0] GEN3_EQ_FMDC_T_MIN_PHASE23 - GEN3_EQ_FB_MODE_DIR_CHANGE_OFF[9:5] GEN3_EQ_FMDC_N_EVALS - GEN3_EQ_FB_MODE_DIR_CHANGE_OFF[13:10] GEN3_EQ_FMDC_MAX_PRE_CUSROR_DELTA - GEN3_EQ_FB_MODE_DIR_CHANGE_OFF[17:14] GEN3_EQ_FMDC_MAX_POST_CUSROR_DELTA Note: This register field is sticky.
23	RW	0x0	**GEN3_EQ_INVREQ_EVAL_DIFF_DISABLE**: Eq InvalidRequest and RxEqEval Different Time Assertion Disable. Disable the assertion of Eq InvalidRequest and RxEqEval at different time. Note: When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shared for Gen3 and Gen4/Gen5 data rate. Note: This register field is sticky.

Bit	R/W	Reset	Description
Bit	**R/W**	**Reset**	**Description**
22	RW	0x1	**USP_SEND_8GT_EQ_TS2_DISABLE**: Upstream Port Send 8GT/s or 16GT/s EQ TS2 Disable. The base spec defines that USP can optionally send 8GT or 16GT EQ TS2, which implies that USP can set DSP TxPreset value in Gen4 or Gen5 Data Rate. This applies to upstream ports only; It does not apply to downstream ports. Note: - When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shadow register for Gen3 and Gen4/Gen5 data rate. - If RATE_SHADOW_SEL==00b, this register is RSVD. - If RATE_SHADOW_SEL==01b, this register is for Gen4 data rate. - If RATE_SHADOW_SEL==10b, this register is for Gen5 data rate. Value after reset in Gen4/Gen5 is 0x1. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: see description Note: This register field is sticky.
21	RW	0x0	**AUTO_EQ_DISABLE**: Autonomous Equalization Disable. Note: - When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shadow register for Gen3 and Gen4/Gen5 data rate. - If RATE_SHADOW_SEL==00b, this register is RSVD. - If RATE_SHADOW_SEL==01b, this register is for Gen4 data rate. - If RATE_SHADOW_SEL==10b, this register is for Gen5 data rate. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: see description Note: This register field is sticky.
20:19	RO	0x0	**RSVDP_19**: Reserved for future use.
18	RW	0x0	**GEN3_DC_BALANCE_DISABLE**: DC Balance Disable. Disable DC Balance feature. Note: When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shared for Gen3 and Gen4/Gen5 data rate. Note: This register field is sticky.
17	RW	0x0	**GEN3_DLLP_XMT_DELAY_DISABLE**: DLLP Transmission Delay Disable. Disable delay transmission of DLLPs before equalization. Note: When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shared for Gen3 and Gen4/Gen5 data rate. Note: This register field is sticky.
16	RW	0x0	**GEN3_EQUALIZATION_DISABLE**: Equalization Disable. Disable equalization feature. This bit cannot be changed once the LTSSM starts link training. Note: When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shared for Gen3 and Gen4/Gen5 data rate. Note: This register field is sticky.
15:14	RO	0x0	**RSVDP_14**: Reserved for future use.
13	RW	0x0	**RXEQ_RGRDLESS_RXTS**: When set to '1', the controller as Gen3 EQ master asserts RxEqEval to instruct the PHY to do Rx adaptation and evaluation after a 500ns timeout from a new preset request. Note: When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shadow register for Gen3 and Gen4/Gen5 data rate. If RATE_SHADOW_SEL==00b, this register is for Gen3 data rate. If RATE_SHADOW_SEL==01b, this register is for Gen4 data rate. If RATE_SHADOW_SEL==10b, this register is for Gen5 data rate. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: see description Note: This register field is sticky.

Bit	R/W	Reset	Description
Bit	**R/W**	**Reset**	**Description**
12	RW	0x0	**RXEQ_PH01_EN**: Rx Equalization Phase 0/Phase 1 Hold Enable. When this bit is set the upstream port holds phase 0 (the downstream port holds phase 1) for 10ms. Holding phase 0 or phase 1 can be used to allow sufficient time for Rx Equalization to be performed by the PHY. This bit is used during Virtex-7 Gen3 equalization. The programmable bits [RXEQ_PH01_EN, EQ_PHASE_2_3] can be used to obtain the following variations of the equalization procedure: Note: When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shadow register for Gen3 and Gen4/Gen5 data rate. If RATE_SHADOW_SEL==00b, this register is for Gen3 data rate. If RATE_SHADOW_SEL==01b, this register is for Gen4 data rate. If RATE_SHADOW_SEL==10b, this register is for Gen5 data rate. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: see description Note: This register field is sticky.
11	RW	0x0	**EQ_REDO**: Equalization Redo Disable. Disable autonomous mechanism for requesting to redo the equalization process. The received presets or coefficients mismatch in Recovery.RcvrLock after Recovery EQ phases causes the EQ redo requests. If the EQ redo is infinite or you do not want eq requests and redo, setting this bit to 1 will stop the EQ requests and EQ redo so that the link can go ahead to L0 state for packet trasmissions. Note: When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shared for Gen3 and Gen4/Gen5 data rate. Note: This register field is sticky.
10	RW	0x0	**EQ_EIEOS_CNT**: Equalization EIEOS Count Reset Disable. Disable requesting reset of EIEOS count during equalization. Note: When CX_GEN4_SPEED/ CX_GEN5_SPEED, this register is shared for Gen3 and Gen4/Gen5 data rate. Note: This register field is sticky.
9	RW	0x0	**EQ_PHASE_2_3**: Equalization Phase 2 and Phase 3 Disable. This applies to downstream ports only. Note: When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shadow register for Gen3 and Gen4/Gen5 data rate. If RATE_SHADOW_SEL==00b, this register is for Gen3 data rate. If RATE_SHADOW_SEL==01b, this register is for Gen4 data rate. If RATE_SHADOW_SEL==10b, this register is for Gen5 data rate. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: see description Note: This register field is sticky.
8	RW	0x0	**DISABLE_SCRAMBLER_GEN_3**: Disable Scrambler for Gen3 and Gen4 Data Rate. The Gen3 and Gen4 scrambler/descrambler within the controller needs to be disabled when the scrambling function is implemented outside of the controller (for example within the PHY). Note: When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shared for Gen3 and Gen4/Gen5 data rate. Note: This register field is sticky.
7:2	RO	0x0	**RSVDP_2**: Reserved for future use.
1	RW	0x1	**NO_SEED_VALUE_CHANGE**: If this bit is set to 1, the seed value of LFSR for scrambler at Gen3 rate does not change after LinkUp = 1. This bit takes effect only when CX_AUTO_LANE_FLIP_CTRL_EN is supported. This feature requires both sides of the link support it. Note: this register is shared for Gen3 and Gen4/Gen5 data rates. Note: This register field is sticky.

Bit	R/W	Reset	Description
Bit	**R/W**	**Reset**	**Description**
0	RW	0x0	**GEN3_ZRXDC_NONCOMPL**: Gen3 Receiver Impedance ZRX-DC Not Compliant. Receivers that operate at 8.0 GT/s with an impedance other than the range defined by the ZRX-DC parameter for 2.5 GT/s (40-60 Ohms) must meet additional behavior requirements in the following LTSSM states: Polling, Rx_L0s, L1, L2, and Disabled. Note: When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shared for Gen3 and Gen4/Gen5 data rates. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_GEN3_EQ_CONTROL_OFF_0

where <*i*> = 1, 4, 8.

Description: This register controls equalization for Phase2 in an upstream port (USP), or Phase3 in a downstream port (DSP).

PCIE_X1_RC_PF0_PORT_LOGIC_GEN3_EQ_CONTROL_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_GEN3_EQ_CONTROL_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_GEN3_EQ_CONTROL_OFF_0

Offset: 0x8a8
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x05000f60 (0b0000,0101,0000,0000,0000,1111,0110,0000)

Bit	R/W	Reset	Description
31:27	RO	0x0	**RSVDP_27**: Reserved for future use.
26	RW	0x1	**GEN3_REQ_SEND_CONSEC_EIEOS_FOR_PSET_MAP**: Request controller to send back-to-back EIEOS in Recovery.RcvrLock state until presets to coefficients mapping is complete. Note: Gen3 and Gen4 share the same register bit and have the same feature. Note: This register field is sticky.
25	RW	0x0	**GEN3_EQ_PSET_REQ_AS_COEF**: GEN3_EQ_PSET_REQ_AS_COEF is an internally reserved field. Do not use. Note: This register field is sticky.

Bit	R/W	Reset	Description
24	RW	0x1	**GEN3_EQ_FOM_INC_INITIAL_EVAL**: Include Initial FOM. Include or not the FOM feedback from the initial preset evaluation performed in the EQ Master, when finding the highest FOM among all preset evaluations. Note: - When CX_GEN4_SPEED/ CX_GEN5_SPEED, this register is shadow register for Gen3 and Gen4/Gen5 data rate. - If RATE_SHADOW_SEL==00b, this register is for Gen3 data rate. - If RATE_SHADOW_SEL==01b, this register is for Gen4 data rate. - If RATE_SHADOW_SEL==10b, this register is for Gen5 data rate. Note: This register field is sticky.
23:8	RW	0xf	**GEN3_EQ_PSET_REQ_VEC**: Preset Request Vector. Requesting of Presets during the initial part of the EQ Master Phase. Encoding scheme is as follows: Bit [15:0] =0x0: No preset is requested and evaluated in EQ Master Phase. Bit [i] =1: "Preset=i" is requested and evaluated in EQ Master Phase. - 0000000000000000: No preset be requested and evaluated in EQ Master Phase - 000000xxxxxxxxx1: Preset 0 is requested and evaluated in EQ Master Phase - 000000xxxxxxxx1x: Preset 1 is requested and evaluated in EQ Master Phase - 000000xxxxxxx1xx: Preset 2 is requested and evaluated in EQ Master Phase - 000000xxxxxx1xxx: Preset 3 is requested and evaluated in EQ Master Phase - 000000xxxxx1xxxx: Preset 4 is requested and evaluated in EQ Master Phase - 000000xxxx1xxxxx: Preset 5 is requested and evaluated in EQ Master Phase - 000000xxx1xxxxxx: Preset 6 is requested and evaluated in EQ Master Phase - 000000xx1xxxxxxx: Preset 7 is requested and evaluated in EQ Master Phase - 000000x1xxxxxxxx: Preset 8 is requested and evaluated in EQ Master Phase - 00000x1xxxxxxxxx: Preset 9 is requested and evaluated in EQ Master Phase - 000001xxxxxxxxxx: Preset 10 is requested and evaluated in EQ Master Phase - All other encodings: Reserved Note: You must contact your PHY vendor to ensure 24 ms timeout does not occur in presets requests in EQ master phase, that is, you must set a proper value to the GEN3_EQ_PSET_REQ_VEC register so that the EQ tunning for Figure of Merit in the EQ master phase completes before 24 ms timeout. Note: - When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shadow register for Gen3 and Gen4/Gen5 data rate. - If RATE_SHADOW_SEL==00b, this register is for Gen3 data rate. - If RATE_SHADOW_SEL==01b, this register is for Gen4 data rate. - If RATE_SHADOW_SEL==10b, this register is for Gen5 data rate. Note: This register field is sticky.
7	RO	0x0	**RSVDP_7**: Reserved for future use.
6	RW	0x1	**GEN3_LOWER_RATE_EQ_REDO_ENABLE**: Support EQ redo and lower rate change. Note: Gen3 and Gen4 share the same register bit and have the same feature. Note: This register field is sticky.
5	RW	0x1	**GEN3_EQ_EVAL_2MS_DISABLE**: Phase2_3 2 ms Timeout Disable. Determine behavior in Phase2 for USP (Phase3 if DSP) when the PHY does not respond within 2ms to the assertion of RxEqEval. Note: - When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shadow register for Gen3 and Gen4/Gen5 data rate. - If RATE_SHADOW_SEL==00b, this register is for Gen3 data rate. - If RATE_SHADOW_SEL==01b, this register is for Gen4 data rate. - If RATE_SHADOW_SEL==10b, this register is for Gen5 data rate. Note: This register field is sticky.

Bit	R/W	Reset	Description
4	RW	0x0	**GEN3_EQ_PHASE23_EXIT_MODE**: Behavior After 24 ms Timeout (when optimal settings are not found). For a USP: Determine next LTSSM state from Phase2 after 24ms Timeout - 0: Recovery.Speed - 1: Recovery.Equalization.Phase3 When optimal settings are not found then: - Equalization Phase 2 Successful status bit is not set in the "Link Status Register 2" when GEN3_EQ_PHASE23_EXIT_MODE = 0 - Equalization Phase 2 Successful status bit is set in the "Link Status Register 2" when GEN3_EQ_PHASE23_EXIT_MODE = 1 - Equalization Phase 2 Complete status bit is set in the "Link Status Register 2" For a DSP: Determine next LTSSM state from Phase3 after 24ms Timeout - 0: Recovery.Speed - 1: Recovery.Equalization.RcvrLock When optimal settings are not found then: - Equalization Phase 3 Successful status bit is not set in the "Link Status Register 2" when GEN3_EQ_PHASE23_EXIT_MODE = 0 - Equalization Phase 3 Successful status bit is set in the "Link Status Register 2" when GEN3_EQ_PHASE23_EXIT_MODE = 1 - Equalization Phase 3 Complete status bit is set in the "Link Status Register 2" Note: GEN3_EQ_PHASE23_EXIT_MODE = 1 affects Direction Change feed back mode. EQ requests for Figure Of Merit mode complete before 24 ms timeout. Please see GEN3_EQ_PSET_REQ_VEC Register for more. Note: - When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shadow register for Gen3 and Gen4/Gen5 data rate. - If RATE_SHADOW_SEL==00b, this register is for Gen3 data rate. - If RATE_SHADOW_SEL==01b, this register is for Gen4 data rate. - If RATE_SHADOW_SEL==10b, this register is for Gen5 data rate. Note: This register field is sticky.
3:0	RW	0x0	**GEN3_EQ_FB_MODE**: Feedback Mode. Other values are reserved. Note: - When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is a shadow register for Gen3 and Gen4/Gen5 data rate. - If RATE_SHADOW_SEL==00b, this register is for Gen3 data rate. - If RATE_SHADOW_SEL==01b, this register is for Gen4 data rate. - If RATE_SHADOW_SEL==10b, this register is for Gen5 data rate. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_ORDER_RULE_CTRL_OFF_0

where <*i*> = 1, 4, 8.

Description: This register controls the order rule.

PCIE_X1_RC_PF0_PORT_LOGIC_ORDER_RULE_CTRL_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_ORDER_RULE_CTRL_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_ORDER_RULE_CTRL_OFF_0

Offset: 0x8b4
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:16	RO	0x0	**RSVDP_16**: Reserved for future use.
15:8	RW	0x0	**CPL_PASS_P**: Completion Passing Posted Ordering Rule Control. Determines if CPL can pass halted P queue.
7:0	RW	0x0	**NP_PASS_P**: Non-Posted Passing Posted Ordering Rule Control. Determines if NP can pass halted P queue.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_PIPE_LOOPBACK_CONTROL_OFF_0

where <*i*> = 1, 4, 8.

Description: This register controls the PIPE Loopback.

Offset: 0x8b8
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_PORT_LOGIC_PIPE_LOOPBACK_CONTROL_OFF_0

Reset: 0x00000001 (0b0000,0000,0000,0000,0000,0000,0000,0001)

Bit	R/W	Reset	Description
31	RW	0x0	**PIPE_LOOPBACK**: PIPE Loopback Enable. Indicates RMMI Loopback if M-PCIe. Note: This register field is sticky.
30:27	RO	0x0	**RSVDP_27**: Reserved for future use.
26:24	WO	0x0	**RXSTATUS_VALUE**: RXSTATUS_VALUE is an internally reserved field. Do not use.
23:22	RO	0x0	**RSVDP_22**: Reserved for future use.
21:16	RW	0x0	**RXSTATUS_LANE**: RXSTATUS_LANE is an internally reserved field. Do not use. Note: This register field is sticky.
15:0	RW	0x1	**LPBK_RXVALID**: LPBK_RXVALID is an internally reserved field. Do not use. Note: This register field is sticky.

PCIE_X4_RC_PF0_PORT_LOGIC_PIPE_LOOPBACK_CONTROL_OFF_0

Reset: 0x0000000f (0b0000,0000,0000,0000,0000,0000,0000,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**PIPE_LOOPBACK**: PIPE Loopback Enable. Indicates RMMI Loopback if M-PCIe. Note: This register field is sticky.
30:27	RO	0x0	**RSVDP_27**: Reserved for future use.
26:24	WO	0x0	**RXSTATUS_VALUE**: RXSTATUS_VALUE is an internally reserved field. Do not use.
23:22	RO	0x0	**RSVDP_22**: Reserved for future use.
21:16	RW	0x0	**RXSTATUS_LANE**: RXSTATUS_LANE is an internally reserved field. Do not use. Note: This register field is sticky.
15:0	RW	0xf	**LPBK_RXVALID**: LPBK_RXVALID is an internally reserved field. Do not use. Note: This register field is sticky.

PCIE_X8_RC_PF0_PORT_LOGIC_PIPE_LOOPBACK_CONTROL_OFF_0

Reset: 0x000000ff (0b0000,0000,0000,0000,0000,0000,1111,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**PIPE_LOOPBACK**: PIPE Loopback Enable. Indicates RMMI Loopback if M-PCIe. Note: This register field is sticky.
30:27	RO	0x0	**RSVDP_27**: Reserved for future use.
26:24	WO	0x0	**RXSTATUS_VALUE**: RXSTATUS_VALUE is an internally reserved field. Do not use.
23:22	RO	0x0	**RSVDP_22**: Reserved for future use.
21:16	RW	0x0	**RXSTATUS_LANE**: RXSTATUS_LANE is an internally reserved field. Do not use. Note: This register field is sticky.
15:0	RW	0xff	**LPBK_RXVALID**: LPBK_RXVALID is an internally reserved field. Do not use. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_MISC_CONTROL_1_OFF_0

where <*i*> = 1, 4, 8.

Description: This is the DBI Read-Only write enable register.

PCIE_X1_RC_PF0_PORT_LOGIC_MISC_CONTROL_1_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_MISC_CONTROL_1_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MISC_CONTROL_1_OFF_0

Offset: 0x8bc
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000bff49 (0b0000,0000,0000,1011,1111,1111,0100,1001)

Bit	R/W	Reset	Description
31:23	RO	0x0	**RSVDP_23**: Reserved for future use.
22	RW	0x0	**PORT_LOGIC_WR_DISABLE**: Disable port logic register write from wire side. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky) Note: This register field is sticky.
21	RW	0x0	**P2P_ERR_RPT_CTRL**: Determines whether to enable Peer to Peer (P2P) error reporting. Note: This register field is sticky.
20	RW	0x0	**P2P_TRACK_CPL_TO_REG**: Determines whether to track completion of transmitted Non-Posted TLPs in P2P mode. Note: This register field is sticky.
19:18	RW	0x2	**TARGET_ABOVE_CONFIG_LIMIT_REG**: Configuration requests with an address greater than CONFIG_LIMIT_REG are directed to either ELBI or TRGT1 interface based on the setting of this field. This field can have the following values: Note: This register field is sticky.
17:8	RW	0x3ff	**CONFIG_LIMIT_REG**: Configuration requests are directed either to CDM or ELBI/RTRGT1 based on the value of this field. - Configuration requests with an address less CONFIG_LIMIT_REG are directed to the CDM - Configuration requests with an address greater than CONFIG_LIMIT_REG are directed to either ELBI or TRGT1 interface based on the setting of TARGET_ABOVE_CONFIG_LIMIT_REG field. Your application must set a proper value for this field based on your extended configuration registers. For more details, see the "CDM/ELBI Register Space Access Through CFG Request" in "Register Module, LBC, and DBI" section in the "Controller Operations" chapter of the Databook. Note: This register field is sticky.
7	RW	0x0	**CFG_TLP_BYPASS_EN_REG**: Setting of this field defines how to decide the destination of Configuration requests. Note: When app_req_retry_en is asserted, the setting of this field is ignored. Note: This register field is sticky.

Bit	R/W	Reset	Description
6	RW	0x1	**CPLQ_MNG_EN**: This field enables the Completion Queue Management feature. Note: This register field is sticky.
5	RW	0x0	**ARI_DEVICE_NUMBER**: When ARI is enabled, this field enables use of the device ID. Note: This register field is sticky.
4	RW	0x0	**DISABLE_AUTO_LTR_CLR_MSG**: Disable the autonomous generation of LTR clear message in upstream port. This field can have the following values: For more details, see "Latency Tolerance Reporting (LTR) Message Generation [EP Mode]" in "Message Generation" section of the "Controller Operations" chapter of the Databook. Note: This register field is sticky.
3	RW	0x1	**SIMPLIFIED_REPLAY_TIMER**: Enables Simplified Replay Timer (Gen4). For more details, see "Transmit Replay" in "Transmit TLP Processing" section in the "Controller Operations" chapter of the Databook. Simplified Replay Timer can have the following Values: - A value from 24,000 to 31,000 Symbol Times when Extended Synch is 0b. - A value from 80,000 to 100,000 Symbol Times when Extended Synch is 1b. The Simplified Replay Timer value must not be changed while the link is in use. Note: This register field is sticky.
2	RW	0x0	**UR_CA_MASK_4_TRGT1**: When this field is set to '1', the controller suppresses error logging, error message generation, and CPL generation for non-posted requests TLPs (with UR filtering status) forwarded to your application (that is, when DEFAULT_TARGET =1). For more details, see "Advanced Error Handling For Received TLPs" chapter of the Databook. Note: This register field is sticky.
1	RW	0x0	**DEFAULT_TARGET**: Default target for an IO or MEM request with UR/CA/CRS received. Based on the value of this field the controller either drops or forwards these requests to your application. For more details, see "ECRC Handling" and "Request TLP Routing Rules" in "Receive Routing" section of the "Controller Operations" chapter of the Databook. Note: This register field is sticky.
0	RW	0x1	**DBI_RO_WR_EN**: Write to RO Registers Using DBI. For more details, see "Writing to Read-Only Registers" in "Register Module, LBC, and DBI" section in the "Controller Operations" chapter of the Databook. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_MULTI_LANE_CONTROL_OFF_0

where <*i*> = 1, 4, 8.

Description: Used when upsizing or downsizing the link width through Configuration state without bringing the link down. For more details, see the "Link Establishment" section in the "Controller Operations" chapter of the Databook.

PCIE_X1_RC_PF0_PORT_LOGIC_MULTI_LANE_CONTROL_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_MULTI_LANE_CONTROL_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MULTI_LANE_CONTROL_OFF_0

Offset: 0x8c0
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:8	RO	0x0	**RSVDP_8**: Reserved for future use.
7	RW	0x0	**UPCONFIGURE_SUPPORT**: Upconfigure Support. The controller sends this value as the Link Upconfigure Capability in TS2 Ordered Sets in Configuration.Complete state. Note: This register field is sticky.
6	RW	0x0	**DIRECT_LINK_WIDTH_CHANGE**: Directed Link Width Change. - If the upconfigure_capable variable is '1' and the PCIE_CAP_HW_AUTO_WIDTH_DISABLE bit in LINK_CONTROL_LINK_STATUS_REG is '0', the controller starts upconfigure or autonomous width downsizing (to the TARGET_LINK_WIDTH value) in the Configuration state. - If TARGET_LINK_WIDTH value is 0x0, the controller does not start upconfigure or autonomous width downsizing in the Configuration state. The controller self-clears this field when the controller accepts this request.
5:0	RW	0x0	**TARGET_LINK_WIDTH**: Target Link Width.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_PHY_INTEROP_CTRL_OFF_0

where <*i*> = 1, 4, 8.

Description: This register controls the PHY interoperability.

PCIE_X1_RC_PF0_PORT_LOGIC_PHY_INTEROP_CTRL_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_PHY_INTEROP_CTRL_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_PHY_INTEROP_CTRL_OFF_0

Offset: 0x8c4
Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x0001e27f (0b0000,0000,0000,0001,1110,x010,0111,1111)

Bit	R/W	Reset	Description
31:18	RO	0x0	**RSVDP_18**: Reserved for future use.
17:12	RW	0x1e	**PHY_RST_TIMER**: Decide how many aux clock cycles the PHY reset lasts (0 to 63 aux clock cycles). Note: This register field is sticky.
10	RW	0x0	**L1_CLK_SEL**: L1 Clock control bit. This field is reserved for internal use. This register field is sticky.
9	RO	0x1	**L1_NOWAIT_P1**: L1 entry control bit. This field is reserved for internal use. The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W (sticky). This register field is sticky.
8	RW	0x0	**L1SUB_EXIT_MODE**: L1 Exit Control Using phy_mac_pclkack_n. This field is reserved for internal use. Note: This register field is sticky.
7	RO	0x0	**RSVDP_7**: Reserved for future use.
6:0	RW	0x7f	**RXSTANDBY_CONTROL**: Rxstandby Control. Bits 0..5 determine if the controller asserts the RxStandby signal (mac_phy_rxstandby) in the indicated condition. Bit 6 enables the controller to perform the RxStandby/ RxStandbyStatus handshake. This field is reserved for internal use. [0]: Rx EIOS and subsequent T TX-IDLE-MIN [1]: Rate Change [2]: Inactive lane for upconfigure/downconfigure [3]: PowerDown=P1orP2 [4]: RxL0s.Idle [5]: EI Infer in L0 [6]: Execute RxStandby/RxStandbyStatus Handshake Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_TRGT_CPL_LUT_DELETE_ENTRY_OFF_0

where <*i*> = 1, 4, 8.

Description: Using this register you can delete one entry in the target completion LUT. You should only use this register when you know that your application will never send the completion because of an FLR or any other reason. Note:: The target completion LUT (and associated target completion timeout event) is watching for completions (from your application on XALI0/1/2 or AXI master read channel) corresponding to previously received non-posted requests from the PCIe wire.

PCIE_X1_RC_PF0_PORT_LOGIC_TRGT_CPL_LUT_DELETE_ENTRY_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_TRGT_CPL_LUT_DELETE_ENTRY_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_TRGT_CPL_LUT_DELETE_ENTRY_OFF_0

Offset: 0x8c8
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	WO	0x0	**DELETE_EN**: This is a one-shot bit. This is a self-clearing register field. Reading from this register field always returns a '0'.
30:0	RW	0x0	**LOOK_UP_ID**: This number selects one entry to delete of the TRGT_CPL_LUT.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_LINK_FLUSH_CONTROL_OFF_0

where <*i*> = 1, 4, 8.

Description: This register controls link reset request flush behaviour.

PCIE_X1_RC_PF0_PORT_LOGIC_LINK_FLUSH_CONTROL_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_LINK_FLUSH_CONTROL_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_LINK_FLUSH_CONTROL_OFF_0

Offset: 0x8cc
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xff000001 (0b1111,1111,0000,0000,0000,0000,0000,0001)

Bit	R/W	Reset	Description
31:24	RW	0xff	**RSVD_I_8**: This is an internally reserved field. Do not use. Note: This register field is sticky.
23:1	RO	0x0	**RSVDP_1**: Reserved for future use.

Bit	R/W	Reset	Description
0	RW	0x1	**AUTO_FLUSH_EN**: Enables automatic flushing of pending requests before sending the reset request to the application logic to reset the PCIe controller and the AXI Bridge. The flushing process is initiated if any of the following events occur: - Hot reset request. A downstream port (DSP) can "hot reset" an upstream port (USP) by sending two consecutive TS1 ordered sets with the hot reset bit asserted. - Warm (Soft) reset request. Generated when exiting from D3 to D0 and cfg_pm_no_soft_rst=0. - Link down reset request. A high to low transition on smlh_req_rst_not indicates the link has gone down and the controller is requesting a reset. If you disable automatic flushing, your application is responsible for resetting the PCIe controller and the AXI Bridge. For more details see "Warm and Hot Resets" section in the Architecture chapter of the Databook. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_AMBA_ERROR_RESPONSE_DEFAULT_OFF_0

where <*i*> = 1, 4, 8.

Description: This register holds the AXI bridge slave error responses.

PCIE_X1_RC_PF0_PORT_LOGIC_AMBA_ERROR_RESPONSE_DEFAULT_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_AMBA_ERROR_RESPONSE_DEFAULT_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_AMBA_ERROR_RESPONSE_DEFAULT_OFF_0

Offset: 0x8d0
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00009c00 (0b0000,0000,0000,0000,1001,1100,0000,0000)

Bit	R/W	Reset	Description
31:16	RO	0x0	**RSVDP_16**: Reserved for future use.
15:10	RW	0x27	**AMBA_ERROR_RESPONSE_MAP**: AXI Slave Response Error Map. Allows you to selectively map the errors received from the PCIe completion (for non-posted requests) to the AXI slave responses, slv_rresp or slv_bresp. The recommended setting is SLVERR. CRS is always mapped to OKAY. The controller sets the AXI slave read databus to 0xFFFF for all error responses. Note: This register field is sticky.
9:5	RO	0x0	**RSVDP_5**: Reserved for future use.

Bit	R/W	Reset	Description
4:3	RW	0x0	**AMBA_ERROR_RESPONSE_CRS**: CRS Slave Error Response Mapping. Determines the AXI slave response for CRS completions. For more details see "Error Handling" in the AXI chapter of the Databook. Note: This register field is sticky.
2	RW	0x0	**AMBA_ERROR_RESPONSE_VENDORID**: Vendor ID Non-existent Slave Error Response Mapping. Determines the AXI slave response for errors on reads to non-existent Vendor ID register. For more details see "Error Handling" in the AXI chapter of the Databook. Note: This register field is sticky.
1	RO	0x0	**RSVDP_1**: Reserved for future use.
0	RW	0x0	**AMBA_ERROR_RESPONSE_GLOBAL**: Global Slave Error Response Mapping. Determines the AXI slave response for all error scenarios on non-posted requests. For more details see "Error Handling" in the AXI chapter of the Databook. The error response mapping is not applicable to Non-existent Vendor ID register reads. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_AMBA_LINK_TIMEOUT_OFF_0

where <*i*> = 1, 4, 8.

Description: If your application AXI master issues outbound requests to the AXI bridge slave interface before the PCIe link is operational, the controller starts a "flush" timer. The timeout value of the timer is set by this register. If the timer times out before the PCIe link is operational, the bridge TX request queues are flushed. For more details, see the "AXI Bridge Initialization, Clocking and Reset" section in the AXI chapter of the Databook.

PCIE_X1_RC_PF0_PORT_LOGIC_AMBA_LINK_TIMEOUT_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_AMBA_LINK_TIMEOUT_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_AMBA_LINK_TIMEOUT_OFF_0

Offset: 0x8d4
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000032 (0b0000,0000,0000,0000,0000,0000,0011,0010)

Bit	R/W	Reset	Description
31:9	RO	0x0	**RSVDP_9**: Reserved for future use.

Bit	R/W	Reset	Description
8	RW	0x0	**LINK_TIMEOUT_ENABLE_DEFAULT**: Disable Flush. Note: This register field is sticky.
7:0	RW	0x32	**LINK_TIMEOUT_PERIOD_DEFAULT**: Timeout Value (ms). The timer will timeout and then flush the bridge TX request queues after this amount of time. The timer counts when there are pending outbound AXI slave interface requests and the PCIe TX link is not transmitting any of these requests. The timer is clocked by core_clk. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_AMBA_ORDERING_CTRL_OFF_0

where <*i*> = 1, 4, 8.

Description: This register controls the AXI Bridge Ordering.

PCIE_X1_RC_PF0_PORT_LOGIC_AMBA_ORDERING_CTRL_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_AMBA_ORDERING_CTRL_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_AMBA_ORDERING_CTRL_OFF_0

Offset: 0x8d8
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:8	RO	0x0	**RSVDP_8**: Reserved for future use.
7	RW	0x0	**AX_MSTR_ZEROLREAD_FW**: AXI Master Zero Length Read Forward to the application. The DW PCIe controller AXI bridge is able to terminate in order with the Posted transactions the zero length read, implementing the PCIe express flush semantics of the Posted transactions. Note: This register field is sticky.
6:5	RO	0x0	**RSVDP_5**: Reserved for future use.

Bit	R/W	Reset	Description
4:3	RW	0x0	**AX_MSTR_ORDR_P_EVENT_SEL**: AXI Master Posted Ordering Event Selector. This field selects how the master interface determines when a P write is completed when enforcing the PCIe ordering rule, "NP must not pass P" at the AXI Master Interface. The AXI protocol does not support ordering between channels. Therefore, NP reads can pass P on your AXI bus fabric. This can result in an ordering violation when the read overtakes a P that is going to the same address. Therefore, the bridge master does not issue any NP requests until all outstanding P writes reach their destination. It does this by waiting for the all of the write responses on the B channel. This can affect the performance of the master read channel. For scenarios where the interconnect serializes the AXI master "AW", "W" and "AR" channels,you can increase the performance by reducing the need to wait until the complete Posted transaction has effectively reached the application slave. Note: This setting will not affect: - MSI interrupt catcher and P data ordering. This is always driven by the B'last event. - DMA read engine TLP ordering. This is always driven by the B'last event. - NP write transactions which are always serialized with P write transactions.Note: This register field is sticky.
2	RO	0x0	**RSVDP_2**: Reserved for future use.
1	RW	0x0	**AX_SNP_EN**: AXI Serialize Non-Posted Requests Enable. This field enables the AXI Bridge to serialize same ID Non-Posted Read/Write Requests on the wire. Serialization implies one outstanding same ID NP Read or Write on the wire and used to avoid AXI RAR and WAW hazards at the remote link partner. For more details, see the "Optional Serialization of AXI Slave Non-posted Requests" section in the AXI chapter of the Databook. Note: This register field is sticky.
0	RO	0x0	**RSVDP_0**: Reserved for future use.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_COHERENCY_CONTROL_1_OFF_0

where <*i*> = 1, 4, 8.

Description: This register controls the ACE cache coherency operation.

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

PCIE_X1_RC_PF0_PORT_LOGIC_COHERENCY_CONTROL_1_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_COHERENCY_CONTROL_1_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_COHERENCY_CONTROL_1_OFF_0

Offset: 0x8e0

Bit	R/W	Reset	Description
31:2	RW	0x0	**CFG_MEMTYPE_BOUNDARY_LOW_ADDR**: Boundary Lower Address For Memory Type. Bits [31:0] of dword-aligned address of the boundary for Memory type. The two lower address LSBs are '00'. Addresses up to but not including this value are in the lower address space region; addresses equal or greater than this value are in the upper address space region. Note: This register field is sticky.
1	RO	0x0	**RSVDP_1**: Reserved for future use.
0	RW	0x0	**CFG_MEMTYPE_VALUE**: Sets the memory type for the lower and upper parts of the address space: Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_COHERENCY_CONTROL_2_OFF_0

where <*i*> = 1, 4, 8.

Description: This register controls the ACE cache coherency operation.

PCIE_X1_RC_PF0_PORT_LOGIC_COHERENCY_CONTROL_2_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_COHERENCY_CONTROL_2_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_COHERENCY_CONTROL_2_OFF_0

Offset: 0x8e4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CFG_MEMTYPE_BOUNDARY_HIGH_ADDR**: Boundary Upper Address For Memory Type. Bits [63:32] of the 64-bit dword-aligned address of the boundary for Memory type. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_COHERENCY_CONTROL_3_OFF_0

where <*i*> = 1, 4, 8.

Description: This register controls the ACE cache coherency operation.

PCIE_X1_RC_PF0_PORT_LOGIC_COHERENCY_CONTROL_3_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_COHERENCY_CONTROL_3_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_COHERENCY_CONTROL_3_OFF_0

Offset: 0x8e8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bx000,0xxx,x000,0xxx,x000,0xxx,x000,0xxx)

Bit	Reset	Description
30:27	0x0	**CFG_MSTR_AWCACHE_VALUE**: Master Write CACHE Signal Value. Value of the individual bits in mstr_awcache when CFG_MSTR_AWCACHE_MODE is '1'. Note: Not applicable to message requests; for message requests the value of mstr_awcache is always '0000'. Note: This register field is sticky.
22:19	0x0	**CFG_MSTR_ARCACHE_VALUE**: Master Read CACHE Signal Value. Value of the individual bits in mstr_arcache when CFG_MSTR_ARCACHE_MODE is '1'. Note: This register field is sticky.
14:11	0x0	**CFG_MSTR_AWCACHE_MODE**: Master Write CACHE Signal Behavior. Defines how the individual bits in mstr_awcache are controlled. Note: for message requests the value of mstr_awcache is always "0000" regardless of the value of this bit. Note: This register field is sticky.
6:3	0x0	**CFG_MSTR_ARCACHE_MODE**: Master Read CACHE Signal Behavior. Defines how the individual bits in mstr_arcache are controlled. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_AXI_MSTR_MSG_ADDR_LOW_OFF_0

where <*i*> = 1, 4, 8.

Description: Lower 20 bits of the programmable AXI address to which Messages coming from wire are mapped. Bits [11:0] of the register are tied to zero for the address to be 4k-aligned. In previous releases, the third and fourth DWORDs of a message (Msg/MsgD) TLP header were delivered though the AXI master address bus (mstr_awaddr). These DWORDS are now supplied through the mstr_awmisc_info_hdr_34dw[63:0] output; and the value on mstr_awaddr is driven to the value you have programmed into the AXI_MSTR_MSG_ADDR_LOW_OFF and AXI_MSTR_MSG_ADDR_HIGH_OFF registers.

PCIE_X1_RC_PF0_PORT_LOGIC_AXI_MSTR_MSG_ADDR_LOW_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_AXI_MSTR_MSG_ADDR_LOW_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_AXI_MSTR_MSG_ADDR_LOW_OFF_0

Offset: 0x8f0
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:12	RW	0x0	**CFG_AXIMSTR_MSG_ADDR_LOW**: Lower 20-bits of the programmable AXI address for Messages. Note: This register field is sticky.
11:0	RO	0x0	**CFG_AXIMSTR_MSG_ADDR_LOW_RESERVED**: Reserved for future use. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_AXI_MSTR_MSG_ADDR_HIGH_OFF_0

where <*i*> = 1, 4, 8.

Description: Upper 32 bits of the programmable AXI address to which Messages coming from wire are mapped.

PCIE_X1_RC_PF0_PORT_LOGIC_AXI_MSTR_MSG_ADDR_HIGH_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_AXI_MSTR_MSG_ADDR_HIGH_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_AXI_MSTR_MSG_ADDR_HIGH_OFF_0

Offset: 0x8f4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CFG_AXIMSTR_MSG_ADDR_HIGH**: Upper 32 bits of the programmable AXI address for Messages. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_PCIE_VERSION_NUMBER_OFF_0

where <*i*> = 1, 4, 8.

Description: The version number is given in hex format. You should convert each pair of hex characters to ASCII to interpret. Using 4.70a (GA) as an example: - VERSION_NUMBER = 0x3437302a which translates to 470* - VERSION_TYPE = 0x67612a2a which translates to ga** Using 4.70a-ea01 as an example: - VERSION_NUMBER = 0x3437302a which translates to 470* - VERSION_TYPE = 0x65613031 which translates to ea01 GA is a general release available on www.designware.com EA is an early release available on a per-customer basis.

PCIE_X1_RC_PF0_PORT_LOGIC_PCIE_VERSION_NUMBER_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_PCIE_VERSION_NUMBER_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_PCIE_VERSION_NUMBER_OFF_0

Offset: 0x8f8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x3536322a (0b0011,0101,0011,0110,0011,0010,0010,1010)

Bit	Reset	Description
31:0	0x3536322a	**VERSION_NUMBER**: Version Number.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_PCIE_VERSION_TYPE_OFF_0

where <*i*> = 1, 4, 8.

Description: The type is given in hex format. You should convert each pair of hex characters to ASCII to interpret. Using 4.70a (GA) as an example: - VERSION_NUMBER = 0x3437302a which translates to 470* - VERSION_TYPE = 0x67612a2a which translates to ga** Using 4.70a-ea01 as an example: - VERSION_NUMBER = 0x3437302a which translates to 470* - VERSION_TYPE = 0x65613031 which translates to ea01 GA is a general release available on www.designware.com EA is an early release available on a per-customer basis.

PCIE_X1_RC_PF0_PORT_LOGIC_PCIE_VERSION_TYPE_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_PCIE_VERSION_TYPE_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_PCIE_VERSION_TYPE_OFF_0

Offset: 0x8fc
Read/Write: RO

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x65613035 (0b0110,0101,0110,0001,0011,0000,0011,0101)

Bit	Reset	Description
31:0	0x65613035	**VERSION_TYPE**: Version Type.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_INTERFACE_TIMER_CONTROL_OFF_0

where <*i*> = 1, 4, 8.

Description: This register controls the interface timer.

PCIE_X1_RC_PF0_PORT_LOGIC_INTERFACE_TIMER_CONTROL_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_INTERFACE_TIMER_CONTROL_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_INTERFACE_TIMER_CONTROL_OFF_0

Offset: 0x930
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:5	RO	0x0	**RSVDP_5**: Reserved for future use.
4	RW	0x0	**FORCE_PENDING**: Writing to this bit forces the value of the pending flags. Note: This register field is sticky.
3:2	RW	0x0	**INTERFACE_TIMER_SCALING**: Interface timer scaling. This field can be used to reduce the timer duration for verification purpose. This field should only be programmed when the INTERFACE_TIMER_EN bit is set to 1'b0. Note: This register field is sticky.
1	RW	0x0	**INTERFACE_TIMER_AER_EN**: When set to 1 the Interface timer internal uncorrectable error generation is enabled. Note: This register field is sticky.
0	RW	0x0	**INTERFACE_TIMER_EN**: Interface timer enable. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_INTERFACE_TIMER_TARGET_OFF_0

where <*i*> = 1, 4, 8.

Description: This is the interface timer target register.

PCIE_X1_RC_PF0_PORT_LOGIC_INTERFACE_TIMER_TARGET_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_INTERFACE_TIMER_TARGET_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_INTERFACE_TIMER_TARGET_OFF_0

Offset: 0x934
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000032 (0b0000,0000,0000,0000,0000,0000,0011,0010)

Bit	R/W	Reset	Description
31:16	RO	0x0	**RSVDP_16**: Reserved for future use.
15:0	RW	0x32	**INTERFACE_TIMER_TARGET**: Interface timer target value. This field should only be programmed when the INTERFACE_TIMER_EN bit is set to 1'b0. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_INTERFACE_TIMER_STATUS_OFF_0

where <*i*> = 1, 4, 8.

Description: This is the interface timer status register.

PCIE_X1_RC_PF0_PORT_LOGIC_INTERFACE_TIMER_STATUS_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_INTERFACE_TIMER_STATUS_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_INTERFACE_TIMER_STATUS_OFF_0

Offset: 0x938
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,000x,x000,0x00)

Bit	Reset	Description
11	0x0	**SLAVE_RD_ADD_TIMEOUT**: Slave read address channel timeout.
10	0x0	**SLAVE_WR_DATA_TIMEOUT**: Slave write data channel timeout.
9	0x0	**SLAVE_WR_ADD_TIMEOUT**: Slave write address channel timeout.
6	0x0	**MASTER_RD_DATA_TIMEOUT**: Master read data channel timeout.
5	0x0	**MASTER_WR_RES_TIMEOUT**: Master write response channel timeout.
4	0x0	**CLIENT2_INTERFACE_TIMEOUT**: Client2 interface timeout.
3	0x0	**CLIENT1_INTERFACE_TIMEOUT**: Client1 interface timeout.
1	0x0	**CPL_INTERFACE_TIMEOUT**: CPL interface timeout.
0	0x0	**MESSAGE_INTERFACE_TIMEOUT**: Message interface timeout.

PCIE_X<*j*>_RC_PF0_PORT_LOGIC_MSIX_ADDRESS_MATCH_LOW_OFF_0

where <j> = 4, 8.

Description: When you enable the MSI-X Table RAM feature (MSIX_TABLE_EN=1), the controller implements the logic and RAM required to generate MSI-X requests. For more details, see the Interrupts section in the "Controller Operations" chapter of the Databook. This register is only used in AXI configurations. When your local AXI application writes (MWr) to the address defined in this register (and MSIX_ADDRESS_MATCH_HIGH_OFF), the controller will load the MSIX_DOORBELL_OFF register with the contents of the MWr and subsequently create and send MSI-X TLPs.

PCIE_X4_RC_PF0_PORT_LOGIC_MSIX_ADDRESS_MATCH_LOW_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSIX_ADDRESS_MATCH_LOW_OFF_0

Offset: 0x940
Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:2	RW	0x0	**MSIX_ADDRESS_MATCH_LOW**: MSI-X Address Match Low Address. Note: This register field is sticky.
1	RO	0x0	**MSIX_ADDRESS_MATCH_RESERVED_1**: Reserved. Note: This register field is sticky.
0	RW	0x0	**MSIX_ADDRESS_MATCH_EN**: MSI-X Match Enable. Enable the MSI-X Address Match feature when the AXI bridge is present. Note: This register field is sticky.

PCIE_X<*j*>_RC_PF0_PORT_LOGIC_MSIX_ADDRESS_MATCH_HIGH_OFF_0

where <j> = 4, 8.

Description: MSI-X Address Match High Register. When you enable the MSI-X Table RAM feature (MSIX_TABLE_EN=1), the controller implements the logic and RAM required to generate MSI-X requests. For more details, see the Interrupts section in the "Controller Operations" chapter of the Databook. This register is only used in AXI configurations. When your local AXI application writes (MWr) to the address defined in this register (and MSIX_ADDRESS_MATCH_LOW_OFF), the controller will load the MSIX_DOORBELL_OFF register with the contents of the MWr and subsequently create and send MSI-X TLPs.

PCIE_X4_RC_PF0_PORT_LOGIC_MSIX_ADDRESS_MATCH_HIGH_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSIX_ADDRESS_MATCH_HIGH_OFF_0

Offset: 0x944
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSIX_ADDRESS_MATCH_HIGH**: MSI-X Address Match High Address. Note: This register field is sticky.

PCIE_X<*j*>_RC_PF0_PORT_LOGIC_MSIX_DOORBELL_OFF_0

where <j> = 4, 8.

Description: When you enable the MSI-X Table RAM feature (MSIX_TABLE_EN=1), the controller implements the logic and RAM required to generate MSI-X requests. For more details, see the Interrupts section in the "Controller Operations" chapter of the Databook. - For AXI configurations: when your local application writes (MWr) to the address defined in MSIX_ADDRESS_MATCH_LOW_OFF, the controller will load this register with the contents of the MWr and subsequently create and send MSI-X TLPs. - For non-AMBA configurations: when your local application writes to this register, the controller will create and send MSI-X TLPs.

PCIE_X4_RC_PF0_PORT_LOGIC_MSIX_DOORBELL_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSIX_DOORBELL_OFF_0

Offset: 0x948
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
Bit	**Reset**	**Description**
31:29	0x0	**MSIX_DOORBELL_RESERVED_29_31**: Reserved.
28:24	0x0	**MSIX_DOORBELL_PF**: MSIX Doorbell Physical Function. This register determines the Physical Function for the MSI-X transaction.
23:16	0x0	**MSIX_DOORBELL_VF**: MSIX Doorbell Virtual Function. This register determines the Virtual Function for the MSI-X transaction.
15	0x0	**MSIX_DOORBELL_VF_ACTIVE**: MSIX Doorbell Virtual Function Active. This register determines whether a Virtual Function is used to generate the MSI-X transaction.
14:12	0x0	**MSIX_DOORBELL_TC**: MSIX Doorbell Traffic Class. This register determines which traffic class to generate the MSI-X transaction with.
11	0x0	**MSIX_DOORBELL_RESERVED_11**: Reserved.
10:0	0x0	**MSIX_DOORBELL_VECTOR**: MSI-X Doorbell Vector. This register determines which vector to generate the MSI-X transaction for.

PCIE_X<*j*>_RC_PF0_PORT_LOGIC_MSIX_RAM_CTRL_OFF_0

where <j> = 4, 8.

Description: When you enable the MSI-X Table RAM feature (MSIX_TABLE_EN=1), the controller implements the logic and RAM required to generate MSI-X requests. For more details, see the Interrupts section in the "Controller Operations" chapter of the Databook.

PCIE_X4_RC_PF0_PORT_LOGIC_MSIX_RAM_CTRL_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_MSIX_RAM_CTRL_OFF_0

Offset: 0x94c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:26	RO	0x0	**MSIX_RAM_CTRL_RESERVED_26_31**: Reserved. Note: This register field is sticky.
25	RW	0x0	**MSIX_RAM_CTRL_DBG_PBA**: MSIX PBA RAM Debug Mode. You can also use the dbg_pba input to activate debug mode. Debug mode turns off the PF/VF/Offset-based addressing into the RAM and maps the entire table linearly from the base address of the BAR (indicated by the BIR) in function 0. Note: This register field is sticky.
24	RW	0x0	**MSIX_RAM_CTRL_DBG_TABLE**: MSIX Table RAM Debug Mode. You can also use the dbg_table input to activate debug mode. Debug mode turns off the PF/VF/Offset-based addressing into the RAM and maps the entire table linearly from the base address of the BAR (indicated by the BIR) in function 0. Note: This register field is sticky.
23:17	RO	0x0	**MSIX_RAM_CTRL_RESERVED_17_23**: Reserved. Note: This register field is sticky.
16	RW	0x0	**MSIX_RAM_CTRL_BYPASS**: MSIX RAM Control Bypass. It is up to the application to ensure the RAMs are in the proper power state before trying to access them. Moreover, the application needs to observe all timing requirements of the RAM low power signals before trying to use the MSIX functionality. Note: This register field is sticky.
15:10	RO	0x0	**MSIX_RAM_CTRL_RESERVED_10_15**: Reserved. Note: This register field is sticky.
9	RW	0x0	**MSIX_RAM_CTRL_PBA_SD**: MSIX PBA RAM Shut Down. Note: This register field is sticky.
8	RW	0x0	**MSIX_RAM_CTRL_PBA_DS**: MSIX PBA RAM Deep Sleep. Note: This register field is sticky.
7:2	RO	0x0	**MSIX_RAM_CTRL_RESERVED_2_7**: Reserved. Note: This register field is sticky.
1	RW	0x0	**MSIX_RAM_CTRL_TABLE_SD**: MSIX Table RAM Shut Down. Note: This register field is sticky.

Bit	R/W	Reset	Description
0	RW	0x0	**MSIX_RAM_CTRL_TABLE_DS**: MSIX Table RAM Deep Sleep. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_SAFETY_MASK_OFF_0

where <*i*> = 1, 4, 8.

Description: This registers holds the masks for functional safety interrupt events.

PCIE_X1_RC_PF0_PORT_LOGIC_SAFETY_MASK_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_SAFETY_MASK_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_SAFETY_MASK_OFF_0

Offset: 0x960
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:6	RO	0x0	**RSVDP_6**: Reserved for future use.
5	RW	0x0	**SAFETY_INT_MASK_5**: Mask for functional safety interrupt event 5 (RASDP correctable). Note: This register field is sticky.
4	RW	0x0	**SAFETY_INT_MASK_4**: Mask for functional safety interrupt event 4 (PCIe correctable). Note: This register field is sticky.
3	RW	0x0	**SAFETY_INT_MASK_3**: Mask for functional safety interrupt event 3 (PCIe uncorrectable). Note: This register field is sticky.
2	RW	0x0	**SAFETY_INT_MASK_2**: Mask for functional safety interrupt event 2 (Interface timers). Note: This register field is sticky.
1	RW	0x0	**SAFETY_INT_MASK_1**: Mask for functional safety interrupt event 1 (CDM register checker). Note: This register field is sticky.
0	RW	0x0	**SAFETY_INT_MASK_0**: Mask for functional safety interrupt event 0 (RASDP). Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_SAFETY_STATUS_OFF_0

where <*i*> = 1, 4, 8.

Description: This register provides the functional safety interrupt events status.

PCIE_X1_RC_PF0_PORT_LOGIC_SAFETY_STATUS_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_SAFETY_STATUS_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_SAFETY_STATUS_OFF_0

Offset: 0x964
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:6	RO	0x0	**RSVDP_6**: Reserved for future use.
5	RW	0x0	**SAFETY_INT_STATUS_5**: Status for functional safety interrupt event 5 (RASDP correctable). This register field is sticky.
4	RW	0x0	**SAFETY_INT_STATUS_4**: Status for functional safety interrupt event 4 (PCIe correctable). This register field is sticky.
3	RW	0x0	**SAFETY_INT_STATUS_3**: Status for functional safety interrupt event 3 (PCIe uncorrectable). This register field is sticky.
2	RW	0x0	**SAFETY_INT_STATUS_2**: Status for functional safety interrupt event 2 (Interface timers). This register field is sticky.
1	RW	0x0	**SAFETY_INT_STATUS_1**: Status for functional safety interrupt event 1 (CDM register checker). This register field is sticky.
0	RW	0x0	**SAFETY_INT_STATUS_0**: Status for functional safety interrupt event 0 (RASDP). This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_PL_APP_BUS_DEV_NUM_STATUS_OFF_0

where <*i*> = 1, 4, 8.

Description: This register reflects the application driven bus and device number.

PCIE_X1_RC_PF0_PORT_LOGIC_PL_APP_BUS_DEV_NUM_STATUS_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_PL_APP_BUS_DEV_NUM_STATUS_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_PL_APP_BUS_DEV_NUM_STATUS_OFF_0

Offset: 0xb10
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RSVDP_16**: Reserved for future use.
15:8	0x0	**RC_DSW_BUS_NUM**: This field reflects the value of bus number driven on app_bus_num input signal by your application. Note: This register field is sticky.
7:3	0x0	**RC_DSW_DEV_NUM**: This field reflects the value of device number driven on app_device_num input signal by your application. Note: This register field is sticky.
2:0	0x0	**RSVDP_0**: Reserved for future use.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_PCIPM_TRAFFIC_CTRL_OFF_0

where <*i*> = 1, 4, 8.

Description: This register provides control over TLP Traffic during Non-D0 States.

PCIE_X1_RC_PF0_PORT_LOGIC_PCIPM_TRAFFIC_CTRL_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_PCIPM_TRAFFIC_CTRL_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_PCIPM_TRAFFIC_CTRL_OFF_0

Offset: 0xb1c
Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:8	RO	0x0	**RSVDP_8**: Reserved for future use.
7:4	RO	0x0	**PCIPM_RESERVED_4_7**: Reserved. Note: This register field is sticky.
3	RW	0x0	**PCIPM_NEW_TLP_CLIENT2_BLOCKED**: This field indicates that all TLPs transmitted by Client 2 interface are blocked during non-D0 states. Note: This register field is sticky.
2	RW	0x0	**PCIPM_NEW_TLP_CLIENT1_BLOCKED**: This field indicates that all TLPs transmitted by Client 1 interface are blocked during non-D0 states. Note: This register field is sticky.
1	RW	0x0	**PCIPM_NEW_TLP_CLIENT0_BLOCKED**: This field indicates that all TLPs transmitted by Client 0 interface are blocked during non-D0 states. Note: This register field is sticky.
0	RW	0x0	**PCIPM_VDM_TRAFFIC_BLOCKED**: This field indicates that VDM Message TLPs are blocked during non-D0 states. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_PL_CHK_REG_CONTROL_STATUS_OFF_0

where <*i*> = 1, 4, 8.

Description: Controls register checking and displays status of register checking.

PCIE_X1_RC_PF0_PORT_LOGIC_PL_CHK_REG_CONTROL_STATUS_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_PL_CHK_REG_CONTROL_STATUS_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_PL_CHK_REG_CONTROL_STATUS_OFF_0

Offset: 0xb20
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:19	RO	0x0	**RSVDP_19**: Reserved for future use.

Bit	R/W	Reset	Description
18	RW	0x0	**CHK_REG_COMPLETE**: The system has completed a checking cycle.
17	RW	0x0	**CHK_REG_LOGIC_ERROR**: The system has detected an error in its own checking logic.
16	RW	0x0	**CHK_REG_COMPARISON_ERROR**: The system has detected that there is a bit error in the CDM Register Data.
15:2	RO	0x0	**RSVDP_2**: Reserved for future use.
1	RW	0x0	**CHK_REG_CONTINUOUS**: Set Continuous Checking Sequence. Note: This register field is sticky.
0	RW	0x0	**CHK_REG_START**: Begins a checking sequence. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_PL_CHK_REG_START_END_OFF_0

where <*i*> = 1, 4, 8.

Description: This register holds the first and last address to check.

PCIE_X1_RC_PF0_PORT_LOGIC_PL_CHK_REG_START_END_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_PL_CHK_REG_START_END_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_PL_CHK_REG_START_END_OFF_0

Offset: 0xb24
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0bff0000 (0b0000,1011,1111,1111,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0xbff	**CHK_REG_END_ADDR**: The last address that is checked by the system. Note: This register field is sticky.
15:0	0x0	**CHK_REG_START_ADDR**: The first address that is checked by the system. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_PL_CHK_REG_ERR_ADDR_OFF_0

where <*i*> = 1, 4, 8.

Description: This register holds the CDM register checking error address.

PCIE_X1_RC_PF0_PORT_LOGIC_PL_CHK_REG_ERR_ADDR_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_PL_CHK_REG_ERR_ADDR_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_PL_CHK_REG_ERR_ADDR_OFF_0

Offset: 0xb28
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CHK_REG_ERR_ADDR**: The address at which an error has been detected. Valid only when the CDM Register Checker Comparison Error bit is set in the status register. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_PL_CHK_REG_ERR_PF_VF_OFF_0

where <*i*> = 1, 4, 8.

Description: This register holds the CDM Register checking error PF and VF numbers.

PCIE_X1_RC_PF0_PORT_LOGIC_PL_CHK_REG_ERR_PF_VF_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_PL_CHK_REG_ERR_PF_VF_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_PL_CHK_REG_ERR_PF_VF_OFF_0

Offset: 0xb2c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**RSVDP_28**: Reserved for future use.
27:16	0x0	**CHK_REG_VF_ERR_NUMBER**: The VF number at which the error was detected. Valid only when the CDM Register Checker Comparison Error bit is set in the status register. Note: This register field is sticky.
15:5	0x0	**RSVDP_5**: Reserved for future use.
4:0	0x0	**CHK_REG_PF_ERR_NUMBER**: The PF number at which the error was detected. Valid only when the CDM Register Checker Comparison Error bit is set in the status register. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_PL_LTR_LATENCY_OFF_0

where <*i*> = 1, 4, 8.

Description: The function of this register field (and all other fields in this register) differs between an upstream port and a downstream port. For an upstream port, the register fields capture the corresponding fields in the LTR messages that are transmitted by the port. For a downstream port, the register fields capture the corresponding fields in the LTR messages that are received by the port. The full content of the register is reflected on the app_ltr_latency[31:0] output.

PCIE_X1_RC_PF0_PORT_LOGIC_PL_LTR_LATENCY_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_PL_LTR_LATENCY_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_PL_LTR_LATENCY_OFF_0

Offset: 0xb30
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**NO_SNOOP_LATENCY_REQUIRE**: No Snoop Latency Requirement. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
30:29	RO	0x0	**RSVDP_29**: Reserved for future use.

Bit	R/W	Reset	Description
28:26	RW	0x0	**NO_SNOOP_LATENCY_SCALE**: No Snoop Latency Scale. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
25:16	RW	0x0	**NO_SNOOP_LATENCY_VALUE**: No Snoop Latency Value. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
15	RW	0x0	**SNOOP_LATENCY_REQUIRE**: Snoop Latency Requirement. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
14:13	RO	0x0	**RSVDP_13**: Reserved for future use.
12:10	RW	0x0	**SNOOP_LATENCY_SCALE**: Snoop Latency Scale. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W
9:0	RW	0x0	**SNOOP_LATENCY_VALUE**: Snoop Latency Value. Note: The access attributes of this field are as follows: - Wire: No access. - Dbi: R/W

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_AUX_CLK_FREQ_OFF_0

where <*i*> = 1, 4, 8.

Description: This register controls the auxiliary clock frequency.

PCIE_X1_RC_PF0_PORT_LOGIC_AUX_CLK_FREQ_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_AUX_CLK_FREQ_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_AUX_CLK_FREQ_OFF_0

Offset: 0xb40
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000013 (0b0000,0000,0000,0000,0000,0000,0001,0011)

Bit	R/W	Reset	Description
31:10	RO	0x0	**RSVDP_10**: Reserved for future use.
9:0	RW	0x13	**AUX_CLK_FREQ**

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_L1_SUBSTATES_OFF_0

where <*i*> = 1, 4, 8.

Description: This register holds L1 substates timing information.

PCIE_X1_RC_PF0_PORT_LOGIC_L1_SUBSTATES_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_L1_SUBSTATES_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_L1_SUBSTATES_OFF_0

Offset: 0xb44
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000000d2 (0b0000,0000,0000,0000,0000,0000,1101,0010)

Bit	R/W	Reset	Description
31:9	RO	0x0	**RSVDP_9**: Reserved for future use.
8	RW	0x0	**L1SUB_LOW_POWER_CLOCK_SWITCH_MODE**: If the bit is set to 1'b1 the controller will delay the switching of aux_clk to the slow platform clock until it detects that the link partner has de-asserted CLKREQ#. Note: This register field is sticky.
7:6	RW	0x3	**L1SUB_T_PCLKACK**: Max delay (in 1us units) between a MAC request to remove the clock on mac_phy_pclkreq_n and a PHY response on phy_mac_pclkack_n. If the PHY does not respond within this time the request is aborted. Range is 1..4 Note: This register field is sticky.
5:2	RW	0x4	**L1SUB_T_L1_2**: Duration (in 1us units) of L1.2. Range is 1..16. Note: This register field is sticky.
1:0	RW	0x2	**L1SUB_T_POWER_OFF**: Duration (in 1us units) of L1.2.Entry. Range is 1..4. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_POWERDOWN_CTRL_STATUS_OFF_0

where <*i*> = 1, 4, 8.

Description: This is the Powerdown Control and Status register.

PCIE_X1_RC_PF0_PORT_LOGIC_POWERDOWN_CTRL_STATUS_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_POWERDOWN_CTRL_STATUS_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_POWERDOWN_CTRL_STATUS_OFF_0

Offset: 0xb48
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000220 (0b0000,0000,0000,0000,0000,0010,0010,0000)

Bit	R/W	Reset	Description
31:12	RO	0x0	**RSVDP_12**: Reserved for future use.
11:8	RO	0x2	**POWERDOWN_PHY_POWERDOWN**: This field represents the Powerdown value that has been acknowledged by the PHY. It is updated with the value of Powerdown driven by the controller, when the PHY has returned the Phystatus acknowledgment for the Powerdown transition.
7:4	RO	0x2	**POWERDOWN_MAC_POWERDOWN**: This field represents the Powerdown value driven by the controller to the PHY.
3:2	RO	0x0	**RSVDP_2**: Reserved for future use.
1	RW	0x0	**POWERDOWN_VMAIN_ACK**: Set this bit to 1 if you do not want to perform the handshake with the power-switch after PERST# assertion. By default the controller will perform the handshake with the power-switch if L1 power gating is enabled Note: This register field is sticky.
0	WO	0x0	**POWERDOWN_FORCE**: This field is a one shot field. This field could be used for debug purposes in event that the P2 Powerdown transition does not complete. It will allow the controller to proceed with the transition to the P1 Powerdown state. This field always reads back as 1'b0.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_GEN4_LANE_MARGINING_1_OFF_0

where <*i*> = 1, 4, 8.

Description: This is the Gen4 Lane Margining 1 Register.

PCIE_X1_RC_PF0_PORT_LOGIC_GEN4_LANE_MARGINING_1_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_GEN4_LANE_MARGINING_1_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_GEN4_LANE_MARGINING_1_OFF_0

Offset: 0xb80
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x05201407 (0b0000,0101,0010,0000,0001,0100,0000,0111)

Bit	R/W	Reset	Description
31:30	RO	0x0	**RSVDP_30**: Reserved for future use.
29:24	RW	0x5	**MARGINING_MAX_VOLTAGE_OFFSET**: M(MaxVoltageOffset) for Lane Margining at the Receiver. This parameter is defined in "Electrical Sub-Block" section of the PCI Express Base Specification. Note: This register field is sticky.
23	RO	0x0	**RSVDP_23**: Reserved for future use.
22:16	RW	0x20	**MARGINING_NUM_VOLTAGE_STEPS**: M(NumVoltageSteps) for Lane Margining at the Receiver. This parameter is defined in "Electrical Sub-Block" section of the PCI Express Base Specification. Note: This register field is sticky.
15:14	RO	0x0	**RSVDP_14**: Reserved for future use.
13:8	RW	0x14	**MARGINING_MAX_TIMING_OFFSET**: M(MaxTimingOffset) for Lane Margining at the Receiver. This parameter is defined in "Electrical Sub-Block" section of the PCI Express Base Specification. Note: This register field is sticky.
7:6	RO	0x0	**RSVDP_6**: Reserved for future use.
5:0	RW	0x7	**MARGINING_NUM_TIMING_STEPS**: M(NumTimingSteps) for Lane Margining at the Receiver. This parameter is defined in "Electrical Sub-Block" section of the PCI Express Base Specification. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_GEN4_LANE_MARGINING_2_OFF_0

where <*i*> = 1, 4, 8.

Description: This is the Gen4 Lane Margining 2 Register.

PCIE_X1_RC_PF0_PORT_LOGIC_GEN4_LANE_MARGINING_2_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_GEN4_LANE_MARGINING_2_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_GEN4_LANE_MARGINING_2_OFF_0

Offset: 0xb84
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x180f0f0f (0bx001,1000,0000,1111,0000,1111,0000,1111)

Bit	R/W	Reset	Description
30:29	RO	0x0	**RSVDP_29**: Reserved for future use.
28	RW	0x1	**MARGINING_IND_ERROR_SAMPLER**: M(IndErrorSampler) for Lane Margining at the Receiver. This parameter is defined in "Electrical Sub-Block" section of the PCI Express Base Specification. Note: This register field is sticky.
27	RW	0x1	**MARGINING_SAMPLE_REPORTING_METHOD**: M(SampleReportingMethod) for Lane Margining at the Receiver. This parameter is defined in "Electrical Sub-Block" section of the PCI Express Base Specification. Note: This register field is sticky.
26	RW	0x0	**MARGINING_IND_LEFT_RIGHT_TIMING**: M(IndLeftRightTiming) for Lane Margining at the Receiver. This parameter is defined in "Electrical Sub-Block" section of the PCI Express Base Specification. Note: This register field is sticky.
25	RW	0x0	**MARGINING_IND_UP_DOWN_VOLTAGE**: M(IndUpDownVoltage) for Lane Margining at the Receiver. This parameter is defined in "Electrical Sub-Block" section of the PCI Express Base Specification. Note: This register field is sticky.
24	RW	0x0	**MARGINING_VOLTAGE_SUPPORTED**: M(VoltageSupported) for Lane Margining at the Receiver. This parameter is defined in "Electrical Sub-Block" section of the PCI Express Base Specification. Note: This register field is sticky.
23:21	RO	0x0	**RSVDP_21**: Reserved for future use.
20:16	RW	0xf	**MARGINING_MAXLANES**: M(MaxLanes) for Lane Margining at the Receiver. This parameter is defined in "Electrical Sub-Block" section of the PCI Express Base Specification. Note: This register field is sticky.
15:14	RO	0x0	**RSVDP_14**: Reserved for future use.
13:8	RW	0xf	**MARGINING_SAMPLE_RATE_TIMING**: M(SamplingRateTiming) for Lane Margining at the Receiver. This parameter is defined in "Electrical Sub-Block" section of the PCI Express Base Specification. Note: This value is not used when MARGINING_IND_ERROR_SAMPLER is 0b. The M(SamplingRateTiming) is fixed to 63 internally. Note: This register field is sticky.
7:6	RO	0x0	**RSVDP_6**: Reserved for future use.

Bit	R/W	Reset	Description
5:0	RW	0xf	**MARGINING_SAMPLE_RATE_VOLTAGE**: M(SamplingRateVoltage) for Lane Margining at the Receiver. This parameter is defined in "Electrical Sub-Block" section of the PCI Express Base Specification. Note: This value is not used when MARGINING_IND_ERROR_SAMPLER is 0b. The M(SamplingRateVoltage) is fixed to 63 internally. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_PORT_LOGIC_PIPE_RELATED_OFF_0

where <*i*> = 1, 4, 8.

Description: This register controls the PIPE's capabitity, control, and status parameters.

PCIE_X1_RC_PF0_PORT_LOGIC_PIPE_RELATED_OFF_0

PCIE_X4_RC_PF0_PORT_LOGIC_PIPE_RELATED_OFF_0

PCIE_X8_RC_PF0_PORT_LOGIC_PIPE_RELATED_OFF_0

Offset: 0xb90
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,xxxx,xxxx)

Bit	R/W	Reset	Description
31:9	RO	0x0	**RSVDP_9**: Reserved for future use.
8	RW	0x0	**PIPE_GARBAGE_DATA_MODE**: PIPE Garbage Data Mode. - RxValid is deasserted - a valid RxStartBlock is received at 128b/130b encoding - a valid COM symbol is received at 8b/10b encoding Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_MSIX_CAP_DBI2_SHADOW_PCI_MSIX_CAP_ID_NEXT_CTRL_REG_0

where <*i*> = 1, 4, 8.

Description: This is the shadow register of the MSI-X Capability ID, Next Pointer, and Control Register.

Offset: 0x10b0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_MSIX_CAP_DBI2_SHADOW_PCI_MSIX_CAP_ID_NEXT_CTRL_REG_0

Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:27	0x0	**PCI_MSIX_RESERVED1**: reserved field 1 in the shadow register. Note: This register field is sticky.
26:16	0x0	**PCI_MSIX_TABLE_SIZE**: MSI-X Table Size in the shadow register. Note: The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
15:0	0x0	**PCI_MSIX_RESERVED0**: reserved field 0 in the shadow register. Note: This register field is sticky.

PCIE_X4_RC_PF0_MSIX_CAP_DBI2_SHADOW_PCI_MSIX_CAP_ID_NEXT_CTRL_REG_0

PCIE_X8_RC_PF0_MSIX_CAP_DBI2_SHADOW_PCI_MSIX_CAP_ID_NEXT_CTRL_REG_0

Reset: 0x00070000 (0b0000,0000,0000,0111,0000,0000,0000,0000)

Bit	Reset	Description
31:27	0x0	**PCI_MSIX_RESERVED1**: reserved field 1 in the shadow register. Note: This register field is sticky.
26:16	0x7	**PCI_MSIX_TABLE_SIZE**: MSI-X Table Size in the shadow register. Note: The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
15:0	0x0	**PCI_MSIX_RESERVED0**: reserved field 0 in the shadow register. Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_MSIX_CAP_DBI2_SHADOW_MSIX_TABLE_OFFSET_REG_0

where <*i*> = 1, 4, 8.

Description: This register provides Table BIR and MSI-x Table offset select.

Offset: 0x10b4
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_MSIX_CAP_DBI2_SHADOW_MSIX_TABLE_OFFSET_REG_0

Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:3	0x0	**PCI_MSIX_TABLE_OFFSET**: MSI-X Table Offset. Used as an offset from the address contained by one of the Function's Base Address Registers to point to the base of the MSI-X Table. The lower 3 Table BIR bits are masked off (set to zero) by software to form a 32-bit QWORD-aligned offset. Note: The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: if (DBI_RO_WR_EN == 1 && MSIX_CAP_ENABLE=1) then R/W else R Note: This register field is sticky.
2:0	0x0	**PCI_MSIX_BIR**: MSI-X Table BAR Indicator Register Field. This field indicates which one of a Function's Base Address Registers , located beginning at 10h in Configuration Space, or entry in the Enhanced Allocation capability with a matching BEI , is used to map the Function's MSI-X Table into Memory Space. All encodings other than the defined encodings are reserved. For a 64-bit Base Address Register , the Table BIR indicates the lower DWORD. For Functions with Type 1 Configuration Space headers, BIR values 2 through 5 are also Reserved. Note: The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: if (DBI_RO_WR_EN == 1 && MSIX_CAP_ENABLE=1) then R/W else R Note: This register field is sticky.

PCIE_X4_RC_PF0_MSIX_CAP_DBI2_SHADOW_MSIX_TABLE_OFFSET_REG_0

PCIE_X8_RC_PF0_MSIX_CAP_DBI2_SHADOW_MSIX_TABLE_OFFSET_REG_0

Reset: 0x00000002 (0b0000,0000,0000,0000,0000,0000,0000,0010)

Bit	Reset	Description
31:3	0x0	**PCI_MSIX_TABLE_OFFSET**: MSI-X Table Offset. Used as an offset from the address contained by one of the Function's Base Address Registers to point to the base of the MSI-X Table. The lower 3 Table BIR bits are masked off (set to zero) by software to form a 32-bit QWORD-aligned offset. Note: The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: if (DBI_RO_WR_EN == 1 && MSIX_CAP_ENABLE=1) then R/W else R Note: This register field is sticky.
2:0	0x2	**PCI_MSIX_BIR**: MSI-X Table BAR Indicator Register Field. This field indicates which one of a Function's Base Address Registers , located beginning at 10h in Configuration Space, or entry in the Enhanced Allocation capability with a matching BEI , is used to map the Function's MSI-X Table into Memory Space. All encodings other than the defined encodings are reserved. For a 64-bit Base Address Register , the Table BIR indicates the lower DWORD. For Functions with Type 1 Configuration Space headers, BIR values 2 through 5 are also Reserved. Note: The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: if (DBI_RO_WR_EN == 1 && MSIX_CAP_ENABLE=1) then R/W else R Note: This register field is sticky.

PCIE_X<*i*>_RC_PF0_MSIX_CAP_DBI2_SHADOW_MSIX_PBA_OFFSET_REG_0

where <*i*> = 1, 4, 8.

Description: This register provides PBA Offset and PBA BIR value.

Offset: 0x10b8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X1_RC_PF0_MSIX_CAP_DBI2_SHADOW_MSIX_PBA_OFFSET_REG_0

Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:3	0x0	**PCI_MSIX_PBA_OFFSET**: MSI-X PBA Offset. Used as an offset from the address contained by one of the Function's Base Address Registers to point to the base of the MSI-X PBA. The lower 3 PBA BIR bits are masked off (set to zero) by software to form a 32-bit QWORD-aligned offset. Note: The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: if (DBI_RO_WR_EN == 1 && MSIX_CAP_ENABLE=1) then R/W else R Note: This register field is sticky.
2:0	0x0	**PCI_MSIX_PBA_BIR**: MSI-X PBA BIR. This field indicates which one of a Function's Base Address Registers , located beginning at 10h in Configuration Space, or entry in the Enhanced Allocation capability with a matching BEI, is used to map the Function's MSI-X PBA into Memory Space. The PBA BIR value definitions are identical to those for the Table BIR . Note: The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: if (DBI_RO_WR_EN == 1 && MSIX_CAP_ENABLE=1) then R/W else R Note: This register field is sticky.

PCIE_X4_RC_PF0_MSIX_CAP_DBI2_SHADOW_MSIX_PBA_OFFSET_REG_0

PCIE_X8_RC_PF0_MSIX_CAP_DBI2_SHADOW_MSIX_PBA_OFFSET_REG_0

Reset: 0x00010002 (0b0000,0000,0000,0001,0000,0000,0000,0010)

Bit	Reset	Description
31:3	0x2000	**PCI_MSIX_PBA_OFFSET**: MSI-X PBA Offset. Used as an offset from the address contained by one of the Function's Base Address Registers to point to the base of the MSI-X PBA. The lower 3 PBA BIR bits are masked off (set to zero) by software to form a 32-bit QWORD-aligned offset. Note: The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: if (DBI_RO_WR_EN == 1 && MSIX_CAP_ENABLE=1) then R/W else R Note: This register field is sticky.

Bit	Reset	Description
2:0	0x2	**PCI_MSIX_PBA_BIR**: MSI-X PBA BIR. This field indicates which one of a Function's Base Address Registers , located beginning at 10h in Configuration Space, or entry in the Enhanced Allocation capability with a matching BEI, is used to map the Function's MSI-X PBA into Memory Space. The PBA BIR value definitions are identical to those for the Table BIR . Note: The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: if (DBI_RO_WR_EN == 1 && MSIX_CAP_ENABLE=1) then R/W else R Note: This register field is sticky.

9.3.5.4 PCIe x4/x8 Endpoint

For a description of these standard PCIe register fields, see the PCI Express Specification.

NOTE:

There are 6 instances for each of the "X4" registers as listed, one for each of the X4 PCIe module instances, namely C0, C4, C6, C8, C9, and C10.
And there are 2 instances for each of the "X8" registers as listed, one for each of the X8 PCIe module instances, namely C5 and C7.
For the base addresses of these different register instances, please refer to the System Address Map.

NOTE:

The binary “x” is turned to “0” for the Reset and PROD hex value. Hence, in some cases, a mismatch occurs between the displayed hexadecimal and binary value. A bit Reset value as “x” or “Undefined” actually means that the bit could be either “0” or “1” after Reset, whereas a bit PROD value as “x” or “Don’t-care” means that either “0” or “1” can be written to the bit by software Initialization.

PCIE_X<*j*>_EP_PF0_TYPE0_HDR_DEVICE_ID_VENDOR_ID_REG_0

where <*j*> = 4, 8.

Description: This register holds the device ID and vendor ID.

PCIE_X4_EP_PF0_TYPE0_HDR_DEVICE_ID_VENDOR_ID_REG_0

PCIE_X8_EP_PF0_TYPE0_HDR_DEVICE_ID_VENDOR_ID_REG_0

Offset: 0x0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x229c10de (0b0010,0010,1001,1100,0001,0000,1101,1110)

Bit	Reset	Description
31:16	0x229c	**PCI_TYPE0_DEVICE_ID**: Device ID. The Device ID register identifies the particular Function. This identifier is allocated by the vendor. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x10de	**PCI_TYPE0_VENDOR_ID**: Vendor ID. The Vendor ID register identifies the manufacturer of the Function. Valid vendor identifiers are allocated by the PCI-SIG to ensure uniqueness. It is not permitted to populate this register with a value of FFFFh, which is an invalid value for Vendor ID. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_TYPE0_HDR_STATUS_COMMAND_REG_0

where <*j*> = 4, 8.

Description: This register provides the status and controls the behavior of a function.

PCIE_X4_EP_PF0_TYPE0_HDR_STATUS_COMMAND_REG_0

PCIE_X8_EP_PF0_TYPE0_HDR_STATUS_COMMAND_REG_0

Offset: 0x4
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00100000 (0b0000,0000,0001,000x,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**DETECTED_PARITY_ERR**: Detected Parity Error.
30	RW	0x0	**SIGNALED_SYS_ERR**: Signaled System Error.
29	RW	0x0	**RCVD_MASTER_ABORT**: Received Master Abort.

Bit	R/W	Reset	Description
28	RW	0x0	**RCVD_TARGET_ABORT**: Received Target Abort.
27	RW	0x0	**SIGNALED_TARGET_ABORT**: Signaled Target Abort.
26:25	RO	0x0	**DEV_SEL_TIMING**: DEVSEL Timing. This field was originally described in the PCI Local Bus Specification. Its functionality does not apply to PCI Express. The controller hardwires this field to 00b.
24	RW	0x0	**MASTER_DPE**: Master Data Parity Error. This bit is set by a Function if the Parity Error Response bit in the Command register is 1b and either of the following two conditions occurs: - Function receives a Poisoned Completion - Function transmits a Poisoned Request If the Parity Error Response bit is 0b, this bit is never set.
23	RO	0x0	**FAST_B2B_CAP**: Fast Back to Back Transaction Capable. This bit was originally described in the PCI Local Bus Specification. Its functionality does not apply to PCI Express. The controller hardwires this bit to 0b.
22	RO	0x0	**RSVDP_22**: Reserved for future use.
21	RO	0x0	**FAST_66MHZ_CAP**: 66MHz Capable. This bit was originally described in the PCI Local Bus Specification. Its functionality does not apply to PCI Express. The controller hardwires this bit to 0b.
20	RO	0x1	**CAP_LIST**: Capabilities List. Indicates the presence of an Extended Capability list item. Since all PCI Express device Functions are required to implement the PCI Express Capability structure, the controller hardwires this bit to 1b.
19	RO	0x0	**INT_STATUS**: Emulation interrupt pending. When set, indicates that an INTx emulation interrupt is pending internally in the Function. Setting the Interrupt Disable bit has no effect on the state of this bit. For Functions that do not generate INTx interrupts, the controller hardwires this bit to 0b.
18:17	RO	0x0	**RSVDP_17**: Reserved for future use.
15:11	RO	0x0	**PCI_TYPE_RESERV**: Reserved.
10	RW	0x0	**PCI_TYPE0_INT_EN**: Interrupt Disable. Controls the ability of a Function to generate INTx emulation interrupts. - When set, Functions are prevented from asserting INTx interrupts. Note: - Any INTx emulation interrupts already asserted by the Function must be de-asserted when this bit is Set. INTx interrupts use virtual wires that must, if asserted, be de-asserted using the appropriate Deassert_INTx message(s) when this bit is set. - Only the INTx virtual wire interrupt(s) associated with the Function(s) for which this bit is set are affected. - For functions that generate INTx interrupts, this bit is required. For functions that do not generate INTx interrupts, this bit is optional.

Bit	R/W	Reset	Description
9	RO	0x0	**RSVDP_9**: Reserved for future use.
8	RW	0x0	**PCI_TYPE0_SERREN**: SERR# Enable. When set, this bit enables reporting upstream of Non-fatal and Fatal errors detected by the Function. Note: The errors are reported if enabled either through this bit or through the PCI Express specific bits in the Device Control register.
7	RO	0x0	**PCI_TYPE_IDSEL_STEPPING**: IDSEL Stepping/Wait Cycle Control. This bit was originally described in the PCI Local Bus Specification. Its functionality does not apply to PCI Express. The controller hardwires this bit to 0b.
6	RW	0x0	**PCI_TYPE0_PARITY_ERR_EN**: Parity Error Response. This bit controls the logging of poisoned TLPs in the Master Data Parity Error bit in the Status register.
5	RO	0x0	**PCI_TYPE_VGA_PALETTE_SNOOP**: VGA Palette Snoop. This bit was originally described in the PCI Local Bus Specification and the PCI-to-PCI Bridge architecture specification. Its functionality does not apply to PCI Express, the controller hardwires this bit to 0b.
4	RO	0x0	**PCI_TYPE_MWI_ENABLE**: Memory Write and Invalidate. This bit was originally described in the PCI Local Bus Specification and the PCI-to-PCI Bridge architecture specification. Its functionality does not apply to PCI Express, the controller hardwires this bit to 0b.
3	RO	0x0	**PCI_TYPE0_SPECIAL_CYCLE_OPERATION**: Special Cycle Enable. This bit was originally described in the PCI Local Bus Specification. Its functionality does not apply to PCI Express. The controller hardwires this bit to 0b.
2	RW	0x0	**PCI_TYPE0_BUS_MASTER_EN**: Bus Master Enable. Controls the ability of a Function to issue Memory and I/O Read/Write requests. - When this bit is set, the Function is allowed to issue Memory or I/O Requests. - When this bit is clear, the Function is not allowed to issue any Memory or I/O Requests. Requests other than Memory or I/O Requests are not controlled by this bit. Note: MSI/MSI-X interrupt Messages are in-band memory writes, setting the Bus Master Enable bit to 0b disables MSI/MSI-X interrupt Messages as well. Note: The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
1	RW	0x0	**PCI_TYPE0_MEM_SPACE_EN**: Memory Space Enable. Controls a Function's response to Memory Space accesses. - When this bit is set, the Function is enabled to decode the address and further process Memory Space accesses. - When this bit is clear, all received Memory Space accesses are caused to be handled as Unsupported Requests. For a Function does not support Memory Space accesses, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: ! has_mem_bar ? RO : RW - Dbi: !has_mem_bar ? RO : RW

Bit	R/W	Reset	Description
0	RW	0x0	**PCI_TYPE0_IO_EN**: IO Space Enable. Controls a Function's response to I/O Space accesses. - When this bit is set, the Function is enabled to decode the address and further process I/O Space accesses. - When this bit is clear, all received I/O accesses are caused to be handled as Unsupported Requests. For a Function that does not support I/O Space accesses, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: !has_io_bar ? RO : RW - Dbi: ! has_io_bar ? RO : RW

PCIE_X<*j*>_EP_PF0_TYPE0_HDR_CLASS_CODE_REVISION_ID_0

where <*j*> = 4, 8.

Description: This register specifies the class code and revision ID of a function.

PCIE_X4_EP_PF0_TYPE0_HDR_CLASS_CODE_REVISION_ID_0

PCIE_X8_EP_PF0_TYPE0_HDR_CLASS_CODE_REVISION_ID_0

Offset: 0x8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x060400a1 (0b0000,0110,0000,0100,0000,0000,1010,0001)

Bit	Reset	Description
31:24	0x6	**BASE_CLASS_CODE**: Base Class Code. A code that broadly classifies the type of operation the Function performs. Encodings for base class, are provided in the PCI Code and ID Assignment Specification. All unspecified encodings are Reserved. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
23:16	0x4	**SUBCLASS_CODE**: Sub-Class Code. Specifies a base class sub-class, which identifies more specifically the operation of the Function. Encodings for sub-class are provided in the PCI Code and ID Assignment Specification. All unspecified encodings are Reserved. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:8	0x0	**PROGRAM_INTERFACE**: Programming Interface. This field identifies a specific register-level programming interface (if any) so that device independent software can interact with the Function. Encodings for interface are provided in the PCI Code and ID Assignment Specification. All unspecified encodings are Reserved. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

Bit	Reset	Description
7:0	0xa1	**REVISION_ID**: Revision ID. The value in this register specifies a Function specific revision identifier. The value is chosen by the vendor. Zero is an acceptable value. The Revision ID should be viewed as a vendor defined extension to the Device ID. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_TYPE0_HDR_BIST_HEADER_TYPE_LATENCY_CACHE_LINE_SIZE_REG_0

where <*j*> = 4, 8.

Description:This register provides the status and controls BIST. It also holds information regarding the header layout, latency timer, and cache line size.

PCIE_X4_EP_PF0_TYPE0_HDR_BIST_HEADER_TYPE_LATENCY_CACHE_LINE_SIZE_REG_0

PCIE_X8_EP_PF0_TYPE0_HDR_BIST_HEADER_TYPE_LATENCY_CACHE_LINE_SIZE_REG_0

Offset: 0xc
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**BIST**: BIST. This register is used for control and status of BIST. For Functions that do not support BIST the controller hardwires the register to 00h. A Function whose BIST is invoked must not prevent normal operation of the PCI Express Link. Bit descriptions: - [31]: BIST Capable. When Set, this bit indicates that the Function supports BIST. When Clear, the Function does not support BIST. - [30]: Start BIST. If BIST Capable is Set, Set this bit to invoke BIST. The Function resets the bit when BIST is complete. Software is permitted to fail the device if this bit is not Clear (BIST is not complete) 2 seconds after it had been Set. Writing this bit to 0b has no effect. This bit must be hardwired to 0b if BIST Capable is Clear. - [29:28]: Reserved. - [27:24]: Completion Code. This field encodes the status of the most recent test. A value of 0000b means that the Function has passed its test. Non-zero values mean the Function failed. Function-specific failure codes can be encoded in the non-zero values. This field's value is only meaningful when BIST Capable is Set and Start BIST is Clear. This field must be hardwired to 0000b if BIST Capable is clear.
23	RW	0x0	**MULTI_FUNC**: Multi-Function Device. Except where stated otherwise, it is recommended that this bit be set if there are multiple Functions, and clear if there is only one Function. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

Bit	R/W	Reset	Description
22:16	RO	0x0	**HEADER_TYPE**: Header Layout. This field identifies the layout of the second part of the predefined header. The controller uses 000 0000b encoding.
15:8	RO	0x0	**LATENCY_MASTER_TIMER**: Latency Timer. The Latency Timer was originally described in the PCI Local Bus Specification and the PCI-to-PCI Bridge Architecture Specification. Its functionality does not apply to PCI Express. The controller hardwires this register to 00h.
7:0	RW	0x0	**CACHE_LINE_SIZE**: Cache Line Size. The Cache Line Size register is programmed by the system firmware or the operating system to system cache line size. However, legacy conventional PCI software may not always be able to program this register correctly especially in the case of Hot-Plug devices. This read-write register is implemented for legacy compatibility purposes but has no effect on any PCI Express device behavior.

PCIE_X<*j*>_EP_PF0_TYPE0_HDR_BAR0_REG_0

where <*j*> = 4, 8.

Description:System software must build a consistent address map before booting the machine to an operating system. This means it has to determine how much memory is in the system, and how much address space the Functions in the system require. After determining this information, system software can map the Functions into reasonable locations and proceed with system boot. In order to do this mapping in a device-independent manner, the base registers for this mapping are placed in the predefined header portion of Configuration Space. It is strongly recommended that power-up firmware/software also support the optional Enhanced Configuration Access Mechanism (ECAM).

PCIE_X4_EP_PF0_TYPE0_HDR_BAR0_REG_0

PCIE_X8_EP_PF0_TYPE0_HDR_BAR0_REG_0

Offset: 0x10
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000c (0b0000,0000,0000,0000,0000,0000,0000,1100)

Bit	Reset	Description
31:4	0x0	**BAR0_START**: BAR0 Base Address. - Memory Space: Base Address. - IO Space: bits[31:2] are used to map the function into IO space/Base. Address. Note: The access attributes of this field are as follows: - Wire: R(Sticky)/W(Sticky) if enabled else R(Sticky) - Dbi: R(Sticky)/W(Sticky) if enabled else R(Sticky) Note: This register field is sticky.

Bit	Reset	Description
3	0x1	**BAR0_PREFETCH**: BAR0 Prefetchable. - Memory Space: Set to one if data is prefetchable. A Function is permitted to mark a range as prefetchable. If there are no side effects on reads, the function returns all bytes on reads regardless of the byte enables, and host bridges can merge processor writes into this range without causing errors. Bit must me clear otherwise. - IO Space: Not applicable Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.
2:1	0x2	**BAR0_TYPE**: BAR0 Type. - Memory Space: Base Address registers that map into Memory Space can be 32 bits or 64 bits wide (to support mapping into a 64-bit address space). The encodings defined in Values: apply. - IO Space: Bit 1 is reserved and must return 0b on reads. Bits[31:2] are used to map the function into IO space. The encodings defined in Values: do not apply. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.
0	0x0	**BAR0_MEM_IO**: BAR0 Memory Space Indicator. This bit is used to determine whether the register maps into Memory or I/O Space. Base Address registers that map to Memory Space must return a 0b. Base Address registers that map to I/O Space must return a 1b. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_TYPE0_HDR_BAR1_REG_0

where <*j*> = 4, 8.

Description:System software must build a consistent address map before booting the machine to an operating system. This means it has to determine how much memory is in the system, and how much address space the Functions in the system require. After determining this information, system software can map the Functions into reasonable locations and proceed with system boot. In order to do this mapping in a device-independent manner, the base registers for this mapping are placed in the predefined header portion of Configuration Space. It is strongly recommended that power-up firmware/software also support the optional Enhanced Configuration Access Mechanism (ECAM).

PCIE_X4_EP_PF0_TYPE0_HDR_BAR1_REG_0

PCIE_X8_EP_PF0_TYPE0_HDR_BAR1_REG_0

Offset: 0x14
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:4	0x0	**BAR1_START**: BAR1 Base Address. - Memory Space: Base Address. - IO Space: bits[31:2] are used to map the function into IO space/Base. Address. Note: The access attributes of this field are as follows: - Wire: R(Sticky)/W(Sticky) if enabled else R(Sticky) - Dbi: R(Sticky)/W(Sticky) if enabled else R(Sticky) Note: This register field is sticky.
3	0x0	**BAR1_PREFETCH**: BAR1 Prefetchable. - Memory Space: Set to one if data is prefetchable. A Function is permitted to mark a range as prefetchable. If there are no side effects on reads, the function returns all bytes on reads regardless of the byte enables, and host bridges can merge processor writes into this range without causing errors. Bit must me clear otherwise. - IO Space: Not applicable Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.
2:1	0x0	**BAR1_TYPE**: BAR1 Type. - Memory Space: Base Address registers that map into Memory Space can be 32 bits or 64 bits wide (to support mapping into a 64-bit address space). The encodings defined in Values: apply. - IO Space: Bit 1 is reserved and must return 0b on reads. Bits[31:2] are used to map the function into IO space. The encodings defined in Values: do not apply. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.
0	0x0	**BAR1_MEM_IO**: BAR1 Memory Space Indicator. This bit is used to determine whether the register maps into Memory or I/O Space. Base Address registers that map to Memory Space must return a 0b. Base Address registers that map to I/O Space must return a 1b. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_TYPE0_HDR_BAR2_REG_0

where <*j*> = 4, 8.

Description:System software must build a consistent address map before booting the machine to an operating system. This means it has to determine how much memory is in the system, and how much address space the Functions in the system require. After determining this information, system software can map the Functions into reasonable locations and proceed with system boot. In order to do this mapping in a device-independent manner, the base registers for this mapping are placed in the predefined header portion of Configuration Space. It is strongly recommended that power-up firmware/software also support the optional Enhanced Configuration Access Mechanism (ECAM).

PCIE_X4_EP_PF0_TYPE0_HDR_BAR2_REG_0

PCIE_X8_EP_PF0_TYPE0_HDR_BAR2_REG_0

Offset: 0x18
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000c (0b0000,0000,0000,0000,0000,0000,0000,1100)

Bit	Reset	Description
31:4	0x0	**BAR2_START**: BAR2 Base Address. - Memory Space: Base Address. - IO Space: bits[31:2] are used to map the function into IO space/Base. Address. Note: The access attributes of this field are as follows: - Wire: R(Sticky)/W(Sticky) if enabled else R(Sticky) - Dbi: R(Sticky)/W(Sticky) if enabled else R(Sticky) Note: This register field is sticky.
3	0x1	**BAR2_PREFETCH**: BAR2 Prefetchable. - Memory Space: Set to one if data is prefetchable. A Function is permitted to mark a range as prefetchable. If there are no side effects on reads, the function returns all bytes on reads regardless of the byte enables, and host bridges can merge processor writes into this range without causing errors. Bit must me clear otherwise. - IO Space: Not applicable Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.
2:1	0x2	**BAR2_TYPE**: BAR2 Type. - Memory Space: Base Address registers that map into Memory Space can be 32 bits or 64 bits wide (to support mapping into a 64-bit address space). The encodings defined in Values: apply. - IO Space: Bit 1 is reserved and must return 0b on reads. Bits[31:2] are used to map the function into IO space. The encodings defined in Values: do not apply. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.
0	0x0	**BAR2_MEM_IO**: BAR2 Memory Space Indicator. This bit is used to determine whether the register maps into Memory or I/O Space. Base Address registers that map to Memory Space must return a 0b. Base Address registers that map to I/O Space must return a 1b. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_TYPE0_HDR_BAR3_REG_0

where <*j*> = 4, 8.

Description:System software must build a consistent address map before booting the machine to an operating system. This means it has to determine how much memory is in the system, and how much address space the Functions in the system require. After determining this information, system software can map the Functions into reasonable locations and proceed with system boot. In order to do this mapping in a device-independent manner, the base registers for this mapping

are placed in the predefined header portion of Configuration Space. It is strongly recommended that power-up firmware/software also support the optional Enhanced Configuration Access Mechanism (ECAM).

PCIE_X4_EP_PF0_TYPE0_HDR_BAR3_REG_0

PCIE_X8_EP_PF0_TYPE0_HDR_BAR3_REG_0

Offset: 0x1c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:4	0x0	**BAR3_START**: BAR3 Base Address. - Memory Space: Base Address. - IO Space: bits[31:2] are used to map the function into IO space/Base. Address. Note: The access attributes of this field are as follows: - Wire: R(Sticky)/W(Sticky) if enabled else R(Sticky) - Dbi: R(Sticky)/W(Sticky) if enabled else R(Sticky) Note: This register field is sticky.
3	0x0	**BAR3_PREFETCH**: BAR3 Prefetchable. - Memory Space: Set to one if data is prefetchable. A Function is permitted to mark a range as prefetchable. If there are no side effects on reads, the function returns all bytes on reads regardless of the byte enables, and host bridges can merge processor writes into this range without causing errors. Bit must me clear otherwise. - IO Space: Not applicable Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.
2:1	0x0	**BAR3_TYPE**: BAR3 Type. - Memory Space: Base Address registers that map into Memory Space can be 32 bits or 64 bits wide (to support mapping into a 64-bit address space). The encodings defined in Values: apply. - IO Space: Bit 1 is reserved and must return 0b on reads. Bits[31:2] are used to map the function into IO space. The encodings defined in Values: do not apply. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.
0	0x0	**BAR3_MEM_IO**: BAR3 Memory Space Indicator. This bit is used to determine whether the register maps into Memory or I/O Space. Base Address registers that map to Memory Space must return a 0b. Base Address registers that map to I/O Space must return a 1b. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_TYPE0_HDR_BAR4_REG_0

where <*j*> = 4, 8.

Description:System software must build a consistent address map before booting the machine to an operating system. This means it has to determine how much memory is in the system, and how much address space the Functions in the system require. After determining this information, system software can map the Functions into reasonable locations and proceed with system boot. In order to do this mapping in a device-independent manner, the base registers for this mapping are placed in the predefined header portion of Configuration Space. It is strongly recommended that power-up firmware/software also support the optional Enhanced Configuration Access Mechanism (ECAM).

PCIE_X4_EP_PF0_TYPE0_HDR_BAR4_REG_0

PCIE_X8_EP_PF0_TYPE0_HDR_BAR4_REG_0

Offset: 0x20
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000004 (0b0000,0000,0000,0000,0000,0000,0000,0100)

Bit	Reset	Description
31:4	0x0	**BAR4_START**: BAR4 Base Address. - Memory Space: Base Address. - IO Space: bits[31:2] are used to map the function into IO space/Base. Address. Note: The access attributes of this field are as follows: - Wire: R(Sticky)/W(Sticky) if enabled else R(Sticky) - Dbi: R(Sticky)/W(Sticky) if enabled else R(Sticky) Note: This register field is sticky.
3	0x0	**BAR4_PREFETCH**: BAR4 Prefetchable. - Memory Space: Set to one if data is prefetchable. A Function is permitted to mark a range as prefetchable. If there are no side effects on reads, the function returns all bytes on reads regardless of the byte enables, and host bridges can merge processor writes into this range without causing errors. Bit must me clear otherwise. - IO Space: Not applicable Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.
2:1	0x2	**BAR4_TYPE**: BAR4 Type. - Memory Space: Base Address registers that map into Memory Space can be 32 bits or 64 bits wide (to support mapping into a 64-bit address space). The encodings defined in Values: apply. - IO Space: Bit 1 is reserved and must return 0b on reads. Bits[31:2] are used to map the function into IO space. The encodings defined in Values: do not apply. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.
0	0x0	**BAR4_MEM_IO**: BAR4 Memory Space Indicator. This bit is used to determine whether the register maps into Memory or I/O Space. Base Address registers that map to Memory Space must return a 0b. Base Address registers that map to I/O Space must return a 1b. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_TYPE0_HDR_BAR5_REG_0

where <*j*> = 4, 8.

Description:System software must build a consistent address map before booting the machine to an operating system. This means it has to determine how much memory is in the system, and how much address space the Functions in the system require. After determining this information, system software can map the Functions into reasonable locations and proceed with system boot. In order to do this mapping in a device-independent manner, the base registers for this mapping are placed in the predefined header portion of Configuration Space. It is strongly recommended that power-up firmware/software also support the optional Enhanced Configuration Access Mechanism (ECAM).

PCIE_X4_EP_PF0_TYPE0_HDR_BAR5_REG_0

PCIE_X8_EP_PF0_TYPE0_HDR_BAR5_REG_0

Offset: 0x24
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:4	0x0	**BAR5_START**: BAR5 Base Address. - Memory Space: Base Address. - IO Space: bits[31:2] are used to map the function into IO space/Base. Address. Note: The access attributes of this field are as follows: - Wire: R(Sticky)/W(Sticky) if enabled else R(Sticky) - Dbi: R(Sticky)/W(Sticky) if enabled else R(Sticky) Note: This register field is sticky.
3	0x0	**BAR5_PREFETCH**: BAR5 Prefetchable. - Memory Space: Set to one if data is prefetchable. A Function is permitted to mark a range as prefetchable. If there are no side effects on reads, the function returns all bytes on reads regardless of the byte enables, and host bridges can merge processor writes into this range without causing errors. Bit must me clear otherwise. - IO Space: Not applicable Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.
2:1	0x0	**BAR5_TYPE**: BAR5 Type. - Memory Space: Base Address registers that map into Memory Space can be 32 bits or 64 bits wide (to support mapping into a 64-bit address space). The encodings defined in Values: apply. - IO Space: Bit 1 is reserved and must return 0b on reads. Bits[31:2] are used to map the function into IO space. The encodings defined in Values: do not apply. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.

Bit	Reset	Description
0	0x0	**BAR5_MEM_IO**: BAR5 Memory Space Indicator. This bit is used to determine whether the register maps into Memory or I/O Space. Base Address registers that map to Memory Space must return a 0b. Base Address registers that map to I/O Space must return a 1b. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (BAR_ENABLED == 1) then (if [DBI_RO_WR_EN == 1] then R(Sticky)/W(Sticky) else R(Sticky)) else RO(Sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_TYPE0_HDR_CARDBUS_CIS_PTR_REG_0

where <*j*> = 4, 8.

Description:This register holds the CardBus CIS pointer.

PCIE_X4_EP_PF0_TYPE0_HDR_CARDBUS_CIS_PTR_REG_0

PCIE_X8_EP_PF0_TYPE0_HDR_CARDBUS_CIS_PTR_REG_0

Offset: 0x28
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CARDBUS_CIS_POINTER**: CardBus CIS Pointer. Its functionality does not apply to PCI Express. It must be hardwired to 0000 0000h. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_TYPE0_HDR_SUBSYSTEM_ID_SUBSYSTEM_VENDOR_ID_REG_0

where <*j*> = 4, 8.

Description:These registers are used to uniquely identify the add-in card or subsystem where the PCI Express component resides. They provide a mechanism for vendors to distinguish their products from one another even though the assemblies may have the same PCI Express component on them (and, therefore, the same Vendor ID and Device ID).

PCIE_X4_EP_PF0_TYPE0_HDR_SUBSYSTEM_ID_SUBSYSTEM_VENDOR_ID_REG_0

PCIE_X8_EP_PF0_TYPE0_HDR_SUBSYSTEM_ID_SUBSYSTEM_VENDOR_ID_REG_0

Offset: 0x2c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**SUBSYS_DEV_ID**: Subsystem ID. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x0	**SUBSYS_VENDOR_ID**: Subsystem Vendor ID. Subsystem Vendor IDs can be obtained from the PCI SIG and are used to identify the vendor of the add-in card or subsystem. Values for the Subsystem ID are vendor-specific. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_TYPE0_HDR_EXP_ROM_BASE_ADDR_REG_0

where <*j*> = 4, 8.

Description:This register handles the base address and size information for this expansion ROM.

PCIE_X4_EP_PF0_TYPE0_HDR_EXP_ROM_BASE_ADDR_REG_0

PCIE_X8_EP_PF0_TYPE0_HDR_EXP_ROM_BASE_ADDR_REG_0

Offset: 0x30
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:11	RW	0x0	**EXP_ROM_BASE_ADDRESS**: Expansion ROM Base Address. Upper 21 bits of the Expansion ROM base address. The number of bits (out of these 21) that a Function actually implements depends on how much address space the Function requires. Note: The access attributes of this field are as follows: - Wire: if (EXP_ROM_BAR_MASK_REG.ROM_BAR_ENABLED == 1) then R/W else R - Dbi: if (EXP_ROM_BAR_MASK_REG.ROM_BAR_ENABLED == 1) then R/W else R

Bit	R/W	Reset	Description
10:8	RO	0x0	**RSVDP_8**: Reserved for future use.
7:4	RW	0x0	**ROM_BAR_VALIDATION_DETAILS**: Expansion ROM Validation Details. The field contains optional, implementation-specific details associated with Expansion ROM Validation. - If validation is in progress (Expansion ROM Validation Status is 001b), non-zero values of this field represent implementation-specific indications of the phase of the validation progress (for example, 50% complete). The value 0000b indicates that no validation progress information is provided. - If validation is completed (Expansion ROM Validation Status 010b to 111b inclusive), non-zero values in this field represent additional implementation-specific information. The value 0000b indicates that no information is provided. - When validation is supported and this field is not implemented, this field must be hardwired to 0000b. Note: The access attributes of this field are as follows: - Wire: if (EXP_ROM_BAR_MASK_REG.ROM_BAR_ENABLED == 1) then R/W else R - Dbi: if (EXP_ROM_BAR_MASK_REG.ROM_BAR_ENABLED == 1 && DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
3:1	RW	0x0	**ROM_BAR_VALIDATION_STATUS**: Expansion ROM Validation Status. When this field is non-zero, it indicates the status of hardware validation of the Expansion ROM contents. - If the Function does not support validation, this field must be hardwired to 000b. - It is optional whether an implementation is capable of returning Validation Status values 011b, 101b, 110b, or 111b. Note: The access attributes of this field are as follows: - Wire: if (EXP_ROM_BAR_MASK_REG.ROM_BAR_ENABLED == 1) then R/W else R - Dbi: if (EXP_ROM_BAR_MASK_REG.ROM_BAR_ENABLED == 1 && DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
0	RW	0x0	**ROM_BAR_ENABLE**: Expansion ROM Enable. This bit controls whether or not the Function accepts accesses to its expansion ROM. The Memory Space Enable bit in the Command register has precedence over the Expansion ROM Enable bit. A Function must claim accesses to its expansion ROM only if both the Memory Space Enable bit and the Expansion ROM Enable bit are set. Note: The access attributes of this field are as follows: - Wire: if (EXP_ROM_BAR_MASK_REG.ROM_BAR_ENABLED == 1) then R/W else R - Dbi: if (EXP_ROM_BAR_MASK_REG.ROM_BAR_ENABLED == 1) then R/W else R

PCIE_X<*j*>_EP_PF0_TYPE0_HDR_PCI_CAP_PTR_REG_0

where <*j*> = 4, 8.

Description:This register is used to point to a linked list of capabilities implemented by a Function.

PCIE_X4_EP_PF0_TYPE0_HDR_PCI_CAP_PTR_REG_0

PCIE_X8_EP_PF0_TYPE0_HDR_PCI_CAP_PTR_REG_0

Offset: 0x34
Read/Write: See table below
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000040 (0b0000,0000,0000,0000,0000,0000,0100,0000)

Bit	R/W	Reset	Description
31:8	RO	0x0	**RSVDP_8**: Reserved for future use.
7:0	RW	0x40	**CAP_POINTER**: Capabilities Pointer. This register points to a valid capability structure. Either this structure is the PCI Express Capability structure, or a subsequent list item points to the PCI Express Capability structure. The bottom two bits are reserved, the controller sets it to 00b. Software must mask these bits off before using this register as a pointer in Configuration Space to the first entry of a linked list of new capabilities. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_TYPE0_HDR_MAX_LATENCY_MIN_GRANT_INTERRUPT_PIN_INTERRUPT_LINE_REG_0

where <*j*> = 4, 8.

Description:The Interrupt Line register communicates interrupt line routing information. The Interrupt Pin register identifies the legacy interrupt Message(s) the Function uses.

PCIE_X4_EP_PF0_TYPE0_HDR_MAX_LATENCY_MIN_GRANT_INTERRUPT_PIN_INTERRUPT_LINE_REG_0

PCIE_X8_EP_PF0_TYPE0_HDR_MAX_LATENCY_MIN_GRANT_INTERRUPT_PIN_INTERRUPT_LINE_REG_0

Offset: 0x3c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000001ff (0b0000,0000,0000,0000,0000,0001,1111,1111)

Bit	R/W	Reset	Description
31:16	RO	0x0	**RSVDP_16**: Reserved for future use.

Bit	R/W	Reset	Description
15:8	RW	0x1	**INT_PIN**: Interrupt Pin. The Interrupt Pin register identifies the legacy interrupt Message(s) the Function uses. All encodings other than the defined encodings are reserved. PCI Express defines one legacy interrupt Message for a single Function device and up to four legacy interrupt Messages for a multi-Function device. For a single Function device, only INTA may be used. Any Function on a multi-Function device can use any of the INTx Messages. If a device implements a single legacy interrupt Message, it must be INTA; if it implements two legacy interrupt Messages, they must be INTA and INTB; and so forth. For a multi-Function device, all Functions may use the same INTx Message or each may have its own (up to a maximum of four Functions) or any combination thereof. A single Function can never generate an interrupt request on more than one INTx Message. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
7:0	RW	0xff	**INT_LINE**: Interrupt Line. The Interrupt Line register communicates interrupt line routing information. The register must be implemented by any Function that uses an interrupt pin. Values in this register are programmed by system software and are system architecture specific. The Function itself does not use this value; rather the value in this register is used by device drivers and operating systems.

PCIE_X<*j*>_EP_PF0_PM_CAP_CAP_ID_NXT_PTR_REG_0

where <*j*> = 4, 8.

Description:This register provides information refarding the Power Management Capabilities.

PCIE_X4_EP_PF0_PM_CAP_CAP_ID_NXT_PTR_REG_0

PCIE_X8_EP_PF0_PM_CAP_CAP_ID_NXT_PTR_REG_0

Offset: 0x40
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x48035001 (0b0100,1000,000x,0011,0101,0000,0000,0001)

Bit	R/W	Reset	Description
31:27	RW	0x9	**PME_SUPPORT**: PME_Support. This 5-bit field indicates the power states in which the function may generate a PME and/or forward PME messages. A value of 0b for any bit indicates that the function is not capable of asserting PME while in that power state. - bit(27) X XXX1b - PME can be generated from D0 - bit(28) X XX1Xb - PME can be generated from D1 - bit(29) X X1XXb - PME can be generated from D2 - bit(30) X 1XXXb - PME can be generated from D3hot - bit(31) 1 XXXXb - PME can be generated from D3cold Bit 31 (PME can be asserted from D3cold) represents a special case. Functions that set this bit require some sort of auxiliary power source. Implementation specific mechanisms are recommended to validate that the power source is available before setting this bit. Each bit that corresponds to a supported D-state must be set for PCI-PCI Bridge structures representing Ports on Root Complexes/Switches to indicate that the Bridge will forward PME Messages. Bit 31 must only be set if the Port is still able to forward PME Messages when main power is not available. The read value from this field is the write value && (sys_aux_pwr_det, 1'b1, D2_SUPPORT, D1_SUPPORT, 1'b1), where D1_SUPPORT and D2_SUPPORT are fields in this register. The reset value PME_SUPPORT_n && (sys_aux_pwr_det, 1'b1, D2_SUPPORT, D1_SUPPORT, 1'b1), where PME_SUPPORT_n is a configuration parameter. Note: The access attributes of this field are as follows: - Wire: R - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
26	RW	0x0	**D2_SUPPORT**: D2_Support. If this bit is set, this function supports the D2 Power Management state. Functions that do not support D2 must always return a value of 0b for this bit. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
25	RW	0x0	**D1_SUPPORT**: D1_Support. If this bit is set, this function supports the D1 Power Management state. Functions that do not support D1 must always return a value of 0b for this bit. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
24:22	RW	0x0	**AUX_CURR**: Aux_Current. This 3 bit field reports the Vaux auxiliary current requirements for the function. If this function implements the Data Register, the controller hardwires this field to 000b. If PME_Support is 0 xxxxb (PME assertion from D3cold is not supported), the controller hardwires this field to 0000b. For functions where PME_Support is 1 xxxxb (PME assertion from D3cold is supported), and which do not implement the Data field, the encodings defined in Values: apply: Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
21	RW	0x0	**DSI**: Device Specific Initialization. The DSI bit indicates whether special initialization of this function is required. When set, indicates that the function requires a device specific initialization sequence following a transition to the D0uninitialized state. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
19	RO	0x0	**PME_CLK**: PME Clock. Does not apply to PCI Express, the controller hardwires it to 0b. Note: This register field is sticky.

Bit	R/W	Reset	Description
18:16	RW	0x3	**PM_SPEC_VER**: Version. This field provides the Power Management specification version. The controller hardwires this field to 011b for functions compliant to PCI Express Base Specification, Revision 4.0, Version 1.0>. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
15:8	RW	0x50	**PM_NEXT_POINTER**: Next Capability Pointer. This field provides an offset into the function's configuration space pointing to the location of next item in the capabilities list. If there are no additional items in the capabilities list, this field is set to 00h. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
7:0	RO	0x1	**PM_CAP_ID**: Capability ID. This field returns 01h to indicate that this is the PCI Power Management Capability. Each function may have only one item in its capability list with Capability ID set to 01h.

PCIE_X<*j*>_EP_PF0_PM_CAP_CON_STATUS_REG_0

where <*j*> = 4, 8.

Description: This register is used to manage the PCI function's power management state as well as to enable/monitor PMEs.

PCIE_X4_EP_PF0_PM_CAP_CON_STATUS_REG_0

PCIE_X8_EP_PF0_PM_CAP_CON_STATUS_REG_0

Offset: 0x44
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000008 (0b0000,0000,0000,0000,0000,0000,0000,1000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**DATA_REG_ADD_INFO**: Data. This field is used to report the state dependent data requested by the Data_Select field. The value of this field is scaled by the value reported by the Data_Scale field.
23	RO	0x0	**BUS_PWR_CLK_CON_EN**: Bus Power/Clock Control Enable. If this field is set, Bus Power/Clock Control is Enable.
22	RO	0x0	**B2_B3_SUPPORT**: B2B3 Support for D3hot. If this field is set, B2B3 support for D3hot is available.

Bit	R/W	Reset	Description
21:16	RO	0x0	**RSVDP_16**: Reserved for future use.
15	RW	0x0	**PME_STATUS**: PME_Status. This bit is set when the function normally generates a PME signal. The value of this bit is not affected by the value of the PME_En bit. If PME_Support bit 31 of the Power Management Capabilities register is clear, this bit is permitted to be hardwired to 0b. Functions that consume Aux power must preserve the value of this sticky register when Aux power is available. In such functions, this register value is not modified by Conventional Reset or FLR.
14:13	RO	0x0	**DATA_SCALE**: Data_Scale. This field indicates the scaling factor to be used when interpreting the value of the Data field. The value and meaning of this field varies depending on which data value has been selected by the Data_Select field. For more details, see 7.5.2.3 section of PCI Express Base Specification.
12:9	RO	0x0	**DATA_SELECT**: Data_Select. This 4-bit field is used to select which data is to be reported through the Data and Data_Scale field. If the Data field is not implemented, this field must be hardwired to 0000b.
8	RW	0x0	**PME_ENABLE**: PME_En. - When set, the function is permitted to generate a PME. - When clear, the function is not permitted to generate a PME. If PME_Support is 1 xxxxb (PME generation from D3cold) or the function consumes Aux power and Aux power is available this bit is RWS and the bit is not modified by Conventional Reset or FLR. If PME_Support is 0 xxxxb, this field is not sticky (RW). If PME_Support is 0 0000b, the controller hardwires this bit to 0b. Note: This register field is sticky.
7:4	RO	0x0	**RSVDP_4**: Reserved for future use.
3	RW	0x1	**NO_SOFT_RST**: No_Soft_Reset. This bit indicates the state of the function after writing the PowerState field to transition the function from D3hot to D0. - When set, this transition preserves internal function state. The function is in D0Active and no additional software intervention is required. - When clear, this transition results in undefined internal function state. Regardless of this bit, functions that transition from D3hot to D0 by Fundamental Reset will return to D0Uninitialized with only PME context preserved if PME is supported and enabled. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
2	RO	0x0	**RSVDP_2**: Reserved for future use.
1:0	RW	0x0	**POWER_STATE**: PowerState. This 2-bit field is used both to determine the current power state of a function and to set the function into a new power state. You can write to this register; however, the read-back value is the actual power state, not the write value. If you attempt to write an unsupported, optional state to this field, the write operation completes normally; however, the data is discarded and no state change occurs. Note: The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

PCIE_X<*j*>_EP_PF0_MSI_CAP_PCI_MSI_CAP_ID_NEXT_CTRL_REG_0

where <*j*> = 4, 8.

Description:This register holds MSI Capability Header information and controls the MSI behaviour.

PCIE_X4_EP_PF0_MSI_CAP_PCI_MSI_CAP_ID_NEXT_CTRL_REG_0

PCIE_X8_EP_PF0_MSI_CAP_PCI_MSI_CAP_ID_NEXT_CTRL_REG_0

Offset: 0x50
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x01807005 (0b0000,0001,1000,0000,0111,0000,0000,0101)

Bit	R/W	Reset	Description
31:27	RO	0x0	**RSVDP_27**: Reserved for future use.
26	RW	0x0	**PCI_MSI_EXT_DATA_EN**: Extended Message Data Enable. - If set, the function is enabled to provide Extended Message Data. - If clear, the function is not enabled to provide Extended Message Data. Note: The access attributes of this field are as follows: - Wire: PCI_MSI_CAP_ID_NEXT_CTRL_REG.PCI_MSI_EXT_DATA_CAP ? RW : RO - Dbi: PCI_MSI_CAP_ID_NEXT_CTRL_REG.PCI_MSI_EXT_DATA_CAP ? RW : RO
25	RW	0x0	**PCI_MSI_EXT_DATA_CAP**: Extended Message Data Capable. - If set, the function is capable of providing Extended Message Data. - If clear, the function does not support providing Extended Message Data. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
24	RO	0x1	**PCI_PVM_SUPPORT**: Per-Vector Masking Capable. - If set, the function supports MSI Per-Vector Masking. - If clear, the function does not support MSI Per-Vector Masking. This bit must be set if the function is a PF or VF within an SR-IOV Device.
23	RW	0x1	**PCI_MSI_64_BIT_ADDR_CAP**: 64 bit address capable. - If set, the function is capable of sending a 64-bit message address. - If clear, the function is not capable of sending a 64-bit message address. This bit must be set if the function is a PCI Express Endpoint. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

Bit	R/W	Reset	Description
22:20	RW	0x0	**PCI_MSI_MULTIPLE_MSG_EN**: Multiple Message Enable. Software writes to this field to indicate the number of allocated vectors (equal to or less than the number of requested vectors). The number of allocated vectors is aligned to a power of two. If a function requests four vectors (indicated by a Multiple Message Capable encoding of 010b), system software can allocate either four, two, or one vector by writing a 010b, 001b, or 000b to this field, respectively. When MSI is enabled, a function will be allocated at least 1 vector. All encodings other than the defined encodings are reserved.
19:17	RW	0x0	**PCI_MSI_MULTIPLE_MSG_CAP**: Multiple Message Capable. System software reads this field to determine the number of requested vectors. The number of requested vectors must be aligned to a power of two (if a function requires three vectors, it requests four by initializing this field to 010b). All encodings other than the defined encodings are reserved. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
16	RW	0x0	**PCI_MSI_ENABLE**: MSI Enable. - If set and the MSI-X Enable bit in the MSI-X Message Control register is clear, the function is permitted to use MSI to request service and is prohibited from using INTx interrupts. System configuration software sets this bit to enable MSI. A device driver is prohibited from writing this bit to mask a function's service request. For more details on control of INTx interrupts, see section 7.5.1.1 of PCI Express Base Specification. - If clear, the function is prohibited from using MSI to request service.
15:8	RW	0x70	**PCI_MSI_CAP_NEXT_OFFSET**: Next Capability Pointer. This field contains the offset to the next PCI Capability structure or 00h if no other items exist in the linked list of Capabilities. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
7:0	RO	0x5	**PCI_MSI_CAP_ID**: Capability ID. Indicates the MSI Capability structure. This field returns a Capability ID of 05h indicating that this is an MSI Capability structure.

PCIE_X<*j*>_EP_PF0_MSI_CAP_MSI_CAP_OFF_04H_REG_0

where <*j*> = 4, 8.

Description:This register holds the system specified message address for an MSI transaction.

PCIE_X4_EP_PF0_MSI_CAP_MSI_CAP_OFF_04H_REG_0

PCIE_X8_EP_PF0_MSI_CAP_MSI_CAP_OFF_04H_REG_0

Offset: 0x54
Read/Write: See table below

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:2	RW	0x0	**PCI_MSI_CAP_OFF_04H**: Message Address - System-specified message address. If the Message Enable bit (bit 16 of the PCI_MSI_CAP_ID_NEXT_CTRL_REG register) is set, the contents of this field specify the DWORD-aligned address (Address[31:02]) for the MSI transaction. Address[1:0] are set to 00b. Note: The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
1:0	RO	0x0	**RSVDP_0**: Reserved for future use.

PCIE_X<*j*>_EP_PF0_MSI_CAP_MSI_CAP_OFF_08H_REG_0

where <*j*> = 4, 8.

Description:For a function that supports a 32-bit message address, - bits[31:16] of this register represent the Extended Message Data, and - bits[15:0] of this register represent the Message Data For a function that supports a 64-bit message address (bit 23 in PCI_MSI_CAP_ID_NEXT_CTRL_REG register set), this register represents the Message Upper Address Register for MSI (Offset 08h). It specifies the Message Upper Address (System-specified message upper address). This register is required for PCI Express Endpoints and is optional for other function types. If the Message Enable bit (bit 0 of the Message Control register) is set, the contents of this register (if non-zero) specify the upper 32-bits of a 64-bit message address (Address[63:32]). If the contents of this register are zero, the Function uses the 32 bit address specified by the Message Address register.

PCIE_X4_EP_PF0_MSI_CAP_MSI_CAP_OFF_08H_REG_0

PCIE_X8_EP_PF0_MSI_CAP_MSI_CAP_OFF_08H_REG_0

Offset: 0x58
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PCI_MSI_CAP_OFF_0AH**: For a function that supports a 32-bit message address, this field contains Extended Message Data (System-specified message data). For the MSI Capability structures without per-vector masking, it must be implemented if the Extended Message Data Capable bit is set; otherwise, it is outside the MSI Capability structure and undefined. For the MSI Capability structures with Per-vector Masking, it must be implemented if the Extended Message Data Capable bit is set; otherwise, it is RsvdP. If the Extended Message Data Enable bit (bit 26 of the PCI_MSI_CAP_ID_NEXT_CTRL_REG) is set, the DWORD Memory Write transaction uses Extended Message Data for the upper 16 bits; otherwise, it uses 0000h for the upper 16 bits. For a function that supports a 64-bit message address, it contains upper 16 bits of the Message Upper Address. Note: The access attributes of this field are as follows: - Wire: PCI_MSI_64_BIT_ADDR_CAP || `DEFAULT_EXT_MSI_DATA_CAPABLE ? R/W : R - Dbi: PCI_MSI_64_BIT_ADDR_CAP || `DEFAULT_EXT_MSI_DATA_CAPABLE ? R/W : R
15:0	0x0	**PCI_MSI_CAP_OFF_08H**: For a function that supports a 32-bit message address, this field contains Message Data (System-specified message data). If the Message Enable bit (bit 16 of the PCI_MSI_CAP_ID_NEXT_CTRL_REG) is set, the function sends a DWORD Memory Write transaction using Message Data for the lower 16 bits. All 4 Byte Enables are set. The Multiple Message Enable field (bits 22:20 of the PCI_MSI_CAP_ID_NEXT_CTRL_REG) defines the number of low order message data bits the function is permitted to modify to generate its system software allocated vectors. For example, a Multiple Message Enable encoding of 010b indicates the function has been allocated four vectors and is permitted to modify message data bits 1 and 0 (a function modifies the lower message data bits to generate the allocated number of vectors). If the Multiple Message Enable field is 000b, the Function is not permitted to modify the message data. For a function that supports a 64-bit message address, it contains lower 16 bits of the Message Upper Address. Note: The access attributes of this field are as follows: - Wire: PCI_MSI_64_BIT_ADDR_CAP ? R/W : R - Dbi: PCI_MSI_64_BIT_ADDR_CAP ? R/W : R

PCIE_X<*j*>_EP_PF0_MSI_CAP_MSI_CAP_OFF_0CH_REG_0

where <*j*> = 4, 8.

Description:For a function that supports a 32-bit message address, this register contains the Mask Bits when the Per-Vector Masking Capable bit (bit 24 of PCI_MSI_CAP_ID_NEXT_CTRL_REG) is set. For a function that supports a 64-bit message address, this register contains Message Data.

PCIE_X4_EP_PF0_MSI_CAP_MSI_CAP_OFF_0CH_REG_0

PCIE_X8_EP_PF0_MSI_CAP_MSI_CAP_OFF_0CH_REG_0

Offset: 0x5c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**PCI_MSI_CAP_OFF_0EH**: For a function that supports a 32-bit message address, this field contains the upper Mask Bits when the Per Vector Masking Capable bit (PCI_MSI_CAP_ID_NEXT_CTRL_REG.PCI_PVM_SUPPORT) is set. For a function that supports a 64-bit message address, this field contains Message Data (System-specified message data). Note: The access attributes of this field are as follows: - Wire: (!MSI_64_EN && MSI_PVM_EN_VALUE) ? RW: MSI_64_EN && DEFAULT_EXT_MSI_DATA_CAPABLE ? RW : RO - Dbi: (!MSI_64_EN && MSI_PVM_EN_VALUE) ? RW: MSI_64_EN && DEFAULT_EXT_MSI_DATA_CAPABLE ? RW : RO
15:0	0x0	**PCI_MSI_CAP_OFF_0CH**: For a function that supports a 32-bit message address, this field contains the lower Mask Bits when the Per Vector Masking Capable bit (PCI_MSI_CAP_ID_NEXT_CTRL_REG.PCI_PVM_SUPPORT) is set. For a function that supports a 64-bit message address, this field contains Message Data (System-specified message data). If the Message Enable bit (bit 16 of the PCI_MSI_CAP_ID_NEXT_CTRL_REG) is set, the function sends a DWORD Memory Write transaction using Message Data for the lower 16 bits. All 4 Byte Enables are set. The Multiple Message Enable field (bits 22:20 of the PCI_MSI_CAP_ID_NEXT_CTRL_REG) defines the number of low order message data bits the function is permitted to modify to generate its system software allocated vectors. For example, a Multiple Message Enable encoding of 010b indicates the function has been allocated four vectors and is permitted to modify message data bits 1 and 0 (a function modifies the lower message data bits to generate the allocated number of vectors). If the Multiple Message Enable field is 000b, the Function is not permitted to modify the message data. Note: The access attributes of this field are as follows: - Wire: PCI_MSI_64_BIT_ADDR_CAP || MSI_PVM_EN ? R/W : R - Dbi: PCI_MSI_64_BIT_ADDR_CAP || MSI_PVM_EN ? R/W : R

PCIE_X<*j*>_EP_PF0_MSI_CAP_MSI_CAP_OFF_10H_REG_0

where <*j*> = 4, 8.

Description:For a function that supports a 32-bit message address, this register contains the Pending Bits when the Per-Vector Masking Capable bit (bit 24 of PCI_MSI_CAP_ID_NEXT_CTRL_REG) is set. For a function that supports a 64-bit message address, this register contains the Mask Bits when the Per-Vector Masking Capable bit (bit 24 of PCI_MSI_CAP_ID_NEXT_CTRL_REG) is set.

PCIE_X4_EP_PF0_MSI_CAP_MSI_CAP_OFF_10H_REG_0

PCIE_X8_EP_PF0_MSI_CAP_MSI_CAP_OFF_10H_REG_0

Offset: 0x60
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PCI_MSI_CAP_OFF_10H**: Used for MSI when the Per Vector Masking Capable bit (PCI_MSI_CAP_ID_NEXT_CTRL_REG.PCI_PVM_SUPPORT) is set. For 32-bit contains Pending Bits. For 64-bit, contains Mask Bits. Note: The access attributes of this field are as follows: - Wire: PCI_MSI_64_BIT_ADDR_CAP && MSI_PVM_EN ? R/W : R - Dbi: PCI_MSI_64_BIT_ADDR_CAP && MSI_PVM_EN ? R/W : R

PCIE_X<*j*>_EP_PF0_MSI_CAP_MSI_CAP_OFF_14H_REG_0

where <*j*> = 4, 8.

Description:Pending Bits Register for MSI. This register is used for a function that supports a 64-bit message address when the Per-Vector Masking Capable bit (bit 24 of PCI_MSI_CAP_ID_NEXT_CTRL_REG) is set.

PCIE_X4_EP_PF0_MSI_CAP_MSI_CAP_OFF_14H_REG_0

PCIE_X8_EP_PF0_MSI_CAP_MSI_CAP_OFF_14H_REG_0

Offset: 0x64
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PCI_MSI_CAP_OFF_14H**: Pending Bits. For each pending bit that is set, the function has a pending associated message.

PCIE_X<*j*>_EP_PF0_PCIE_CAP_PCIE_CAP_ID_PCIE_NEXT_CAP_PTR_PCIE_CAP_REG_0

where <*j*> = 4, 8.

Description:This is the PCI Express Capabilities, ID, and Next Pointer Register.

PCIE_X4_EP_PF0_PCIE_CAP_PCIE_CAP_ID_PCIE_NEXT_CAP_PTR_PCIE_CAP_REG_0

PCIE_X8_EP_PF0_PCIE_CAP_PCIE_CAP_ID_PCIE_NEXT_CAP_PTR_PCIE_CAP_REG_0

Offset: 0x70
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0002b010 (0b0000,0000,0000,0010,1011,0000,0001,0000)

Bit	R/W	Reset	Description
31	RO	0x0	**RSVDP_31**: Reserved for future use.
30	RO	0x0	**RSVD**: Reserved.
29:25	RW	0x0	**PCIE_INT_MSG_NUM**: PCIE Interrupt Message Number. Interrupt Message Number. This field indicates which MSI/MSI-X vector is used for the interrupt message generated in association with any of the status bits of this Capability structure. For MSI, the value in this field indicates the offset between the base Message Data and the interrupt message that is generated. Hardware is required to update this field so that it is correct if the number of MSI Messages assigned to the Function changes when software writes to the Multiple Message Enable field in the MSI Message Control register. For MSI-X, the value in this field indicates which MSI-X Table entry is used to generate the interrupt message. The entry must be one of the first 32 entries even if the Function implements more than 32 entries. For a given MSI-X implementation, the entry must remain constant. If both MSI and MSI-X are implemented, they are permitted to use different vectors, though software is permitted to enable only one mechanism at a time. If MSI-X is enabled, the value in this field must indicate the vector for MSI-X. If MSI is enabled or neither is enabled, the value in this field must indicate the vector for MSI. If software enables both MSI and MSI-X at the same time, the value in this field is undefined. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
24	RW	0x0	**PCIE_SLOT_IMP**: Slot Implemented. When set, this bit indicates that the Link associated with this Port is connected to a slot (as compared to being connected to a system-integrated device or being disabled). This bit is valid for Downstream Ports. This bit is undefined for Upstream Ports. Note: The access attributes of this field are as follows: - Wire: HWINIT - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
23:20	RO	0x0	**PCIE_DEV_PORT_TYPE**: Device/Port Type. Indicates the specific type of this PCI Express function. Note: Different functions in a Multi-Function Device can generally be of different types. Defined encodings for functions that implement a Type 00h PCI Configuration Space header are: Defined encodings for functions that implement a Type 01h PCI Configuration Space header are: All other encodings are Reserved. Note: Different Endpoint types have notably different requirements in Section 1.3.2 of PCI Express Base Specification regarding I/O resources, Extended Configuration Space, and other capabilities.

Bit	R/W	Reset	Description
19:16	RO	0x2	**PCIE_CAP_REG**: Capability Version. Indicates PCI-SIG defined PCI Express Capability structure version number. A version of the specification that changes the PCI Express Capability structure in a way that is not otherwise identifiable (for example, through a new Capability field) is permitted to increment this field. All such changes to the PCI Express Capability structure must be software-compatible. Software must check for Capability Version numbers that are greater than or equal to the highest number defined when the software is written, as functions reporting any such Capability Version numbers will contain a PCI Express Capability structure that is compatible with that piece of software. The controller hardwires this field to 2h for functions compliant to PCI Express Base Specification, Revision 4.0, Version 1.0. Note: This register field is sticky.
15:8	RW	0xb0	**PCIE_CAP_NEXT_PTR**: Next Capability Pointer. This field contains the offset to the next PCI Capability structure or 00h if no other items exist in the linked list of Capabilities. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
7:0	RO	0x10	**PCIE_CAP_ID**: Capability ID. Indicates the PCI Express Capability structure. This field must return a Capability ID of 10h indicating that this is a PCI Express Capability structure.

PCIE_X<*j*>_EP_PF0_PCIE_CAP_DEVICE_CAPABILITIES_REG_0

where <*j*> = 4, 8.

Description:The Device Capabilities register identifies PCI Express device function specific capabilities.

PCIE_X4_EP_PF0_PCIE_CAP_DEVICE_CAPABILITIES_REG_0

PCIE_X8_EP_PF0_PCIE_CAP_DEVICE_CAPABILITIES_REG_0

Offset: 0x74
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00008fc1 (0b0000,0000,0000,0000,1000,1111,1100,0001)

Bit	R/W	Reset	Description
31:29	RO	0x0	**RSVDP_29**: Reserved for future use.

Bit	R/W	Reset	Description
28	RW	0x0	**PCIE_CAP_FLR_CAP**: Function Level Reset Capability. A value of 1b indicates the function supports the optional Function Level Reset mechanism described in section 6.6.2 of the PCI Express Base Specification. This bit applies to Endpoints only. For all other function types the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
27:26	RO	0x0	**PCIE_CAP_CAP_SLOT_PWR_LMT_SCALE**: Captured Slot Power Limit Scale. Captured Slot Power Limit Scale (Upstream Ports only). Specifies the scale used for the Slot Power Limit Value. This value is set by the Set_Slot_Power_Limit Message or hardwired to 00b (for more details, see section 6.9 of PCI Express Base Specification).
25:18	RO	0x0	**PCIE_CAP_CAP_SLOT_PWR_LMT_VALUE**: Captured Slot Power Limit Value. Captured Slot Power Limit Value (Upstream Ports only). In combination with the Captured Slot Power Limit Scale value, specifies the upper limit on power available to the adapter. Power limit (in Watts) is calculated by multiplying the value in this field by the value in the Captured Slot Power Limit Scale field except when the Captured Slot Power Limit Scale field equals 00b (1.0x) and the Captured Slot Power Limit Value exceeds EFh, then alternative encodings are used (for more details, see section 7.5.3.9 of PCI Express Base Specification). This value is set by the Set_Slot_Power_Limit Message or hardwired to 00h (for more details, see section 6.9 of PCI Express Base Specification).
17:16	RO	0x0	**RSVDP_16**: Reserved for future use.
15	RW	0x1	**PCIE_CAP_ROLE_BASED_ERR_REPORT**: Role-Based Error Reporting. When set, this bit indicates that the function implements the functionality originally defined in the Error Reporting ECN for PCI Express Base Specification, Revision 1.0a, and later incorporated into PCI Express Base Specification, Revision 1.1. This bit must be set by all functions conforming to the ECN, PCI Express Base Specification, Revision 1.1., or subsequent PCI Express Base Specification revisions. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
14:12	RO	0x0	**RSVDP_12**: Reserved for future use.
11:9	RW	0x7	**PCIE_CAP_EP_L1_ACCPT_LATENCY**: Endpoint L1 Acceptable Latency. This field indicates the acceptable latency that an Endpoint can withstand due to the transition from L1 state to the L0 state. It is essentially an indirect measure of the Endpoint's internal buffering. Power management software uses the reported L1 Acceptable Latency number to compare against the L1 Exit Latencies reported (see below) by all components comprising the data path from this Endpoint to the Root Complex Root Port to determine whether ASPM L1 entry can be used with no loss of performance. For functions other than Endpoints, this field is Reserved and the controller hardwires it to 000b. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W (Sticky) else R(Sticky) Note: This register field is sticky.

Bit	R/W	Reset	Description
8:6	RW	0x7	**PCIE_CAP_EP_L0S_ACCPT_LATENCY**: Endpoint L0s Acceptable Latency. This field indicates the acceptable total latency that an Endpoint can withstand due to the transition from L0s state to the L0 state. It is essentially an indirect measure of the Endpoint's internal buffering. Power management software uses the reported L0s Acceptable Latency number to compare against the L0s exit latencies reported by all components comprising the data path from this Endpoint to the Root Complex Root Port to determine whether ASPM L0s entry can be used with no loss of performance. For functions other than Endpoints, this field is Reserved and the controller hardwires it to 000b. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
5	RW	0x0	**PCIE_CAP_EXT_TAG_SUPP**: Extended Tag Field Supported. This bit, in combination with the 10-Bit Tag Requester Supported bit in the Device Capabilities 2 register, indicates the maximum supported size of the Tag field as a Requester. This bit must be set if the 10-Bit Tag Requester Supported bit is set. Note: 8-bit Tag field generation must be enabled by the Extended Tag Field Enable bit in the Device Control register of the Requester Function before 8-bit Tags can be generated by the Requester. See Section 2.2.6.2 of PCI Express Base Specification for interactions with enabling the use of 10-Bit Tags. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
4:3	RW	0x0	**PCIE_CAP_PHANTOM_FUNC_SUPPORT**: Phantom Functions Supported. This field indicates the support for use of unclaimed function numbers to extend the number of outstanding transactions allowed by logically combining unclaimed function numbers (called Phantom Functions) with the Tag identifier (see Section 2.2.6.2 of PCI Express Base Specification for a description of Tag Extensions). With every Function in an ARI Device, the Phantom Functions Supported field must be set to 00b. The remainder of this field description applies only to non-ARI Multi-Function Devices. This field indicates the number of most significant bits of the Function Number portion of Requester ID that are logically combined with the Tag identifier. Note: Phantom Function support for the function must be enabled by the Phantom Functions Enable field in the Device Control register before the Function is permitted to use the Function Number field in the Requester ID for Phantom Functions. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
2:0	RW	0x1	**PCIE_CAP_MAX_PAYLOAD_SIZE**: Max_Payload_Size Supported. This field indicates the maximum payload size that the function can support for TLPs. All encodings other than the defined encodings are reserved. The functions of a Multi-Function Device are permitted to report different values for this field. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PCIE_CAP_DEVICE_CONTROL_DEVICE_STATUS_0

where <*j*> = 4, 8.

Description:This register controls PCI Express device specific parameters and provides information about PCI Express device (function) specific parameters.

PCIE_X4_EP_PF0_PCIE_CAP_DEVICE_CONTROL_DEVICE_STATUS_0

PCIE_X8_EP_PF0_PCIE_CAP_DEVICE_CONTROL_DEVICE_STATUS_0

Offset: 0x78
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00102810 (0b0000,0000,0001,0000,x010,1000,0001,0000)

Bit	R/W	Reset	Description
31:22	RO	0x0	**RSVDP_22**: Reserved for future use.
21	RO	0x0	**PCIE_CAP_TRANS_PENDING**: Transactions Pending. Endpoints: When set, this bit indicates that the function has issued Non-Posted Requests that have not been completed. A Function reports this bit cleared only when all outstanding Non-Posted Requests have completed or have been terminated by the Completion Timeout mechanism. This bit must also be cleared upon the completion of an FLR. Root and Switch Ports: The controller hardwires this bit to 0b.
20	RO	0x1	**PCIE_CAP_AUX_POWER_DETECTED**: AUX Power Detected. Functions that require Aux power report this bit as set if Aux power is detected by the function. This bit is derived by sampling the sys_aux_pwr_det input.
19	RW	0x0	**PCIE_CAP_UNSUPPORTED_REQ_DETECTED**: Unsupported Request Detected. This bit indicates that the function received an Unsupported Request. Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. For a Multi-Function Device, each function indicates status of errors as perceived by the respective function.
18	RW	0x0	**PCIE_CAP_FATAL_ERR_DETECTED**: Fatal Error Detected. This bit indicates status of Fatal errors detected. Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. For a Multi-Function device, each function indicates status of errors as perceived by the respective Function. For Functions supporting Advanced Error Handling, errors are logged in this register regardless of the settings of the Uncorrectable Error Mask register.
17	RW	0x0	**PCIE_CAP_NON_FATAL_ERR_DETECTED**: Non-Fatal Error Detected. This bit indicates status of Non-fatal errors detected. Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. For a Multi-Function device, each function indicates status of errors as perceived by the respective Function. For functions supporting Advanced Error Handling, errors are logged in this register regardless of the settings of the Uncorrectable Error Mask register.

Bit	R/W	Reset	Description
16	RW	0x0	**PCIE_CAP_CORR_ERR_DETECTED**: Correctable Error Detected. This bit indicates status of correctable errors detected. Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. For a Multi-Function device, each function indicates status of errors as perceived by the respective function. For functions supporting Advanced Error Handling, errors are logged in this register regardless of the settings of the Correctable Error Mask register.
14:12	RW	0x2	**PCIE_CAP_MAX_READ_REQ_SIZE**: Max_Read_Request_Size. This field sets the maximum Read Request size for the function as a Requester. The function must not generate Read Requests with a size exceeding the set value. For functions that do not generate Read Requests larger than 128 bytes and functions that do not generate Read Requests on their own behalf, the controller implements this field as Read Only (RO) with a value of 000b.
11	RW	0x1	**PCIE_CAP_EN_NO_SNOOP**: Enable No Snoop. If this bit is set, the function is permitted to Set the No Snoop bit in the Requester Attributes of transactions it initiates that do not require hardware enforced cache coherency (see section 2.2.6.5 in PCI Express Base Specification). Note: Setting this bit to 1b should not cause a function to set the No Snoop attribute on all transactions that it initiates. Even when this bit is set, a function is only permitted to set the No Snoop attribute on a transaction when it can guarantee that the address of the transaction is not stored in any cache in the system. The controller hardwires this bit 0b if a function would never set the No Snoop attribute in transactions it initiates. Note: The access attributes of this field are as follows: - Wire: R - Dbi: R
10	RW	0x0	**PCIE_CAP_AUX_POWER_PM_EN**: Aux Power PM Enable. This bit is derived by sampling the sys_aux_pwr_det input. When set this bit, enables a function to draw Aux power independent of PME Aux power. Functions that require Aux power on legacy operating systems should continue to indicate PME Aux power requirements. Aux power is allocated as requested in the Aux_Current field of the Power Management Capabilities register (PMC), independent of the PME_En bit in the Power Management Control/Status register (PMCSR). For Multi-Function devices, a component is allowed to draw Aux power if at least one of the functions has this bit set. Note: Functions that consume Aux power must preserve the value of this sticky register when Aux power is available. In such functions, this bit is not modified by Conventional Reset. For functions that do not implement this capability, the controller hardwires this bit to 0b. Note: This register field is sticky.
9	RO	0x0	**PCIE_CAP_PHANTOM_FUNC_EN**: Phantom Functions Enable. This bit, in combination with the 10-Bit Tag Requester Enable bit in the Device Control 2 register, determines how many Tag field bits a Requester is permitted to use. When the 10-Bit Tag Requester Enable bit is clear, - If this bit is set, it enables a function to use unclaimed functions as Phantom functions to extend the number of outstanding transaction identifiers - If this bit is clear, the function is not allowed to use Phantom functions For more details, see section 2.2.6.2 of PCI Express Base Specification. Software should not change the value of this bit while the function has outstanding Non-Posted Requests; otherwise, the result is undefined. For functions that do not implement this capability, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: DEVICE_CAPABILITIES_REG.PCIE_CAP_PHANTOM_FUNC_SUPPORT ? RW : RO - Dbi: DEVICE_CAPABILITIES_REG.PCIE_CAP_PHANTOM_FUNC_SUPPORT ? RW : RO

Bit	R/W	Reset	Description
8	RO	0x0	**PCIE_CAP_EXT_TAG_EN**: Extended Tag Field Enable. This bit, in combination with the 10-Bit Tag Requester Enable bit in the Device Control 2 register, determines how many Tag field bits a Requester is permitted to use. When the 10-Bit Tag Requester Enable bit is clear, - If the Extended Tag Field Enable bit is set, the function is permitted to use an 8-bit Tag field as a Requester - If the Extended Tag Field Enable bit is clear, the Function is restricted to a 5-bit Tag field See section 2.2.6.2 of PCI Express Base Specification for required behavior when the 10-Bit Tag Requester Enable bit is set. If software changes the value of the Extended Tag Field Enable bit while the function has outstanding Non-Posted Requests, the result is undefined. For functions that do not implement this capability, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: DEVICE_CAPABILITIES_REG.PCIE_CAP_EXT_TAG_SUPP ? RW : RO - Dbi: DEVICE_CAPABILITIES_REG.PCIE_CAP_EXT_TAG_SUPP ? RW : RO
7:5	RW	0x0	**PCIE_CAP_MAX_PAYLOAD_SIZE_CS**: Max_Payload_Size. This field sets maximum TLP payload size for the Function. As a Receiver, the Function must handle TLPs as large as the set value. As a Transmitter, the Function must not generate TLPs exceeding the set value. Permissible values that can be programmed are indicated by the Max_Payload_Size Supported field (PCIE_CAP_MAX_PAYLOAD_SIZE) in the Device Capabilities (DEVICE_CAPABILITIES_REG) register (for more details, see section 7.5.3.3 of PCI Express Base Specification). This field sets the maximum Read Request size for the function as a Requester. The function must not generate Read Requests with a size exceeding the set value. For Functions that support only the 128-byte max payload size, the controller hardwires this field to 000b. System software is not required to program the same value for this field for all the Functions of a Multi-Function device (for more details, see section 2.2.2 of PCI Express Base Specification). For ARI Devices, Max_Payload_Size is determined solely by the setting in Function0. The settings in the other Functions always return whatever value software programmed for each, but otherwise are ignored by the component.
4	RW	0x1	**PCIE_CAP_EN_REL_ORDER**: Enable Relaxed Ordering. If this bit is set, the function is permitted to set the Relaxed Ordering bit in the Attributes field of transactions it initiates that do not require strong write ordering (for more details, see section 2.2.6.4 and section 2.4 of PCI Express Base Specification). For a function that never sets the Relaxed Ordering attribute in transactions it initiates as a Requester, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
3	RW	0x0	**PCIE_CAP_UNSUPPORT_REQ_REP_EN**: Unsupported Request Reporting Enable. This bit, in conjunction with other bits, controls the signaling of Unsupported Request Errors by sending error Messages (for more details, see section 6.2.5 and section 6.2.6 of PCI Express Base Specification). For a Multi-Function Device, this bit controls error reporting for each Function from point-of-view of the respective Function.
2	RW	0x0	**PCIE_CAP_FATAL_ERR_REPORT_EN**: Fatal Error Reporting Enable. This bit, in conjunction with other bits, controls sending ERR_FATAL Messages (for more details, see section 6.2.5 and section 6.2.6 of of PCI Express Base Specification). For a Multi-Function device, this bit controls error reporting for each function from point-of-view of the respective function. For a Root Port, the reporting of Fatal errors is internal to the root. No external ERR_FATAL Message is generated.

Bit	R/W	Reset	Description
1	RW	0x0	**PCIE_CAP_NON_FATAL_ERR_REPORT_EN**: Non-Fatal Error Reporting Enable. This bit, in conjunction with other bits, controls sending ERR_NONFATAL Messages (for more details, see section 6.2.5 and Section 6.2.6 of PCI Express Base Specification). For a Multi-Function Device, this bit controls error reporting for each function from point-of-view of the respective Function. For a Root Port, the reporting of Non-fatal errors is internal to the root. No external ERR_NONFATAL Message is generated.
0	RW	0x0	**PCIE_CAP_CORR_ERR_REPORT_EN**: Correctable Error Reporting Enable. This bit, in conjunction with other bits, controls sending ERR_COR Messages (for more details, see section 6.2.5, section 6.2.6, and section 6.2.10.2 of PCI Express Base Specification). For a Multi-Function device, this bit controls error reporting for each function from point-of-view of the respective function. For a Root Port, the reporting of correctable errors is internal to the root. No external ERR_COR Message is generated.

PCIE_X<*j*>_EP_PF0_PCIE_CAP_LINK_CAPABILITIES_REG_0

where <*j*> = 4, 8.

Description:The Link Capabilities register identifies PCI Express Link specific capabilities.

Offset: 0x7c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X4_EP_PF0_PCIE_CAP_LINK_CAPABILITIES_REG_0

Reset: 0x00474c44 (0b0000,0000,0100,0111,0100,1100,0100,0100)

Bit	R/W	Reset	Description
31:24	RW	0x0	**PCIE_CAP_PORT_NUM**: Port Number. This field indicates the PCI Express Port number for the given PCI Express Link. Multi-Function Devices associated with an Upstream Port must report the same value in this field for all functions. Note: The access attributes of this field are as follows: - Wire: HWINIT - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
23	RO	0x0	**RSVDP_23**: Reserved for future use.
22	RW	0x1	**PCIE_CAP_ASPM_OPT_COMPLIANCE**: ASPM Optionality Compliance. This field must be set to 1b in all functions. Components implemented against certain earlier versions of this specification will have this bit set to 0b. Software is permitted to use the value of this bit to help determine whether to enable ASPM or whether to run ASPM compliance tests. Note: The access attributes of this field are as follows: - Wire: HWINIT - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R

Bit	R/W	Reset	Description
21	RO	0x0	**PCIE_CAP_LINK_BW_NOT_CAP**: Link Bandwidth Notification Capability. A value of 1b indicates support for the Link Bandwidth Notification status and interrupt mechanisms. This capability is required for all Root Ports and Switch Downstream Ports supporting Links wider than x1 and/or multiple Link speeds. This field is not applicable and is Reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches. For functions that do not implement the Link Bandwidth Notification Capability the controller hardwires this bit to 0b. Note: This register field is sticky.
20	RO	0x0	**PCIE_CAP_DLL_ACTIVE_REP_CAP**: Data Link Layer Link Active Reporting Capable. For a Downstream Port, the controller hardwires this bit to 1b if the component supports the optional capability of reporting the DL_Active state of the Data Link Control and Management State Machine. For a hot-plug capable Downstream Port (as indicated by the Hot-Plug Capable bit of the Slot Capabilities register) or a Downstream Port that supports Link speeds greater than 5.0 GT/s, the controller hardwires this bit to 1b. For Upstream Ports and components that do not support this optional capability, the controller hardwires this bit to 0b.
19	RO	0x0	**PCIE_CAP_SURPRISE_DOWN_ERR_REP_CAP**: Surprise Down Error Reporting Capable. For a Downstream Port, this bit must be set if the component supports the optional capability of detecting and reporting a Surprise Down error condition. For Upstream Ports and components that do not support this optional capability, the controller hardwires this bit to 0b. Note: This register field is sticky.
18	RW	0x1	**PCIE_CAP_CLOCK_POWER_MAN**: Clock Power Management. For Upstream Ports, a value of 1b in this bit indicates that the component tolerates the removal of any reference clock(s) via the "clock request" (CLKREQ#) mechanism when the Link is in the L1 and L2/L3 Ready Link states. A value of 0b indicates the component does not have this capability and that reference clock(s) must not be removed in these Link states. L1 PM Substates defines other semantics for the CLKREQ# signal, which are managed independently of Clock Power Management. This Capability is applicable only in form factors that support "clock request" (CLKREQ#) capability. For a Multi-Function device associated with an Upstream Port, each Function indicates its capability independently. Power Management configuration software must only permit reference clock removal if all functions of the Multi-Function device indicate a 1b in this bit. For ARI Devices, all Functions must indicate the same value in this bit. For Downstream Ports, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

Bit	R/W	Reset	Description
17:15	RW	0x6	**PCIE_CAP_L1_EXIT_LATENCY**: L1 Exit Latency. This field indicates the L1 exit latency for the given PCI Express Link. The value reported indicates the length of time this Port requires to complete transition from ASPM L1 to L0. If ASPM L1 is not supported, the value is undefined. Note: Exit latencies may be influenced by PCI Express reference clock configuration depending upon whether a component uses a common or separate reference clock. Multi-Function Devices associated with an Upstream Port must report the same value in this field for all functions. There are two each of these register fields, this one and a shadow one at the same address. The Common Clock bit (PCIE_CAP_COMMON_CLK_CONFIG) of the Link Control Register (LINK_CONTROL_LINK_STATUS_REG) determines which one is used by the controller and which one is accessed by a read request. Common Clock operation is supported (possible) in the controller when one or more of the following expressions is true: - CX_NFTS !=CX_COMM_NFTS - DEFAULT_L0S_EXIT_LATENCY !=DEFAULT_COMM_L0S_EXIT_LATENCY - DEFAULT_L1_EXIT_LATENCY !=DEFAULT_COMM_L1_EXIT_LATENCY Common Clock operation is enabled in the controller when you set the Common Clock bit (PCIE_CAP_COMMON_CLK_CONFIG) of the Link Control Register (LINK_CONTROL_LINK_STATUS_REG). The assertion of CS2 (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) is required to write to the shadow field at this location. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
14:12	RW	0x4	**PCIE_CAP_L0S_EXIT_LATENCY**: L0s Exit Latency. This field indicates the L0s exit latency for the given PCI Express Link. The value reported indicates the length of time this Port requires to complete transition from L0s to L0. If L0s is not supported, the value is undefined; however, see the Implementation Note "Potential Issues With Legacy Software When L0s is Not Supported" in section 5.4.1.1 of PCI Express Base Specification for the recommended value. Note: Exit latencies may be influenced by PCI Express reference clock configuration depending upon whether a component uses a common or separate reference clock. Multi-Function Devices associated with an Upstream Port must report the same value in this field for all functions. There are two each of these register fields, this one and a shadow one at the same address. The Common Clock bit (PCIE_CAP_COMMON_CLK_CONFIG) of the Link Control Register (LINK_CONTROL_LINK_STATUS_REG) determines which one is used by the controller and which one is accessed by a read request. Common Clock operation is supported (possible) in the controller when one or more of the following expressions is true: - CX_NFTS ! =CX_COMM_NFTS - DEFAULT_L0S_EXIT_LATENCY ! =DEFAULT_COMM_L0S_EXIT_LATENCY - DEFAULT_L1_EXIT_LATENCY ! =DEFAULT_COMM_L1_EXIT_LATENCY Common Clock operation is enabled in the controller when you set the Common Clock bit (PCIE_CAP_COMMON_CLK_CONFIG) of the Link Control Register (LINK_CONTROL_LINK_STATUS_REG). The assertion of CS2 (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) is required to write to the shadow field at this location. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
11:10	RW	0x3	**PCIE_CAP_ACTIVE_STATE_LINK_PM_SUPPORT**: Active State Power Management (ASPM) Support. This field indicates the level of ASPM supported on the given PCI Express Link. For more details on ASPM support requirements, see section 5.4.1 of PCI Express Base Specification. Multi-Function devices associated with an Upstream Port must report the same value in this field for all functions. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

Bit	R/W	Reset	Description
9:4	RW	0x4	**PCIE_CAP_MAX_LINK_WIDTH**: Maximum Link Width. This field indicates the maximum Link width (xN - corresponding to N Lanes) implemented by the component. This value is permitted to exceed the number of Lanes routed to the slot (Downstream Port), adapter connector (Upstream Port), or in the case of component-to-component connections, the actual wired connection width. All encodings other than the defined encodings are reserved. Multi-Function devices associated with an Upstream Port must report the same value in this field for all functions. In M-PCIe mode, the reset and dynamic values of this field are calculated by the controller. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
3:0	RW	0x4	**PCIE_CAP_MAX_LINK_SPEED**: Max Link Speed. This field indicates the maximum Link speed of the associated Port. The encoded value specifies a bit location in the Supported Link Speeds Vector (in the Link Capabilities 2 register) that corresponds to the maximum Link speed. All encodings other than the defined encodings are reserved. Multi-Function Devices associated with an Upstream Port must report the same value in this field for all functions. In M-PCIe mode, the reset and dynamic values of this field are calculated by the controller. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

PCIE_X8_EP_PF0_PCIE_CAP_LINK_CAPABILITIES_REG_0

Reset: 0x00474c84 (0b0000,0000,0100,0111,0100,1100,1000,0100)

Bit	R/W	Reset	Description
Bit	R/W	Reset	Description
31:24	RW	0x0	**PCIE_CAP_PORT_NUM**: Port Number. This field indicates the PCI Express Port number for the given PCI Express Link. Multi-Function Devices associated with an Upstream Port must report the same value in this field for all functions. Note: The access attributes of this field are as follows: - Wire: HWINIT - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
23	RO	0x0	**RSVDP_23**: Reserved for future use.
22	RW	0x1	**PCIE_CAP_ASPM_OPT_COMPLIANCE**: ASPM Optionality Compliance. This field must be set to 1b in all functions. Components implemented against certain earlier versions of this specification will have this bit set to 0b. Software is permitted to use the value of this bit to help determine whether to enable ASPM or whether to run ASPM compliance tests. Note: The access attributes of this field are as follows: - Wire: HWINIT - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
21	RO	0x0	**PCIE_CAP_LINK_BW_NOT_CAP**: Link Bandwidth Notification Capability. A value of 1b indicates support for the Link Bandwidth Notification status and interrupt mechanisms. This capability is required for all Root Ports and Switch Downstream Ports supporting Links wider than x1 and/or multiple Link speeds. This field is not applicable and is Reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches. For functions that do not implement the Link Bandwidth Notification Capability the controller hardwires this bit to 0b. Note: This register field is sticky.

Bit	R/W	Reset	Description
Bit	**R/W**	**Reset**	**Description**
20	RO	0x0	**PCIE_CAP_DLL_ACTIVE_REP_CAP**: Data Link Layer Link Active Reporting Capable. For a Downstream Port, the controller hardwires this bit to 1b if the component supports the optional capability of reporting the DL_Active state of the Data Link Control and Management State Machine. For a hot-plug capable Downstream Port (as indicated by the Hot-Plug Capable bit of the Slot Capabilities register) or a Downstream Port that supports Link speeds greater than 5.0 GT/s, the controller hardwires this bit to 1b. For Upstream Ports and components that do not support this optional capability, the controller hardwires this bit to 0b.
19	RO	0x0	**PCIE_CAP_SURPRISE_DOWN_ERR_REP_CAP**: Surprise Down Error Reporting Capable. For a Downstream Port, this bit must be set if the component supports the optional capability of detecting and reporting a Surprise Down error condition. For Upstream Ports and components that do not support this optional capability, the controller hardwires this bit to 0b. Note: This register field is sticky.
18	RW	0x1	**PCIE_CAP_CLOCK_POWER_MAN**: Clock Power Management. For Upstream Ports, a value of 1b in this bit indicates that the component tolerates the removal of any reference clock(s) via the "clock request" (CLKREQ#) mechanism when the Link is in the L1 and L2/L3 Ready Link states. A value of 0b indicates the component does not have this capability and that reference clock(s) must not be removed in these Link states. L1 PM Substates defines other semantics for the CLKREQ# signal, which are managed independently of Clock Power Management. This Capability is applicable only in form factors that support "clock request" (CLKREQ#) capability. For a Multi-Function device associated with an Upstream Port, each Function indicates its capability independently. Power Management configuration software must only permit reference clock removal if all functions of the Multi-Function device indicate a 1b in this bit. For ARI Devices, all Functions must indicate the same value in this bit. For Downstream Ports, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
17:15	RW	0x6	**PCIE_CAP_L1_EXIT_LATENCY**: L1 Exit Latency. This field indicates the L1 exit latency for the given PCI Express Link. The value reported indicates the length of time this Port requires to complete transition from ASPM L1 to L0. If ASPM L1 is not supported, the value is undefined. Note: Exit latencies may be influenced by PCI Express reference clock configuration depending upon whether a component uses a common or separate reference clock. Multi-Function Devices associated with an Upstream Port must report the same value in this field for all functions. There are two each of these register fields, this one and a shadow one at the same address. The Common Clock bit (PCIE_CAP_COMMON_CLK_CONFIG) of the Link Control Register (LINK_CONTROL_LINK_STATUS_REG) determines which one is used by the controller and which one is accessed by a read request. Common Clock operation is supported (possible) in the controller when one or more of the following expressions is true: - CX_NFTS !=CX_COMM_NFTS - DEFAULT_L0S_EXIT_LATENCY !=DEFAULT_COMM_L0S_EXIT_LATENCY - DEFAULT_L1_EXIT_LATENCY !=DEFAULT_COMM_L1_EXIT_LATENCY Common Clock operation is enabled in the controller when you set the Common Clock bit (PCIE_CAP_COMMON_CLK_CONFIG) of the Link Control Register (LINK_CONTROL_LINK_STATUS_REG). The assertion of CS2 (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) is required to write to the shadow field at this location. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

Bit	R/W	Reset	Description
Bit	**R/W**	**Reset**	**Description**
14:12	RW	0x4	**PCIE_CAP_L0S_EXIT_LATENCY**: L0s Exit Latency. This field indicates the L0s exit latency for the given PCI Express Link. The value reported indicates the length of time this Port requires to complete transition from L0s to L0. If L0s is not supported, the value is undefined; however, see the Implementation Note "Potential Issues With Legacy Software When L0s is Not Supported" in section 5.4.1.1 of PCI Express Base Specification for the recommended value. Note: Exit latencies may be influenced by PCI Express reference clock configuration depending upon whether a component uses a common or separate reference clock. Multi-Function Devices associated with an Upstream Port must report the same value in this field for all functions. There are two each of these register fields, this one and a shadow one at the same address. The Common Clock bit (PCIE_CAP_COMMON_CLK_CONFIG) of the Link Control Register (LINK_CONTROL_LINK_STATUS_REG) determines which one is used by the controller and which one is accessed by a read request. Common Clock operation is supported (possible) in the controller when one or more of the following expressions is true: - CX_NFTS ! =CX_COMM_NFTS - DEFAULT_L0S_EXIT_LATENCY ! =DEFAULT_COMM_L0S_EXIT_LATENCY - DEFAULT_L1_EXIT_LATENCY ! =DEFAULT_COMM_L1_EXIT_LATENCY Common Clock operation is enabled in the controller when you set the Common Clock bit (PCIE_CAP_COMMON_CLK_CONFIG) of the Link Control Register (LINK_CONTROL_LINK_STATUS_REG). The assertion of CS2 (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) is required to write to the shadow field at this location. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
11:10	RW	0x3	**PCIE_CAP_ACTIVE_STATE_LINK_PM_SUPPORT**: Active State Power Management (ASPM) Support. This field indicates the level of ASPM supported on the given PCI Express Link. For more details on ASPM support requirements, see section 5.4.1 of PCI Express Base Specification. Multi-Function devices associated with an Upstream Port must report the same value in this field for all functions. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
9:4	RW	0x8	**PCIE_CAP_MAX_LINK_WIDTH**: Maximum Link Width. This field indicates the maximum Link width (xN - corresponding to N Lanes) implemented by the component. This value is permitted to exceed the number of Lanes routed to the slot (Downstream Port), adapter connector (Upstream Port), or in the case of component-to-component connections, the actual wired connection width. All encodings other than the defined encodings are reserved. Multi-Function devices associated with an Upstream Port must report the same value in this field for all functions. In M-PCIe mode, the reset and dynamic values of this field are calculated by the controller. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
3:0	RW	0x4	**PCIE_CAP_MAX_LINK_SPEED**: Max Link Speed. This field indicates the maximum Link speed of the associated Port. The encoded value specifies a bit location in the Supported Link Speeds Vector (in the Link Capabilities 2 register) that corresponds to the maximum Link speed. All encodings other than the defined encodings are reserved. Multi-Function Devices associated with an Upstream Port must report the same value in this field for all functions. In M-PCIe mode, the reset and dynamic values of this field are calculated by the controller. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PCIE_CAP_LINK_CONTROL_LINK_STATUS_REG_0

where <*j*> = 4, 8.

Description:This register controls and provides information about PCI Express Link specific parameters.

PCIE_X4_EP_PF0_PCIE_CAP_LINK_CONTROL_LINK_STATUS_REG_0

PCIE_X8_EP_PF0_PCIE_CAP_LINK_CONTROL_LINK_STATUS_REG_0

Offset: 0x80
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x10110000 (0b0001,0000,0001,0001,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RO	0x0	**PCIE_CAP_LINK_AUTO_BW_STATUS**: Link Autonomous Bandwidth Status. This bit is set by hardware to indicate that hardware has autonomously changed Link speed or width, without the Port transitioning through DL_Down status, for reasons other than to attempt to correct unreliable Link operation. This bit must be set if the Physical Layer reports a speed or width change was initiated by the Downstream component that was indicated as an autonomous change. The write value is gated with the PCIE_CAP_LINK_BW_NOT_CAP field in LINK_CAPABILITIES_REG. This bit is not applicable and is Reserved for Endpoints, PCI Express-to-PCI/PCI-X bridges, and Upstream Ports of Switches. For functions that do not implement the Link Bandwidth Notification Capability, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: RSVDP - Dbi: R
30	RO	0x0	**PCIE_CAP_LINK_BW_MAN_STATUS**: Link Bandwidth Management Status. This bit is set by hardware to indicate that either of the following has occurred without the Port transitioning through DL_Down status: - A Link retraining has completed following a write of 1b to the Retrain Link bit. Note: This bit is set following any write of 1b to the Retrain Link bit, including when the Link is in the process of retraining for some other reason. - Hardware has changed Link speed or width to attempt to correct unreliable Link operation, either through an LTSSM timeout or a higher level process. This bit must be set if the Physical Layer reports a speed or width change was initiated by the Downstream component that was not indicated as an autonomous change. This bit is not applicable and is Reserved for Endpoints, PCI Express-to-PCI/PCI-X bridges, and Upstream Ports of Switches. For functions that do not implement the Link Bandwidth Notification Capability, the controller hardwires this bit to 0b. The default value of this bit is 0b. The write value is gated with the PCIE_CAP_LINK_BW_NOT_CAP field in LINK_CAPABILITIES_REG. Note: The access attributes of this field are as follows: - Wire: RSVDP - Dbi: R

Bit	R/W	Reset	Description
29	RO	0x0	**PCIE_CAP_DLL_ACTIVE**: Data Link Layer Link Active. This bit indicates the status of the Data Link Control and Management State Machine. It returns a 1b to indicate the DL_Active state, 0b otherwise. This bit must be implemented if the Data Link Layer Link Active Reporting Capable bit is 1b. Otherwise, the controller hardwires it to 0b.
28	RW	0x1	**PCIE_CAP_SLOT_CLK_CONFIG**: Slot Clock Configuration. This bit indicates that the component uses the same physical reference clock that the platform provides on the connector. If the device uses an independent clock irrespective of the presence of a reference clock on the connector, this bit must be clear. For a Multi-Function Device, each Function must report the same value for this bit. Note: The access attributes of this field are as follows: - Wire: HWINIT - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
27	RO	0x0	**PCIE_CAP_LINK_TRAINING**: Link Training. This read-only bit indicates that the Physical Layer LTSSM is in the Configuration or Recovery state, or that 1b was written to the Retrain Link bit but Link training has not yet begun. Hardware clears this bit when the LTSSM exits the Configuration/Recovery state. This bit is not applicable and Reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches, and the controller hardwires it to 0b. Note: The access attributes of this field are as follows: - Wire: RSVDP - Dbi: R
26	RO	0x0	**RSVDP_26**: Reserved for future use.
25:20	RO	0x1	**PCIE_CAP_NEGO_LINK_WIDTH**: Negotiated Link Width. This field indicates the negotiated width of the given PCI Express Link. All encodings other than the defined encodings are reserved. The value in this field is undefined when the Link is not up.
19:16	RO	0x1	**PCIE_CAP_LINK_SPEED**: Current Link Speed. This field indicates the negotiated Link speed of the given PCI Express Link. The encoded value specifies a bit location in the Supported Link Speeds Vector (in the Link Capabilities 2 register) that corresponds to the current Link speed. All encodings other than the defined encodings are reserved. The value in this field is undefined when the Link is not up.
15:14	RO	0x0	**PCIE_CAP_DRS_SIGNALING_CONTROL**: DRS Signaling Control. Indicates the mechanism used to report reception of a DRS message. Must be implemented for Downstream Ports with the DRS Supported bit Set in the Link Capabilities 2 Register. Encodings are: If DRS Supported is set, receiving a DRS Message will set DRS Message Received in the Link Status 2 Register but will otherwise have no effect If the DRS Message Received bit in the Link Status 2 Register transitions from 0 to 1, and either MSI or MSI-X is enabled, an MSI or MSI-X interrupt is generated using the vector in Interrupt Message Number (section 7.5.3.2) If the DRS Message Received bit in the Link Status 2 Register transitions from 0 to 1, the Port must send an FRS Message Upstream with the FRS Reason field set to DRS Message Received. Behavior is undefined if this field is set to 10b and the FRS Supported bit in the Device Capabilities 2 Register is Clear. Behavior is undefined if this field is set to 11b. For Downstream Ports with the DRS Supported bit clear in the Link Capabilities 2 register, the controller hardwires this field to 00b. This field is Reserved for Upstream Ports.
13:12	RO	0x0	**RSVDP_12**: Reserved for future use.

Bit	R/W	Reset	Description
11	RW	0x0	**PCIE_CAP_LINK_AUTO_BW_INT_EN**: Link Autonomous Bandwidth Management Interrupt Enable. When set, this bit enables the generation of an interrupt to indicate that the Link Autonomous Bandwidth Status bit has been set. The write value is gated with the PCIE_CAP_LINK_BW_NOT_CAP field in LINK_CAPABILITIES_REG. This bit is not applicable and is Reserved for Endpoints, PCI Express-to-PCI/PCI-X bridges, and Upstream Ports of Switches. For functions that do not implement the Link Bandwidth Notification Capability, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: LINK_CAPABILITIES_REG.PCIE_CAP_LINK_BW_NOT_CAP ? RW : RO - Dbi: LINK_CAPABILITIES_REG.PCIE_CAP_LINK_BW_NOT_CAP ? RW : RO
10	RW	0x0	**PCIE_CAP_LINK_BW_MAN_INT_EN**: Link Bandwidth Management Interrupt Enable. When set, this bit enables the generation of an interrupt to indicate that the Link Bandwidth Management Status bit has been set. The write value is gated with the PCIE_CAP_LINK_BW_NOT_CAP field in LINK_CAPABILITIES_REG. This bit is not applicable and is Reserved for Endpoints, PCI Express-to-PCI/PCI-X bridges, and Upstream Ports of Switches. For functions that do not implement the Link Bandwidth Notification Capability, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: LINK_CAPABILITIES_REG.PCIE_CAP_LINK_BW_NOT_CAP ? RW : RO - Dbi: LINK_CAPABILITIES_REG.PCIE_CAP_LINK_BW_NOT_CAP ? RW : RO
9	RW	0x0	**PCIE_CAP_HW_AUTO_WIDTH_DISABLE**: Hardware Autonomous Width Disable. When set, this bit disables hardware from changing the Link width for reasons other than attempting to correct unreliable Link operation by reducing Link width. For a Multi-Function Device associated with an Upstream Port, the bit in Function 0 is of type RW, and only Function 0 controls the component's Link behavior. In all other Functions of that device, this bit is of type RsvdP. For components that do not implement the ability autonomously to change Link width, the ciontroller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
8	RW	0x0	**PCIE_CAP_EN_CLK_POWER_MAN**: Enable Clock Power Management. Applicable only for Upstream Ports and with form factors that support a "Clock Request" (CLKREQ#) mechanism, this bit operates as follows: For a non-ARI Multi-Function Device, power-management-configuration software must only Set this bit if all Functions of the Multi-Function Device indicate a 1b in the Clock Power Management bit of the Link Capabilities register. The component is permitted to use the CLKREQ# signal to power manage Link clock only if this bit is Set for all Functions. For ARI Devices, Clock Power Management is enabled solely by the setting in Function 0. The settings in the other Functions always return whatever value software programmed for each, but otherwise are ignored by the component. The CLKREQ# signal may also be controlled via the L1 PM Substates mechanism. Such control is not affected by the setting of this bit. For Downstream Ports and components that do not support Clock Power Management (as indicated by a 0b value in the Clock Power Management bit of the Link Capabilities register), the controller hardwires this bit to 0b. The write value is gated with the PCIE_CAP_CLOCK_POWER_MAN field in LINK_CAPABILITIES_REG. Note: The access attributes of this field are as follows: - Wire: LINK_CAPABILITIES_REG.PCIE_CAP_CLOCK_POWER_MAN ? RWS : ROS - Dbi: LINK_CAPABILITIES_REG.PCIE_CAP_CLOCK_POWER_MAN ? RWS : ROS Note: This register field is sticky.

Bit	R/W	Reset	Description
7	RW	0x0	**PCIE_CAP_EXTENDED_SYNCH**: Extended Synch. When set, this bit forces the transmission of additional Ordered Sets when exiting the L0s state (see section 4.2.4.5 of PCI Express Base Specification) and when in the Recovery state (see section 4.2.6.4.1 of PCI Express Base Specification). This mode provides external devices (for example, logic analyzers) monitoring the Link time to achieve bit and Symbol lock before the Link enters the L0 state and resumes communication. For Multi-Function devices if any function has this bit set, then the component must transmit the additional Ordered Sets when exiting L0s or when in Recovery.
6	RW	0x0	**PCIE_CAP_COMMON_CLK_CONFIG**: Common Clock Configuration. When set, this bit indicates that this component and the component at the opposite end of this Link are operating with a distributed common reference clock. A value of 0b indicates that this component and the component at the opposite end of this Link are operating with asynchronous reference clock. For non-ARI Multi-Function Devices, software must program the same value for this bit in all Functions. If not all Functions are Set, then the component must as a whole assume that its reference clock is not common with the Upstream component. For ARI Devices, Common Clock Configuration is determined solely by the setting in Function 0. The settings in the other Functions always return whatever value software programmed for each, but otherwise are ignored by the component. Components utilize this common clock configuration information to report the correct L0s and L1 Exit Latencies. After changing the value in this bit in both components on a Link, software must trigger the Link to retrain by writing a 1b to the Retrain Link bit of the Downstream Port.
5	RW	0x0	**PCIE_CAP_RETRAIN_LINK**: Retrain Link. A write of 1b to this bit initiates Link retraining by directing the Physical Layer LTSSM to the Recovery state. If the LTSSM is already in Recovery or Configuration, re-entering Recovery is permitted but not required. If the Port is in DPC when a write of 1b to this bit occurs, the result is undefined. Reads of this bit always return 0b. It is permitted to write 1b to this bit while simultaneously writing modified values to other fields in this register. If the LTSSM is not already in Recovery or Configuration, the resulting Link training must use the modified values. If the LTSSM is already in Recovery or Configuration, the modified values are not required to affect the Link training that's already in progress. This bit is not applicable and is Reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches. This bit always returns 0b when read. Note: The access attributes of this field are as follows: - Wire: see description - Dbi: see description
4	RW	0x0	**PCIE_CAP_LINK_DISABLE**: Link Disable. This bit disables the Link by directing the LTSSM to the Disabled state when set; this bit is Reserved on Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches. Writes to this bit are immediately reflected in the value read from the bit, regardless of actual Link state. After clearing this bit, software must honor timing requirements defined in Section 6.6.1 with respect to the first Configuration Read following a Conventional Reset. In a DSP that supports crosslink, the controller gates the write value with the CROSS_LINK_EN field in PORT_LINK_CTRL_OFF. Note: The access attributes of this field are as follows: - Wire: CX_CROSSLINK_ENABLE=1 && PORT_LINK_CTRL_OFF.CROSS_LINK_EN=1||CX_CROSSLINK_ENABLE=0 && dsp=1 ? RW : RO - Dbi: CX_CROSSLINK_ENABLE=1 && PORT_LINK_CTRL_OFF.CROSS_LINK_EN=1||CX_CROSSLINK_ENABLE=0 && dsp=1? RW : RO

Bit	R/W	Reset	Description
3	RW	0x0	**PCIE_CAP_RCB**: Read Completion Boundary (RCB). Root Ports: Indicates the RCB value for the Root Port. Refer to section 2.3.1.1 of PCI Express Base Specification for the definition of the parameter RCB. The controller hardwires this bit for a Root Port and returns its RCB support capabilities. Endpoints and Bridges: Optionally set by configuration software to indicate the RCB value of the Root Port Upstream from the Endpoint or Bridge. Refer to Section 2.3.1.1 of PCI Express Base Specification for the definition of the parameter RCB. Configuration software must only set this bit if the Root Port Upstream from the Endpoint or Bridge reports an RCB value of 128 bytes (a value of 1b in the Read Completion Boundary bit). For functions that do not implement this feature, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
2	RO	0x0	**RSVDP_2**: Reserved for future use.
1:0	RW	0x0	**PCIE_CAP_ACTIVE_STATE_LINK_PM_CONTROL**: Active State Power Management (ASPM) Control. This field controls the level of ASPM enabled on the given PCI Express Link. See section 5.4.1.3 of PCI Express Base Specification for requirements on when and how to enable ASPM. Note: "L0s Entry Enabled" enables the Transmitter to enter L0s. If L0s is supported, the Receiver must be capable of entering L0s even when the Transmitter is disabled from entering L0s (00b or 10b). ASPM L1 must be enabled by software in the Upstream component on a Link prior to enabling ASPM L1 in the Downstream component on that Link. When disabling ASPM L1, software must disable ASPM L1 in the Downstream component on a Link prior to disabling ASPM L1 in the Upstream component on that Link. ASPM L1 must only be enabled on the Downstream component if both components on a Link support ASPM L1. For Multi-Function Devices (including ARI Devices), it is recommended that software program the same value for this field in all Functions. For non-ARI Multi-Function Devices, only capabilities enabled in all Functions are enabled for the component as a whole. For ARI Devices, ASPM Control is determined solely by the setting in Function0, regardless of Function 0's D-state. The settings in the other Functions always return whatever value software programmed for each, but otherwise are ignored by the component. Software must not enable L0s in either direction on a given Link unless components on both sides of the Link each support L0s; otherwise, the result is undefined.

PCIE_X<*j*>_EP_PF0_PCIE_CAP_DEVICE_CAPABILITIES2_REG_0

where <*j*> = 4, 8.

Description:This register identifies PCI Express device specific capabilities; in addition to the Device Capabilities Register.

PCIE_X4_EP_PF0_PCIE_CAP_DEVICE_CAPABILITIES2_REG_0

PCIE_X8_EP_PF0_PCIE_CAP_DEVICE_CAPABILITIES2_REG_0

Offset: 0x94
Read/Write: See table below

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0001081f (0bx000,0000,xxxx,xx01,0000,1000,0001,1111)

Bit	R/W	Reset	Description
30:24	RO	0x0	**RSVDP_24**: Reserved for future use.
17	RO	0x0	**PCIE_CAP2_10_BIT_TAG_REQ_SUPPORT**: 10-Bit Tag Requester Supported. If this bit is set, the Function supports 10-Bit Tag Requester capability; otherwise, the Function does not. This bit must not be set if the 10-Bit Tag Completer Supported bit is clear. Note: 10-Bit Tag field generation must be enabled by the 10-Bit Tag Requester Enable bit in the Device Control 2 register of the Requester Function before 10-Bit Tags can be generated by the Requester. For more details, see section 2.2.6.2. of PCI Express Base Specification.
16	RO	0x1	**PCIE_CAP2_10_BIT_TAG_COMP_SUPPORT**: 10-Bit Tag Completer Supported. If this bit is set, the Function supports 10-Bit Tag Completer capability; otherwise, the Function does not. For more details, see section 2.2.6.2. of PCI Express Base Specification.
15:14	RO	0x0	**PCIE_CAP2_LN_SYS_CLS**: LN System CLS. Applicable only to Root Ports and RCRBs; must be 00b for all other Function types. This field indicates if the Root Port or RCRB supports LN protocol as an LN Completer, and if so, what cacheline size is in effect. All encodings other than the defined encodings are reserved. Note: This register field is sticky.
13	RO	0x0	**PCIE_CAP_TPH_CMPLT_SUPPORT_1**: TPH Completer Supported Bit 1.
12	RO	0x0	**PCIE_CAP_TPH_CMPLT_SUPPORT_0**: TPH Completer Supported Bit 0. Value of this bit along with TPH Completer Supported Bit 1 indicates Completer support for TPH or Extended TPH. Applicable only to Root Ports and Endpoints. For all other Functions, this field is Reserved. When TPH completer support bit 0 is clear. When TPH completer support bit 1 is set. For more details, see section 6.17 of PCI Express Base Specification.
11	RW	0x1	**PCIE_CAP_LTR_SUPP**: LTR Mechanism Supported. A value of 1b indicates support for the optional Latency Tolerance Reporting (LTR) mechanism. Root Ports, Switches and Endpoints are permitted to implement this capability. For a Multi-Function Device associated with an Upstream Port, each Function must report the same value for this bit. For Bridges and other Functions that do not implement this capability, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(Sticky) else R(Sticky) Note: This register field is sticky.
10	RO	0x0	**PCIE_CAP_NO_RO_EN_PR2PR_PAR**: No RO-enabled PR-PR Passing. If this bit is set, the routing element never carries out the passing permitted by Table 2-39 of PCI Express Base Specification entry A2b that is associated with the Relaxed Ordering Attribute field being Set. This bit applies only for Switches and RCs that support peer-to-peer traffic between Root Ports. This bit applies only to Posted Requests being forwarded through the Switch or RC and does not apply to traffic originating or terminating within the Switch or RC itself. All Ports on a Switch or RC must report the same value for this bit. For all other functions, this bit must be 0b.

Bit	R/W	Reset	Description
9	RO	0x0	**PCIE_CAP_128_CAS_CPL_SUPP**: 128-bit CAS Completer Supported. Applicable to Functions with Memory Space BARs as well as all Root Ports; must be 0b otherwise. This bit must be set to 1b if the Function supports this optional capability. For more details, see section 6.15 of PCI Express Base Specification.
8	RO	0x0	**PCIE_CAP_64_ATOMIC_CPL_SUPP**: 64-bit AtomicOp Completer Supported. Applicable to Functions with Memory Space BARs as well as all Root Ports; must be 0b otherwise. Includes FetchAdd, Swap, and CAS AtomicOps. This bit must be set to 1b if the Function supports this optional capability. For more details on additional RC requirements, see section 6.15.3.1 of PCI Express Base Specification.
7	RO	0x0	**PCIE_CAP_32_ATOMIC_CPL_SUPP**: 32-bit AtomicOp Completer Supported. Applicable to Functions with Memory Space BARs as well as all Root Ports; must be 0b otherwise. Includes FetchAdd, Swap, and CAS AtomicOps. This bit must be set to 1b if the Function supports this optional capability. For more details on additional RC requirements, see section 6.15.3.1 of PCI Express Base Specification.
6	RO	0x0	**PCIE_CAP_ATOMIC_ROUTING_SUPP**: AtomicOp Routing Supported. Applicable only to Switch Upstream Ports, Switch Downstream Ports, and Root Ports; must be 0b for other Function types. This bit must be set to 1b if the Port supports this optional capability. For more details, see section 6.15 of PCI Express Base Specification.
5	RO	0x0	**PCIE_CAP_ARI_FORWARD_SUPPORT**: ARI Forwarding Supported. Applicable only to Switch Downstream Ports and Root Ports; must be 0b for other Function types. This bit must be set to 1b if a Switch Downstream Port or Root Port supports this optional capability. For more details, see section 6.13 of PCI Express Base Specification.
4	RO	0x1	**PCIE_CAP_CPL_TIMEOUT_DISABLE_SUPPORT**: Completion Timeout Disable Supported. A value of 1b indicates support for the Completion Timeout Disable mechanism. The Completion Timeout Disable mechanism is required for Endpoints that issue Requests on their own behalf and PCI Express to PCI/PCI-X Bridges that take ownership of Requests issued on PCI Express. This mechanism is optional for Root Ports. For all other Functions this field is Reserved and the controller hardwires this bit to 0b.
3:0	RO	0xf	**PCIE_CAP_CPL_TIMEOUT_RANGE**: Completion Timeout Ranges Supported. This field indicates device Function support for the optional Completion Timeout programmability mechanism. This mechanism allows system software to modify the Completion Timeout value. This field is applicable only to Root Ports, Endpoints that issue Requests on their own behalf, and PCI Express to PCI/PCI-X Bridges that take ownership of Requests issued on PCI Express. For all other Functions this field is Reserved and must be hardwired to 0000b. Four time value ranges are defined: - Range A: 50 us to 10 ms - Range B: 10 ms to 250 ms - Range C: 250 ms to 4 s - Range D: 4 s to 64 s Bits are set according to the list below to show timeout value ranges supported. All encodings other than the defined encodings are reserved. It is strongly recommended that the Completion Timeout mechanism not expire in less than 10 ms.

PCIE_X<*j*>_EP_PF0_PCIE_CAP_DEVICE_CONTROL2_DEVICE_STATUS2_REG_0

where <*j*> = 4, 8.

Description:This register controls PCI Express device specific parameters and provides information about PCI Express device (function) specific parameters; in addition to the Device Control and Device Status Register.

PCIE_X4_EP_PF0_PCIE_CAP_DEVICE_CONTROL2_DEVICE_STATUS2_REG_0

PCIE_X8_EP_PF0_PCIE_CAP_DEVICE_CONTROL2_DEVICE_STATUS2_REG_0

Offset: 0x98
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,x0xx,xx00,0000)

Bit	R/W	Reset	Description
10	RW	0x0	**PCIE_CAP_LTR_EN**: LTR Mechanism Enable. When set to 1b, this bit enables Upstream Ports to send LTR messages and Downstream Ports to process LTR Messages. For a Multi-Function Device associated with an Upstream Port of a device that implements LTR, the bit in Function 0 is RW, and only Function 0 controls the component's Link behavior. In all other Functions of that device, this bit is RsvdP. Functions that do not implement the LTR mechanism are permitted to hardwire this bit to 0b. For Downstream Ports, this bit must be reset to the default value if the Port goes to DL_Down status. The write value is gated with the PCIE_CAP_LTR_SUPP field of DEVICE_CAPABILITIES2_REG. Note: RW for function #0 and RsdvP for all other functions. Note: The access attributes of this field are as follows: - Wire: if (pf=0 && DEVICE_CAPABILITIES2_REG.PCIE_CAP_LTR_SUPP) then R/W else R - Dbi: if (pf=0 && DEVICE_CAPABILITIES2_REG.PCIE_CAP_LTR_SUPP) then R/W else R
5	RO	0x0	**PCIE_CAP_ARI_FORWARD_SUPPORT_CS**: ARI Forwarding Enable. When set, the Downstream Port disables its traditional Device Number field being 0 enforcement when turning a Type 1 Configuration Request into a Type 0 Configuration Request, permitting access to Extended Functions in an ARI Device immediately below the Port. For more details, see Section 6.13 of PCI Express Base Specification.
4	RW	0x0	**PCIE_CAP_CPL_TIMEOUT_DISABLE**: Completion Timeout Disable. When set, this bit disables the Completion Timeout mechanism. This bit is required for all Functions that support the Completion Timeout Disable Capability. Functions that do not support this optional capability are permitted to hardwire this bit to 0b Software is permitted to set or clear this bit at any time. When set, the Completion Timeout detection mechanism is disabled. If there are outstanding Requests when the bit is cleared, it is permitted but not required for hardware to apply the completion timeout mechanism to the outstanding Requests. If this is done, it is permitted to base the start time for each Request on either the time this bit was cleared or the time each Request was issued.

Bit	R/W	Reset	Description
3:0	RW	0x0	**PCIE_CAP_CPL_TIMEOUT_VALUE**: Completion Timeout Value. In device Functions that support Completion Timeout programmability, this field allows system software to modify the Completion Timeout value. This field is applicable to Root Ports, Endpoints that issue Requests on their own behalf, and PCI Express to PCI/PCI-X Bridges that take ownership of Requests issued on PCI Express. For all other Functions this field is Reserved and controller hardwires it to 0000b. A Function that does not support this optional capability must hardwire this field to 0000b and is required to implement a timeout value in the range 50 us to 50 ms. Functions that support Completion Timeout programmability must support the values given below corresponding to the programmability ranges indicated in the Completion Timeout Ranges Supported field. All encodings other than the defined encodings are reserved. It is strongly recommended that the Completion Timeout mechanism not expire in less than 10 ms. Values available if Range A (50 us to 10 ms) programmability range is supported: Values available if Range B (10 ms to 250 ms) programmability range is supported: Values available if Range C (250 ms to 4 s) programmability range is supported: Values available if the Range D (4 s to 64 s) programmability range is supported: Software is permitted to change the value in this field at any time. For Requests already pending when the Completion Timeout Value is changed, hardware is permitted to use either the new or the old value for the outstanding Requests, and is permitted to base the start time for each Request either on when this value was changed or on when each request was issued. Note: The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

PCIE_X<*j*>_EP_PF0_PCIE_CAP_LINK_CAPABILITIES2_REG_0

where <*j*> = 4, 8.

Description:This register identifies PCI Express Link specific capabilities; in addition to the Link Capabilities Register.

PCIE_X4_EP_PF0_PCIE_CAP_LINK_CAPABILITIES2_REG_0

PCIE_X8_EP_PF0_PCIE_CAP_LINK_CAPABILITIES2_REG_0

Offset: 0x9c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0180001e (0bx000,0001,1000,0000,0000,0000,0001,1110)

Bit	R/W	Reset	Description
30:25	RO	0x0	**RSVDP_25**: Reserved for future use.

Bit	R/W	Reset	Description
24	RW	0x1	**PCIE_CAP_TWO_RETIMERS_PRE_DET_SUPPORT**: Two Retimers Presence Detect Supported. When set to 1b, this bit indicates that the associated Port supports detection and reporting of two Retimers presence. This bit must be set to 1b in a Port when the Supported Link Speeds Vector of the Link Capabilities 2 register indicates support for a Link speed of 16.0 GT/s or higher. It is permitted to be set to 1b regardless of the supported Link speeds if the Retimer Presence Detect Supported bit is also set to 1b. Multi-Function Devices associated with an Upstream Port must report the same value in this field for all Functions. Note: The access attributes of this field are as follows: - Wire: HWINIT - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky)
23	RW	0x1	**PCIE_CAP_RETIMER_PRE_DET_SUPPORT**: Retimer Presence Detect Supported. When set to 1b, this bit indicates that the associated Port supports detection and reporting of Retimer presence. This bit must be set to 1b in a Port when the Supported Link Speeds Vector of the Link Capabilities 2 register indicates support for a Link speed of 16.0 GT/s or higher. It is permitted to be set to 1b regardless of the supported Link speeds. Multi-Function Devices associated with an Upstream Port must report the same value in this field for all Functions. Note: The access attributes of this field are as follows: - Wire: HWINIT - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky)
22:9	RO	0x0	**RSVDP_9**: Reserved for future use.
8	RO	0x0	**PCIE_CAP_CROSS_LINK_SUPPORT**: Crosslink Supported. When set to 1b, this bit indicates that the associated Port supports crosslinks (for more details, see section 4.2.6.3.1 of PCI Express Base Specification). When set to 0b on a Port that supports Link speeds of 8.0 GT/s or higher, this bit indicates that the associated Port does not support crosslinks. When set to 0b on a Port that only supports Link speeds of 2.5 GT/s or 5.0 GT/s, this bit provides no information regarding the Port's level of crosslink support. It is recommended that this bit be Set in any Port that supports crosslinks even though doing so is only required for Ports that also support operating at 8.0 GT/s or higher Link speeds. Note: Software should use this bit when referencing fields whose definition depends on whether or not the Port supports crosslinks (for more details, see section 7.7.3.4 of PCI Express Base Specification). Multi-Function Devices associated with an Upstream Port must report the same value in this field for all Functions.
7:1	RO	0xf	**PCIE_CAP_SUPPORT_LINK_SPEED_VECTOR**: Supported Link Speeds Vector. This field indicates the supported Link speed(s) of the associated Port. For each bit, a value of 1b indicates that the corresponding Link speed is supported; otherwise, the Link speed is not supported. For more details, see section 8.2.1 of PCI Express Base Specification. Bit definitions within this field are: - Bit 0 2.5 GT/s - Bit 1 5.0 GT/s - Bit 2 8.0 GT/s - Bit 3 16.0 GT/s - Bit 4 32.0 GT/s - Bits 6:5 RsvdP Multi-Function Devices associated with an Upstream Port must report the same value in this field for all Functions. This field has a default of (PCIE_CAP_MAX_LINK_SPEED == 0101) ? 0011111 : (PCIE_CAP_MAX_LINK_SPEED == 0100) ? 0001111 : (PCIE_CAP_MAX_LINK_SPEED == 0011) ? 0000111 : (PCIE_CAP_MAX_LINK_SPEED == 0010) ? 0000011 : 0000001 where PCIE_CAP_MAX_LINK_SPEED is a field in the LINK_CAPABILITIES_REG register.
0	RO	0x0	**RSVDP_0**: Reserved for future use.

PCIE_X<*j*>_EP_PF0_PCIE_CAP_LINK_CONTROL2_LINK_STATUS2_REG_0

where <*j*> = 4, 8.

Description:This register controls and provides information about PCI Express Link specific parameters; in addition to the Link Control and Link Status Register.

PCIE_X4_EP_PF0_PCIE_CAP_LINK_CONTROL2_LINK_STATUS2_REG_0

PCIE_X8_EP_PF0_PCIE_CAP_LINK_CONTROL2_LINK_STATUS2_REG_0

Offset: 0xa0
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x01010004 (0bxxxx,0001,0000,0001,0000,0000,0000,0100)

Bit	R/W	Reset	Description
27:26	RO	0x0	**RSVDP_26**: Reserved for future use.
25:24	RO	0x1	**PCIE_CAP_CROSSLINK_RESOLUTION**: Crosslink Resolution. This field indicates the state of the Crosslink negotiation. It must be implemented if Crosslink Supported is Set and the Port supports 16.0 GT/s or higher data rate. It is permitted to be implemented in all other Ports. If Crosslink Supported is clear, the controller hardwires this field to 01b or 10b. Once a value of 01b or 10b is returned in this field, that value must continue to be returned while the Link is Up.
23	RO	0x0	**PCIE_CAP_TWO_RETIMERS_PRE_DET**: Two Retimers Presence Detected. When set to 1b, this bit indicates that two Retimers were present during the most recent Link negotiation. For more details, see section 4.2.6.3.5.1 of PCI Express Base Specification. This bit is required for Ports that have the Two Retimers Presence Detect Supported bit of the Link Capabilities 2 register set to 1b. Ports that have the Two Retimers Presence Detect Supported bit set to 0b are permitted to hardwire this bit to 0b. For Multi-Function Devices associated with an Upstream Port, this bit must be implemented in Function 0 and RsvdZ in all other Functions. Note: This register field is sticky.
22	RO	0x0	**PCIE_CAP_RETIMER_PRE_DET**: Retimer Presence Detected. When set to 1b, this bit indicates that a Retimer was present during the most recent Link negotiation. For more details, see section 4.2.6.3.5.1 of PCI Express Base Specification. This bit is required for Ports that have the Retimer Presence Detect Supported bit of the Link Capabilities 2 register set to 1b. For Ports that have the Retimer Presence Detect Supported bit set to 0b, the controller hardwires this bit to 0b. For Multi-Function Devices associated with an Upstream Port, this bit must be implemented in Function 0 and is RsvdZ in all other Functions. Note: This register field is sticky.

Bit	R/W	Reset	Description
21	RW	0x0	**PCIE_CAP_LINK_EQ_REQ**: Link Equalization Request 8.0 GT/s. This bit is set by hardware to request the 8.0 GT/s Link equalization process to be performed on the Link. For more details, see sections 4.2.3 and 4.2.6.4.2 of PCI Express Base Specification. For Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions. For components that only support speeds below 8.0 GT/s, the controller hardwires this bit to 0b.
20	RO	0x0	**PCIE_CAP_EQ_CPL_P3**: EEqualization 8.0 GT/s Phase 3 Successful. When set to 1b, this bit indicates that Phase 3 of the 8.0 GT/s Transmitter Equalization procedure has successfully completed. Details of the Transmitter Equalization process and when this bit needs to be set to 1b is provided in section 4.2.6.4.2 of PCI Express Base Specification. For Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions. For components that only support speeds below 8.0 GT/s, the controller hardwires this bit to 0b. Note: This register field is sticky.
19	RO	0x0	**PCIE_CAP_EQ_CPL_P2**: Equalization 8.0 GT/s Phase 2 Successful. When set to 1b, this bit indicates that Phase 2 of the 8.0 GT/s Transmitter Equalization procedure has successfully completed. Details of the Transmitter Equalization process and when this bit needs to be set to 1b is provided in section 4.2.6.4.2 of PCI Express Base Specification. For Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions. For components that only support speeds below 8.0 GT/s, the controller hardwires this bit to 0b. Note: This register field is sticky.
18	RO	0x0	**PCIE_CAP_EQ_CPL_P1**: Equalization 8.0 GT/s Phase 1 Successful. When set to 1b, this bit indicates that Phase 1 of the 8.0 GT/s Transmitter Equalization procedure has successfully completed. Details of the Transmitter Equalization process and when this bit needs to be set to 1b is provided in section 4.2.6.4.2 of PCI Express Base Specification. For Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions. For components that only support speeds below 8.0 GT/s, the controller hardwires this bit to 0b. Note: This register field is sticky.
17	RO	0x0	**PCIE_CAP_EQ_CPL**: Equalization 8.0 GT/s Complete. When set to 1b, this bit indicates that the Transmitter Equalization procedure at the 8.0 GT/s data rate has completed. Details of the Transmitter Equalization process and when this bit needs to be set to 1b is provided in section 4.2.6.4.2 of PCI Express Base Specification. For Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions. For components that only support speeds below 8.0 GT/s, the controller hardwires this bit to 0b. Note: This register field is sticky.
16	RO	0x1	**PCIE_CAP_CURR_DEEMPHASIS**: Current De-emphasis Level. When the Link is operating at 5.0 GT/s speed, this bit reflects the level of de-emphasis. The value in this bit is undefined when the Link is not operating at 5.0 GT/s speed. Components that support only the 2.5 GT/s speed are permitted to hardwire this bit to 0b. For components that support speeds greater than 2.5 GT/s, Multi-Function Devices associated with an Upstream Port must report the same value in this field for all Functions of the Port. In M-PCIe mode this register is always 0x0. In C-PCIe mode, its contents are derived by sampling the PIPE.

Bit	R/W	Reset	Description
15:12	RW	0x0	**PCIE_CAP_COMPLIANCE_PRESET**: Compliance Preset/De-emphasis. - For 8.0 GT/s and higher Data Rate: This field sets the Transmitter Preset in Polling.Compliance state if the entry occurred due to the Enter Compliance bit being 1b. The encodings are defined in section 4.2.3.2 of PCI Express Base Specification . Results are undefined if a reserved preset encoding is used when entering Polling.Compliance in this way. - For 5.0 GT/s Data Rate: This field sets the de-emphasis level in Polling.Compliance state if the entry occurred due to the Enter Compliance bit being 1b. - When the Link is operating at 2.5 GT/s, the setting of this field has no effect. Components that support only 2.5 GT/s speed are permitted to hardwire this field to 0000b. - For a Multi-Function Device associated with an Upstream Port, the field in Function 0 is of type RWS, and only Function 0 controls the component's Link behavior. In all other Functions of that device, this field is of type RsvdP. - This field is intended for debug and compliance testing purposes. System firmware and software is allowed to modify this field only during debug or compliance testing. In all other cases, the system must ensure that this field is set to the default value. Note: The access attributes of this field are as follows: - Wire: R/W (sticky) - Dbi: R/W (sticky) Note: This register field is sticky.
11	RW	0x0	**PCIE_CAP_COMPLIANCE_SOS**: Compliance SOS. When set to 1b, the LTSSM is required to send SKP Ordered Sets between sequences when sending the Compliance Pattern or Modified Compliance Pattern. For a Multi-Function Device associated with an Upstream Port, the bit in Function 0 is of type RWS, and only Function 0 controls the component's Link behavior. In all other Functions of that device, this bit is of type RsvdP. This bit is applicable when the Link is operating at 2.5 GT/s or 5.0 GT/s data rates only. For components that support only the 2.5 GT/s speed, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Wire: R/W (sticky) - Dbi: R/W (sticky) Note: This register field is sticky.
10	RW	0x0	**PCIE_CAP_ENTER_MODIFIED_COMPLIANCE**: Enter Modified Compliance. When this bit is set to 1b, the device transmits Modified Compliance Pattern if the LTSSM enters Polling.Compliance substate. Components that support only the 2.5 GT/s speed are permitted to hardwire this bit to 0b. For a Multi-Function Device associated with an Upstream Port, the bit in Function 0 is of type RWS, and only Function 0 controls the component's Link behavior. In all other Functions of that device, this bit is of type RsvdP. This bit is intended for debug, compliance testing purposes only. System firmware and software is allowed to modify this bit only during debug or compliance testing. In all other cases, the system must ensure that this bit is set to the default value. Note: The access attributes of this field are as follows: - Wire: R/W (sticky) - Dbi: R/W (sticky) Note: This register field is sticky.
9:7	RW	0x0	**PCIE_CAP_TX_MARGIN**: Transmit Margin, This field controls the value of the non-deemphasized voltage level at the Transmitter pins. This field is reset to 000b on entry to the LTSSM Polling.Configuration substate (see Chapter 4 of PCI Express Base Specification for details of how the Transmitter voltage level is determined in various states). - 001b-111b: As defined in Section 8.3.4 not all encodings are required to be implemented. For a Multi-Function Device associated with an Upstream Port, the field in Function 0 is of type RWS, and only Function 0 controls the component's Link behavior. In all other Functions of that device, this field is of type RsvdP. For components that support only the 2.5 GT/s speed, the controller hardwires this bit to 000b. This field is intended for debug, compliance testing purposes only. System firmware and software is allowed to modify this field only during debug or compliance testing. In all other cases, the system must ensure that this field is set to the default value. Note: This register field is sticky.

Bit	R/W	Reset	Description
6	RO	0x0	**PCIE_CAP_SEL_DEEMPHASIS**: Selectable De-emphasis. When the Link is operating at 5.0 GT/s speed, this bit is used to control the transmit de-emphasis of the link in specific situations. For more details, see section 4.2.6 of PCI Express Base Specification. When the Link is not operating at 5.0 GT/s speed, the setting of this bit has no effect. Components that support only the 2.5 GT/s speed are permitted to hardwire this bit to 0b. This bit is not applicable and Reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches. Note: This register field is sticky.
5	RW	0x0	**PCIE_CAP_HW_AUTO_SPEED_DISABLE**: Hardware Autonomous Speed Disable. When set, this bit disables hardware from changing the Link speed for device-specific reasons other than attempting to correct unreliable Link operation by reducing Link speed. Initial transition to the highest supported common link speed is not blocked by this bit. For a Multi-Function Device associated with an Upstream Port, the bit in Function 0 is of type RWS, and only Function 0 controls the component's Link behavior. In all other Functions of that device, this bit is of type RsvdP. Functions that do not implement the associated mechanism are permitted to hardwire this bit to 0b. Note: The access attributes of this field are as follows: - Wire: R/W (sticky) - Dbi: R/W (sticky) Note: This register field is sticky.
4	RW	0x0	**PCIE_CAP_ENTER_COMPLIANCE**: Enter Compliance. Software is permitted to force a Link to enter Compliance mode (at the speed indicated in the Target Link Speed field and the de-emphasis/preset level indicated by the Compliance Preset/De-emphasis field) by setting this bit to 1b in both components on a Link and then initiating a hot reset on the Link. Default value of this bit following Fundamental Reset is 0b. For a Multi-Function Device associated with an Upstream Port, the bit in Function 0 is of type RWS, and only Function 0 controls the component's Link behavior. In all other Functions of that device, this bit is of type RsvdP. Components that support only the 2.5 GT/s speed are permitted to hardwire this bit to 0b. This bit is intended for debug, compliance testing purposes only. System firmware and software is allowed to modify this bit only during debug or compliance testing. In all other cases, the system must ensure that this bit is set to the default value. Note: This register field is sticky.

Bit	R/W	Reset	Description
3:0	RW	0x4	**PCIE_CAP_TARGET_LINK_SPEED**: Target Link Speed. For Downstream Ports, this field sets an upper limit on Link operational speed by restricting the values advertised by the Upstream component in its training sequences. The encoded value specifies a bit location in the Supported Link Speeds Vector (in the Link Capabilities 2 register) that corresponds to the desired target Link speed. All encodings other than the defined encodings are reserved. If a value is written to this field that does not correspond to a supported speed (as indicated by the Supported Link Speeds Vector), the result is undefined. If either of the Enter Compliance or Enter Modified Compliance bits are implemented, then this field must also be implemented. The default value of this field is the highest Link speed supported by the component (as reported in the Max Link Speed field of the Link Capabilities register) unless the corresponding platform/form factor requires a different default value. For both Upstream and Downstream Ports, this field is used to set the target compliance mode speed when software is using the Enter Compliance bit to force a Link into compliance mode. For Upstream Ports, if the Enter Compliance bit is Clear, this field is permitted to have no effect. For a Multi-Function Device associated with an Upstream Port, the field in Function 0 is of type RWS, and only Function 0 controls the component's Link behavior. In all other Functions of that device, this field is of type RsvdP. Components that support only the 2.5 GT/s speed are permitted to hardwire this field to 0000b. In M-PCIe mode, the contents of this field are derived from other registers. Note: The access attributes of this field are as follows: - Wire: R/W (sticky) - Dbi: R/W (sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_MSIX_CAP_PCI_MSIX_CAP_ID_NEXT_CTRL_REG_0

where <*j*> = 4, 8.

Description:This Register holds MSI-X Capability ID, Next Capability pointer. It also controls the MSI-X behaviour.

PCIE_X4_EP_PF0_MSIX_CAP_PCI_MSIX_CAP_ID_NEXT_CTRL_REG_0

PCIE_X8_EP_PF0_MSIX_CAP_PCI_MSIX_CAP_ID_NEXT_CTRL_REG_0

Offset: 0xb0
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00070011 (0b0000,0000,0000,0111,0000,0000,0001,0001)

Bit	R/W	Reset	Description
31	RW	0x0	**PCI_MSIX_ENABLE**: MSI-X Enable. If Set and the MSI Enable bit in the MSI Message Control Register for MSI is Clear, the Function is permitted to use MSI-X to request service and is prohibited from using INTx interrupts (if implemented). System configuration software Sets this bit to enable MSI-X. A device driver is prohibited from writing this bit to mask a Function's service request. If Clear, the Function is prohibited from using MSI-X to request service.
30	RW	0x0	**PCI_MSIX_FUNCTION_MASK**: Function Mask. If Set, all of the vectors associated with the Function are masked, regardless of their per-vector Mask bit values. If Clear, each vector's Mask bit determines whether the vector is masked or not. Setting or Clearing the MSI-X Function Mask bit has no effect on the value of the per-vector Mask bits. Note: The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
29:27	RO	0x0	**RSVDP_27**: Reserved for future use.
26:16	RW	0x7	**PCI_MSIX_TABLE_SIZE**: MSI-X Table Size. System software reads this field to determine the MSI-X Table Size N, which is encoded as N-1. For example, a returned value of 000 0000 0011b indicates a table size of 4. SRIOV Note: All VFs in a single PF have the same value for "MSI-X Table Size" (PCI_MSIX_TABLE_SIZE field in SHADOW_PCI_MSIX_CAP_ID_NEXT_CTRL_REG). To write this common value, you must perform a DBI_CS2 write (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) while accessing the PCI_MSIX_TABLE_SIZE field in the PF PCI_MSIX_CAP_ID_NEXT_CTRL_REG register. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
15:8	RW	0x0	**PCI_MSIX_CAP_NEXT_OFFSET**: MSI-X Next Capability Pointer. This field contains the offset to the next PCI Capability structure or 00h if no other items exist in the linked list of Capabilities. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
7:0	RO	0x11	**PCI_MSIX_CAP_ID**: MSI-X Capability ID. This field indicates the MSI-X Capability structure. This field must return a Capability ID of 11h indicating that this is an MSI-X Capability structure.

PCIE_X<*j*>_EP_PF0_MSIX_CAP_MSIX_TABLE_OFFSET_REG_0

where <*j*> = 4, 8.

Description:This register provides Table BIR and MSI-x Table offset select.

PCIE_X4_EP_PF0_MSIX_CAP_MSIX_TABLE_OFFSET_REG_0

PCIE_X8_EP_PF0_MSIX_CAP_MSIX_TABLE_OFFSET_REG_0

Offset: 0xb4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000002 (0b0000,0000,0000,0000,0000,0000,0000,0010)

Bit	Reset	Description
31:3	0x0	**PCI_MSIX_TABLE_OFFSET**: MSI-X Table Offset. Used as an offset from the address contained by one of the Function's Base Address Registers to point to the base of the MSI-X Table. The lower 3 Table BIR bits are masked off (set to zero) by software to form a 32-bit QWORD-aligned offset. SRIOV Note: All VFs in a single PF have the same value for "MSI-X Table Offset" (PCI_MSIX_TABLE_OFFSET field in SHADOW_MSIX_TABLE_OFFSET_REG). To write this common value, you must perform a DBI_CS2 write (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) while accessing the PCI_MSIX_TABLE_OFFSET field in the PF MSIX_TABLE_OFFSET_REG register. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
2:0	0x2	**PCI_MSIX_BIR**: MSI-X Table BAR Indicator Register Field. This field indicates which one of a Function's Base Address Registers , located beginning at 10h in Configuration Space, or entry in the Enhanced Allocation capability with a matching BEI , is used to map the Function's MSI-X Table into Memory Space. All encodings other than the defined encodings are reserved. For a 64-bit Base Address Register , the Table BIR indicates the lower DWORD. For Functions with Type 1 Configuration Space headers, BIR values 2 through 5 are also Reserved. SRIOV Note: All VFs in a single PF have the same value for "MSI-X Table BAR Indicator Register" (PCI_MSIX_BIR field in SHADOW_MSIX_TABLE_OFFSET_REG). To write this common value, you must perform a DBI_CS2 write (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) while accessing the PCI_MSIX_BIR field in the PF MSIX_TABLE_OFFSET_REG register. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_MSIX_CAP_MSIX_PBA_OFFSET_REG_0

where <*j*> = 4, 8.

Description:This register provides PBA Offset and PBA BIR value.

PCIE_X4_EP_PF0_MSIX_CAP_MSIX_PBA_OFFSET_REG_0

PCIE_X8_EP_PF0_MSIX_CAP_MSIX_PBA_OFFSET_REG_0

Offset: 0xb8
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00010002 (0b0000,0000,0000,0001,0000,0000,0000,0010)

Bit	Reset	Description
31:3	0x2000	**PCI_MSIX_PBA_OFFSET**: MSI-X PBA Offset. Used as an offset from the address contained by one of the Function's Base Address Registers to point to the base of the MSI-X PBA. The lower 3 PBA BIR bits are masked off (set to zero) by software to form a 32-bit QWORD-aligned offset. SRIOV Note: All VFs in a single PF have the same value for "MSI-X PBA Offset" (PCI_MSIX_PBA_OFFSET field in SHADOW_MSIX_PBA_OFFSET_REG). To write this common value, you must perform a DBI_CS2 write (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) while accessing the PCI_MSIX_PBA_OFFSET field in the PF MSIX_PBA_OFFSET_REG register. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
2:0	0x2	**PCI_MSIX_PBA_BIR**: MSI-X PBA BIR. This field indicates which one of a Function's Base Address Registers , located beginning at 10h in Configuration Space, or entry in the Enhanced Allocation capability with a matching BEI, is used to map the Function's MSI-X PBA into Memory Space. The PBA BIR value definitions are identical to those for the Table BIR . SRIOV Note: All VFs in a single PF have the same value for "MSI-X PBA BIR" (PCI_MSIX_PBA_BIR field in SHADOW_MSIX_PBA_OFFSET_REG). To write this common value, you must perform a DBI_CS2 write (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) while accessing the PCI_MSIX_PBA_BIR field in the PF MSIX_PBA_OFFSET_REG register. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_AER_CAP_AER_EXT_CAP_HDR_OFF_0

where <*j*> = 4, 8.

Description:Advanced Error Reporting Extended Capability Header provides information about Capability ID, Version, and next offset.

PCIE_X4_EP_PF0_AER_CAP_AER_EXT_CAP_HDR_OFF_0

PCIE_X8_EP_PF0_AER_CAP_AER_EXT_CAP_HDR_OFF_0

Offset: 0x100
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x14820001 (0b0001,0100,1000,0010,0000,0000,0000,0001)

Bit	Reset	Description
31:20	0x148	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x2	**CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. This field must be 2h if the End-End TLP Prefix Supported bit is set and must be 1h or 2h otherwise. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x1	**CAP_ID**: AER Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. The Extended Capability ID for the Advanced Error Reporting Capability is 0001h. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_AER_CAP_UNCORR_ERR_STATUS_OFF_0

where <*j*> = 4, 8.

Description:The Uncorrectable Error Status Register (Offset 04h) indicates error detection status of individual errors on a PCI Express device Function. An individual error status bit that is Set indicates that a particular error was detected; software may clear an error status by writing a 1b to the respective bit. Register bits not implemented by the Function are hardwired to 0b.

PCIE_X4_EP_PF0_AER_CAP_UNCORR_ERR_STATUS_OFF_0

PCIE_X8_EP_PF0_AER_CAP_UNCORR_ERR_STATUS_OFF_0

Offset: 0x104
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0xxx,00x0,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:27	RO	0x0	**RSVDP_27**: Reserved for future use.

Bit	R/W	Reset	Description
23	RO	0x0	**RSVDP_23**: Reserved for future use.
22	RW	0x0	**INTERNAL_ERR_STATUS**: Uncorrectable Internal Error Status. This field gives status of the Uncorrectable Internal Error. The controller sets this bit when your application asserts app_err_bus[9]. It does not set this bit when it detects internal uncorrectable internal errors such as parity and ECC failures. You should use the outputs from these errors to drive the app_err_bus[9] input. For more details, see the "Data Integrity (Wire, Datapath, and RAM Protection)" section in the Databook.
20	RW	0x0	**UNSUPPORTED_REQ_ERR_STATUS**: Unsupported Request Error Status. This field represents status of Unsupported Request Error.
19	RW	0x0	**ECRC_ERR_STATUS**: ECRC Error Status. This field represents status of ECRC Error. Note:If CX_ECRC_ENABLE=0 the register field always reads 0.
18	RW	0x0	**MALF_TLP_ERR_STATUS**: Malformed TLP Status. This field represents status of Malformed TLP.
17	RW	0x0	**REC_OVERFLOW_ERR_STATUS**: Receiver Overflow Status. Status bit for Receiver Overflow.
16	RW	0x0	**UNEXP_CMPLT_ERR_STATUS**: Unexpected Completion Status. Status bit for Unexpected Completion.
15	RW	0x0	**CMPLT_ABORT_ERR_STATUS**: Completer Abort Status. Status bit for Completer Abort.
14	RW	0x0	**CMPLT_TIMEOUT_ERR_STATUS**: Completion Timeout Status. Status for Completion Timeout.
13	RW	0x0	**FC_PROTOCOL_ERR_STATUS**: Flow Control Protocol Error Status. Status bit for Flow Control Protocol Error.
12	RW	0x0	**POIS_TLP_ERR_STATUS**: Poisoned TLP Status. Status bit for Poisoned TLP.
11:6	RO	0x0	**RSVDP_6**: Reserved for future use.
5	RW	0x0	**SURPRISE_DOWN_ERR_STATUS**: Surprise Down Error Status (Optional). Status bit for Surprise Down Error.
4	RW	0x0	**DL_PROTOCOL_ERR_STATUS**: Data Link Protocol Error Status. Status bit for Data Link Protocol Error.
3:0	RO	0x0	**RSVDP_0**: Reserved for future use.

PCIE_X<*j*>_EP_PF0_AER_CAP_UNCORR_ERR_MASK_OFF_0

where <*j*> = 4, 8.

Description:The Uncorrectable Error Mask Register controls reporting of individual errors by the device Function to the PCI Express Root Complex through a PCI Express error Message. A masked error (respective bit Set in the mask register) is not recorded or reported in the Header Log, TLP Prefix Log, or First Error Pointer, and is not reported to the PCI Express Root Complex by this Function. There is a mask bit per error bit of the Uncorrectable Error Status register. Register fields for bits not implemented by the Function are hardwired to 0b.

PCIE_X4_EP_PF0_AER_CAP_UNCORR_ERR_MASK_OFF_0

PCIE_X8_EP_PF0_AER_CAP_UNCORR_ERR_MASK_OFF_0

Offset: 0x108
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00400000 (0b0000,0x00,01x0,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:27	RO	0x0	**RSVDP_27**: Reserved for future use.
25	RO	0x0	**TLP_PRFX_BLOCKED_ERR_MASK**: TLP Prefix Blocked Error Mask. Mask bit for TLP Prefix Blocked Error. Note: Not supported. Note: This register field is sticky.
24	RO	0x0	**ATOMIC_EGRESS_BLOCKED_ERR_MASK**: AtomicOp Egress Block Mask (Optional). Mask bit for AtomicOp Egress Block Error. Note: This register field is sticky.
23	RO	0x0	**RSVDP_23**: Reserved for future use.
22	RW	0x1	**INTERNAL_ERR_MASK**: Uncorrectable Internal Error Mask (Optional). Mask bit for Uncorrectable Internal Error. Note: This register field is sticky.
20	RW	0x0	**UNSUPPORTED_REQ_ERR_MASK**: Unsupported Request Error Mask. Mask bit for Unsupported Request Error. Note: This register field is sticky.
19	RW	0x0	**ECRC_ERR_MASK**: ECRC Error Mask (Optional). Mask bit for ECRC Error. Note: The access attributes of this field are as follows: - Wire: R/W (sticky) - Dbi: R/W (sticky) Note: This register field is sticky.
18	RW	0x0	**MALF_TLP_ERR_MASK**: Malformed TLP Mask. Mask bit for Malformed TLP. Note: This register field is sticky.
17	RW	0x0	**REC_OVERFLOW_ERR_MASK**: Receiver Overflow Mask (Optional). This field represents Receiver Overflow Mask. Note: This register field is sticky.

Bit	R/W	Reset	Description
16	RW	0x0	**UNEXP_CMPLT_ERR_MASK**: Unexpected Completion Mask. Mask bit for Unexpected Completion Error. Note: This register field is sticky.
15	RW	0x0	**CMPLT_ABORT_ERR_MASK**: Completer Abort Error Mask (Optional). Mask bit for Completer Abort Error. Note: This register field is sticky.
14	RW	0x0	**CMPLT_TIMEOUT_ERR_MASK**: Completion Timeout Error Mask. Mask bit for Completion Timeout Error. Note: This register field is sticky.
13	RW	0x0	**FC_PROTOCOL_ERR_MASK**: Flow Control Protocol Error Mask. Mask bit for Flow Control Protocol Error. Note: This register field is sticky.
12	RW	0x0	**POIS_TLP_ERR_MASK**: Poisoned TLP Error Mask. Mask bit for Poisoned TLP Error. Note: This register field is sticky.
11:6	RO	0x0	**RSVDP_6**: Reserved for future use.
5	RO	0x0	**SURPRISE_DOWN_ERR_MASK**: Surprise Down Error Mask. Mask bit for Surprise Down Error. Note: This register field is sticky.
4	RW	0x0	**DL_PROTOCOL_ERR_MASK**: Data Link Protocol Error Mask. This field informs whether Data Link Protocol Error is masked or not. Note: This register field is sticky.
3:0	RO	0x0	**RSVDP_0**: Reserved for future use.

PCIE_X<*j*>_EP_PF0_AER_CAP_UNCORR_ERR_SEV_OFF_0

where <*j*> = 4, 8.

Description:The Uncorrectable Error Severity Register controls whether an individual error is reported as a Non-fatal or Fatal error. An error is reported as fatal when the corresponding error bit in the severity register is Set. If the bit is Clear, the corresponding error is considered non-fatal.

PCIE_X4_EP_PF0_AER_CAP_UNCORR_ERR_SEV_OFF_0

PCIE_X8_EP_PF0_AER_CAP_UNCORR_ERR_SEV_OFF_0

Offset: 0x10c
Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00462030 (0b0000,0x00,01x0,0110,0010,0000,0011,0000)

Bit	R/W	Reset	Description
31:27	RO	0x0	**RSVDP_27**: Reserved for future use.
25	RO	0x0	**TLP_PRFX_BLOCKED_ERR_SEVERITY**: TLP Prefix Blocked Error Severity (Optional). Severity bit for TLP Prefix Blocked Error. Note: Not supported. Note: This register field is sticky.
24	RO	0x0	**ATOMIC_EGRESS_BLOCKED_ERR_SEVERITY**: AtomicOp Egress Blocked Severity (Optional). Severity bit for AtomicOp Egress Blocked Error. Note: This register field is sticky.
23	RO	0x0	**RSVDP_23**: Reserved for future use.
22	RW	0x1	**INTERNAL_ERR_SEVERITY**: Uncorrectable Internal Error Severity (Optional). Severity bit for Uncorrectable Internal Error. Note: This register field is sticky.
20	RW	0x0	**UNSUPPORTED_REQ_ERR_SEVERITY**: Unsupported Request Error Severity. Severity bit for Unsupported Request Error. Note: This register field is sticky.
19	RW	0x0	**ECRC_ERR_SEVERITY**: ECRC Error Severity (Optional). Severity bit for ECRC Error. Note: The access attributes of this field are as follows: - Wire: R/W (sticky) - Dbi: R/W (sticky) Note: This register field is sticky.
18	RW	0x1	**MALF_TLP_ERR_SEVERITY**: Malformed TLP Severity. Severity bit for Malformed TLP. Note: This register field is sticky.
17	RW	0x1	**REC_OVERFLOW_ERR_SEVERITY**: Receiver Overflow Error Severity (Optional). Severity bit for Receiver Overflow Error. Note: This register field is sticky.
16	RW	0x0	**UNEXP_CMPLT_ERR_SEVERITY**: Unexpected Completion Error Severity. Severity bit for Unexpected Completion Error. Note: This register field is sticky.
15	RW	0x0	**CMPLT_ABORT_ERR_SEVERITY**: Completer Abort Error Severity (Optional). Severity bit for Completer Abort Error. Note: This register field is sticky.
14	RW	0x0	**CMPLT_TIMEOUT_ERR_SEVERITY**: Completion Timeout Error Severity. Severity bit for Completion Timeout Error. Note: This register field is sticky.
13	RW	0x1	**FC_PROTOCOL_ERR_SEVERITY**: Flow Control Protocol Error Severity (Optional). Severity bit for Flow Control Protocol Error. Note: This register field is sticky.
12	RW	0x0	**POIS_TLP_ERR_SEVERITY**: Poisoned TLP Severity. Severity bit for Poisoned TLP. Note: This register field is sticky.
11:6	RO	0x0	**RSVDP_6**: Reserved for future use.

Bit	R/W	Reset	Description
5	RO	0x1	**SURPRISE_DOWN_ERR_SVRITY**: Surprise Down Error Severity (Optional). Severity bit for Surprise Down Error. Note: This register field is sticky.
4	RW	0x1	**DL_PROTOCOL_ERR_SEVERITY**: Data Link Protocol Error Severity. Severity bit for Data Link Protocol Error. Note: This register field is sticky.
3:0	RO	0x0	**RSVDP_0**: Reserved for future use.

PCIE_X<*j*>_EP_PF0_AER_CAP_CORR_ERR_STATUS_OFF_0

where <*j*> = 4, 8.

Description:The Correctable Error Status register reports error status of individual correctable error sources on a PCI Express device Function. When an individual error status bit is Set, it indicates that a particular error occurred; software may clear an error status by writing a 1b to the respective bit. Register bits not implemented by the Function are hardwired to 0b by the controller.

PCIE_X4_EP_PF0_AER_CAP_CORR_ERR_STATUS_OFF_0

PCIE_X8_EP_PF0_AER_CAP_CORR_ERR_STATUS_OFF_0

Offset: 0x110
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:16	RO	0x0	**RSVDP_16**: Reserved for future use.
15	RW	0x0	**HEADER_LOG_OVERFLOW_STATUS**: Header Log Overflow Error Status (Optional). This field provides status of Header Log Overflow Error.
14	RW	0x0	**CORRECTED_INT_ERR_STATUS**: Corrected Internal Error Status (Optional). This field provides status of Corrected Internal Error.
13	RW	0x0	**ADVISORY_NON_FATAL_ERR_STATUS**: Advisory Non-Fatal Error Status. Status bit for Advisory Non-Fatal Error.
12	RW	0x0	**RPL_TIMER_TIMEOUT_STATUS**: Replay Timer Timeout Status. Status bit for Replay Timer Timeout.

Bit	R/W	Reset	Description
11:9	RO	0x0	**RSVDP_9**: Reserved for future use.
8	RW	0x0	**REPLAY_NO_ROLEOVER_STATUS**: REPLAY_NUM Rollover Status. Status bit for REPLAY_NUM Rollover.
7	RW	0x0	**BAD_DLLP_STATUS**: Bad DLLP Status. Status bit for Bad DLLP.
6	RW	0x0	**BAD_TLP_STATUS**: Bad TLP Status. Status bit for Bad TLP.
5:1	RO	0x0	**RSVDP_1**: Reserved for future use.
0	RW	0x0	**RX_ERR_STATUS**: Receiver Error Status (Optional). This field provides status of Receiver Error.

PCIE_X<*j*>_EP_PF0_AER_CAP_CORR_ERR_MASK_OFF_0

where <*j*> = 4, 8.

Description:The Correctable Error Mask Register controls reporting of individual correctable errors by this Function to the PCI Express Root Complex through a PCI Express error Message. A masked error (respective bit Set in the mask register) is not reported to the PCI Express Root Complex by this Function. There is a mask bit per error bit in the Correctable Error Status register. Register fields for bits not implemented by the Function are hardwired to 0b by the controller.

PCIE_X4_EP_PF0_AER_CAP_CORR_ERR_MASK_OFF_0

PCIE_X8_EP_PF0_AER_CAP_CORR_ERR_MASK_OFF_0

Offset: 0x114
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000e000 (0b0000,0000,0000,0000,1110,0000,0000,0000)

Bit	R/W	Reset	Description
31:16	RO	0x0	**RSVDP_16**: Reserved for future use.
15	RW	0x1	**HEADER_LOG_OVERFLOW_MASK**: Header Log Overflow Error Mask (Optional). Masking bit for Header Log Overflow Error. Note: This register field is sticky.

Bit	R/W	Reset	Description
14	RW	0x1	**CORRECTED_INT_ERR_MASK**: Corrected Internal Error Mask (Optional). Masking bit for Corrected Internal Error Mask. Note: This register field is sticky.
13	RW	0x1	**ADVISORY_NON_FATAL_ERR_MASK**: Advisory Non-Fatal Error Mask. Masking bit for Advisory Non-Fatal Error. Note: This register field is sticky.
12	RW	0x0	**RPL_TIMER_TIMEOUT_MASK**: Replay Timer Timeout Mask. Masking bit for Replay Timer Timeout. Note: This register field is sticky.
11:9	RO	0x0	**RSVDP_9**: Reserved for future use.
8	RW	0x0	**REPLAY_NO_ROLEOVER_MASK**: REPLAY_NUM Rollover Mask. Masking bit for REPLAY_NUM Rollover. Note: This register field is sticky.
7	RW	0x0	**BAD_DLLP_MASK**: Bad DLLP Mask. Masking bit for Bad DLLP. Note: This register field is sticky.
6	RW	0x0	**BAD_TLP_MASK**: Bad TLP Mask. Masking bit for Bad TLP. Note: This register field is sticky.
5:1	RO	0x0	**RSVDP_1**: Reserved for future use.
0	RW	0x0	**RX_ERR_MASK**: Receiver Error Mask (Optional). Masking bit for Receiver Error. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_AER_CAP_ADV_ERR_CAP_CTRL_OFF_0

where <*j*> = 4, 8.

Description:Advanced Error Capabilities and Control Register provides information whether the individual capability is supported or not. If the capability is supported then it is enabled or not.

PCIE_X4_EP_PF0_AER_CAP_ADV_ERR_CAP_CTRL_OFF_0

PCIE_X8_EP_PF0_AER_CAP_ADV_ERR_CAP_CTRL_OFF_0

Offset: 0x118
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000002a0 (0b0000,0000,0000,0000,0000,x010,1010,0000)

Bit	R/W	Reset	Description
31:13	RO	0x0	**RSVDP_13**: Reserved for future use.
12	RO	0x0	**CTO_PRFX_HDR_LOG_CAP**: TLP Prefix Log Present. If Set, this bit indicates that the Function records the prefix/header of Request TLPs that experience a Completion Timeout error.
10	RW	0x0	**MULTIPLE_HEADER_EN**: Multiple Header Recording Enable. When Set, this bit enables the Function to record more than one error header. Functions that do not implement the associated mechanism are permitted to hardwire this bit to 0b. Default value of this bit is 0b. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: R (sticky) Note: This register field is sticky.
9	RO	0x1	**MULTIPLE_HEADER_CAP**: Multiple Header Recording Capable. If Set, this bit indicates that the Function is capable of recording more than one error header. Note: This register field is sticky.
8	RW	0x0	**ECRC_CHECK_EN**: ECRC Check Enable. When Set, ECRC checking is enabled. Functions that do not implement the associated mechanism are permitted to hardwire this bit to 0b. Default value of this bit is 0b. Note: This register field is sticky.
7	RO	0x1	**ECRC_CHECK_CAP**: ECRC Check Capable. If Set, this bit indicates that the Function is capable of checking ECRC. Note: This register field is sticky.
6	RW	0x0	**ECRC_GEN_EN**: ECRC Generation Enable. When Set, ECRC generation is enabled. Functions that do not implement the associated mechanism are permitted to hardwire this bit to 0b. Default value of this bit is 0b. Note: This register field is sticky.
5	RO	0x1	**ECRC_GEN_CAP**: ECRC Generation Capable. If Set, this bit indicates that the Function is capable of generating ECRC. Note: This register field is sticky.
4:0	RO	0x0	**FIRST_ERR_POINTER**: First Error Pointer. The First Error Pointer is a field that identifies the bit position of the first error reported in the Uncorrectable Error Status register. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_AER_CAP_HDR_LOG_0_OFF_0

where <*j*> = 4, 8.

Description:The Header Log Register 0 contains the header for the TLP corresponding to a detected error; The header is captured such that, when read using DW accesses, the fields of the header are laid out in the same way the headers are presented in the specification. Therefore, byte 0 of the header is located in byte 3 of the Header Log Register 0, byte 1 of the header is in byte 2 of the Header Log Register 0, and so forth. For 12-byte headers, only bytes 0 through 11 of the

Header Log Register are used and values in bytes 12 through 15 are undefined. In certain cases where a Malformed TLP is reported, the Header Log Register may contain TLP Prefix information.

PCIE_X4_EP_PF0_AER_CAP_HDR_LOG_0_OFF_0

PCIE_X8_EP_PF0_AER_CAP_HDR_LOG_0_OFF_0

Offset: 0x11c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**FIRST_DWORD_FOURTH_BYTE**: Byte 3 of Header log register of First 32-bit Data Word. This field represents fourth byte of First DW of Header. Note: This register field is sticky.
23:16	0x0	**FIRST_DWORD_THIRD_BYTE**: Byte 2 of Header log register of First 32-bit Data Word. This field represents third byte of First DW of Header. Note: This register field is sticky.
15:8	0x0	**FIRST_DWORD_SECOND_BYTE**: Byte 1 of Header log register of First 32-bit Data Word. This field represents second byte of First DW of Header. Note: This register field is sticky.
7:0	0x0	**FIRST_DWORD_FIRST_BYTE**: Byte 0 of Header log register of First 32-bit Data Word. This field represents first byte of First DW of Header. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_AER_CAP_HDR_LOG_1_OFF_0

where <*j*> = 4, 8.

Description:The Header Log Register 1 contains the header for the TLP corresponding to a detected error; The header is captured such that, when read using DW accesses, the fields of the header are laid out in the same way the headers are presented in the specification. Therefore, byte 0 of the header is located in byte 3 of the Header Log Register 1, byte 1 of the header is in byte 2 of the Header Log Register 1 and so forth. For 12-byte headers, only bytes 0 through 11 of the Header Log Register are used and values in bytes 12 through 15 are undefined. In certain cases where a Malformed TLP is reported, the Header Log Register may contain TLP Prefix information.

PCIE_X4_EP_PF0_AER_CAP_HDR_LOG_1_OFF_0

PCIE_X8_EP_PF0_AER_CAP_HDR_LOG_1_OFF_0

Offset: 0x120
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**SECOND_DWORD_FOURTH_BYTE**: Byte 3 of Header log register of Second 32-bit Data Word. This field represents fourth byte of Second DW of Header. Note: This register field is sticky.
23:16	0x0	**SECOND_DWORD_THIRD_BYTE**: Byte 2 of Header log register of Second 32-bit Data Word. This field represents third byte of Second DW of Header. Note: This register field is sticky.
15:8	0x0	**SECOND_DWORD_SECOND_BYTE**: Byte 1 of Header log register of Second 32-bit Data Word. This field represents second byte of Second DW of Header. Note: This register field is sticky.
7:0	0x0	**SECOND_DWORD_FIRST_BYTE**: Byte 0 of Header log register of Second 32-bit Data Word. This field represents first byte of Second DW of Header. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_AER_CAP_HDR_LOG_2_OFF_0

where <*j*> = 4, 8.

Description:The Header Log Register 2 contains the header for the TLP corresponding to a detected error; The header is captured such that, when read using DW accesses, the fields of the header are laid out in the same way the headers are presented in the specification. Therefore, byte 0 of the header is located in byte 3 of the Header Log Register 2, byte 1 of the header is in byte 2 of the Header Log Register 2 and so forth. For 12-byte headers, only bytes 0 through 11 of the Header Log Register are used and values in bytes 12 through 15 are undefined. In certain cases where a Malformed TLP is reported, the Header Log Register may contain TLP Prefix information.

PCIE_X4_EP_PF0_AER_CAP_HDR_LOG_2_OFF_0

PCIE_X8_EP_PF0_AER_CAP_HDR_LOG_2_OFF_0

Offset: 0x124
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**THIRD_DWORD_FOURTH_BYTE**: Byte 3 of Header log register of Third 32-bit Data Word. This field represents fourth byte of Third DW of Header. Note: This register field is sticky.
23:16	0x0	**THIRD_DWORD_THIRD_BYTE**: Byte 2 of Header log register of Third 32-bit Data Word. This field represents third byte of Third DW of Header. Note: This register field is sticky.
15:8	0x0	**THIRD_DWORD_SECOND_BYTE**: Byte 1 of Header log register of Third 32-bit Data Word. This field represents second byte of Third DW of Header. Note: This register field is sticky.
7:0	0x0	**THIRD_DWORD_FIRST_BYTE**: Byte 0 of Header log register of Third 32-bit Data Word. This field represents first byte of Third DW of Header. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_AER_CAP_HDR_LOG_3_OFF_0

where <*j*> = 4, 8.

Description:The Header Log Register 3 contains the header for the TLP corresponding to a detected error; The header is captured such that, when read using DW accesses, the fields of the header are laid out in the same way the headers are presented in the specification. Therefore, byte 0 of the header is located in byte 3 of the Header Log Register 3, byte 1 of the header is in byte 2 of the Header Log Register 3 and so forth. For 12-byte headers, only bytes 0 through 11 of the Header Log Register are used and values in bytes 12 through 15 are undefined. In certain cases where a Malformed TLP is reported, the Header Log Register may contain TLP Prefix information.

PCIE_X4_EP_PF0_AER_CAP_HDR_LOG_3_OFF_0

PCIE_X8_EP_PF0_AER_CAP_HDR_LOG_3_OFF_0

Offset: 0x128
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**FOURTH_DWORD_FOURTH_BYTE**: Byte 3 of Header log register of Fourth 32-bit Data Word. This field represents fourth byte of Fourth DW of Header. Note: This register field is sticky.
23:16	0x0	**FOURTH_DWORD_THIRD_BYTE**: Byte 2 of Header log register of Fourth 32-bit Data Word. This field represents third byte of Fourth DW of Header. Note: This register field is sticky.
15:8	0x0	**FOURTH_DWORD_SECOND_BYTE**: Byte 1 of Header log register of Fourth 32-bit Data Word. This field represents second byte of Fourth DW of Header. Note: This register field is sticky.
7:0	0x0	**FOURTH_DWORD_FIRST_BYTE**: Byte 0 of Header log register of Fourth 32-bit Data Word. This field represents first byte of Fourth DW of Header. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_AER_CAP_TLP_PREFIX_LOG_1_OFF_0

where <*j*> = 4, 8.

Description:The First TLP Prefix Log Register contains the first End-End TLP Prefix from the TLP corresponding to the detected error. The TLP Prefix Log Register is only meaningful when the TLP Prefix Log Present bit is Set. The TLP Prefixes are captured such that, when read using DW accesses, the fields of the TLP Prefix are laid out in the same way the fields of the TLP Prefix are described. Therefore, byte 0 of a TLP Prefix is located in byte 3 of the associated TLP Prefix Log Register; byte 1 of a TLP Prefix is located in byte 2; and so forth. The TLP Prefix Log Registers beyond the number supported by the Function are hardwired to zero by controller. If the End-End TLP Prefix Supported bit is Clear, the TLP Prefix Log Register is not required to be implemented.

PCIE_X4_EP_PF0_AER_CAP_TLP_PREFIX_LOG_1_OFF_0

PCIE_X8_EP_PF0_AER_CAP_TLP_PREFIX_LOG_1_OFF_0

Offset: 0x138
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**CFG_TLP_PFX_LOG_1_FOURTH_BYTE**: Byte 3 of Error TLP Prefix Log 1. This field contains fourth byte of First DW of TLP Prefix. Note: This register field is sticky.

Bit	Reset	Description
23:16	0x0	**CFG_TLP_PFX_LOG_1_THIRD_BYTE**: Byte 2 of Error TLP Prefix Log 1. This field contains third byte of First DW of TLP Prefix. Note: This register field is sticky.
15:8	0x0	**CFG_TLP_PFX_LOG_1_SECOND_BYTE**: Byte 1 of Error TLP Prefix Log 1. This field contains second byte of First DW of TLP Prefix. Note: This register field is sticky.
7:0	0x0	**CFG_TLP_PFX_LOG_1_FIRST_BYTE**: Byte 0 of Error TLP Prefix Log 1. This field contains first byte of First DW of TLP Prefix. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_AER_CAP_TLP_PREFIX_LOG_2_OFF_0

where <*j*> = 4, 8.

Description:The Second TLP Prefix Log Register contains the second End-End TLP Prefix from the TLP corresponding to the detected error. The TLP Prefix Log Register is only meaningful when the TLP Prefix Log Present bit is Set. The TLP Prefixes are captured such that, when read using DW accesses, the fields of the TLP Prefix are laid out in the same way the fields of the TLP Prefix are described. Therefore, byte 0 of a TLP Prefix is located in byte 3 of the associated TLP Prefix Log Register; byte 1 of a TLP Prefix is located in byte 2; and so forth. The TLP Prefix Log Registers beyond the number supported by the Function are hardwired to zero by controller. If the End-End TLP Prefix Supported bit is Clear, the TLP Prefix Log Register is not required to be implemented.

PCIE_X4_EP_PF0_AER_CAP_TLP_PREFIX_LOG_2_OFF_0

PCIE_X8_EP_PF0_AER_CAP_TLP_PREFIX_LOG_2_OFF_0

Offset: 0x13c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**CFG_TLP_PFX_LOG_2_FOURTH_BYTE**: Byte 3 Error TLP Prefix Log 2. This field contains fourth byte of Second DW of TLP Prefix. Note: This register field is sticky.
23:16	0x0	**CFG_TLP_PFX_LOG_2_THIRD_BYTE**: Byte 2 Error TLP Prefix Log 2. This field contains third byte of Second DW of TLP Prefix. Note: This register field is sticky.
15:8	0x0	**CFG_TLP_PFX_LOG_2_SECOND_BYTE**: Byte 1 Error TLP Prefix Log 2. This field contains second byte of Second DW of TLP Prefix. Note: This register field is sticky.

Bit	Reset	Description
7:0	0x0	**CFG_TLP_PFX_LOG_2_FIRST_BYTE**: Byte 0 Error TLP Prefix Log 2. This field contains first byte of Second DW of TLP Prefix. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_AER_CAP_TLP_PREFIX_LOG_3_OFF_0

where <*j*> = 4, 8.

Description:The Third TLP Prefix Log Register contains the third End-End TLP Prefix from the TLP corresponding to the detected error. The TLP Prefix Log Register is only meaningful when the TLP Prefix Log Present bit is Set. The TLP Prefixes are captured such that, when read using DW accesses, the fields of the TLP Prefix are laid out in the same way the fields of the TLP Prefix are described. Therefore, byte 0 of a TLP Prefix is located in byte 3 of the associated TLP Prefix Log Register; byte 1 of a TLP Prefix is located in byte 2; and so forth. The TLP Prefix Log Registers beyond the number supported by the Function are hardwired to zero by controller. If the End-End TLP Prefix Supported bit is Clear, the TLP Prefix Log Register is not required to be implemented.

PCIE_X4_EP_PF0_AER_CAP_TLP_PREFIX_LOG_3_OFF_0

PCIE_X8_EP_PF0_AER_CAP_TLP_PREFIX_LOG_3_OFF_0

Offset: 0x140
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**CFG_TLP_PFX_LOG_3_FOURTH_BYTE**: Byte 3 Error TLP Prefix Log 3. This field contains fourth byte of Third DW of TLP Prefix. Note: This register field is sticky.
23:16	0x0	**CFG_TLP_PFX_LOG_3_THIRD_BYTE**: Byte 2 Error TLP Prefix Log 3. This field contains third byte of Third DW of TLP Prefix. Note: This register field is sticky.
15:8	0x0	**CFG_TLP_PFX_LOG_3_SECOND_BYTE**: Byte 1 Error TLP Prefix Log 3. This field contains second byte of Third DW of TLP Prefix. Note: This register field is sticky.
7:0	0x0	**CFG_TLP_PFX_LOG_3_FIRST_BYTE**: Byte 0 Error TLP Prefix Log 3. This field contains first byte of Third DW of TLP Prefix. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_AER_CAP_TLP_PREFIX_LOG_4_OFF_0

where <*j*> = 4, 8.

Description:The Fourth TLP Prefix Log Register contains the fourth End-End TLP Prefix from the TLP corresponding to the detected error. The TLP Prefix Log Register is only meaningful when the TLP Prefix Log Present bit is Set. The TLP Prefixes are captured such that, when read using DW accesses, the fields of the TLP Prefix are laid out in the same way the fields of the TLP Prefix are described. Therefore, byte 0 of a TLP Prefix is located in byte 3 of the associated TLP Prefix Log Register; byte 1 of a TLP Prefix is located in byte 2; and so forth. The TLP Prefix Log Registers beyond the number supported by the Function are hardwired to zero by controller. If the End-End TLP Prefix Supported bit is Clear, the TLP Prefix Log Register is not required to be implemented.

PCIE_X4_EP_PF0_AER_CAP_TLP_PREFIX_LOG_4_OFF_0

PCIE_X8_EP_PF0_AER_CAP_TLP_PREFIX_LOG_4_OFF_0

Offset: 0x144
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**CFG_TLP_PFX_LOG_4_FOURTH_BYTE**: Byte 3 Error TLP Prefix Log 4. This field contains fourth byte of Fourth DW of TLP Prefix. Note: This register field is sticky.
23:16	0x0	**CFG_TLP_PFX_LOG_4_THIRD_BYTE**: Byte 2 Error TLP Prefix Log 4. This field contains third byte of Fourth DW of TLP Prefix. Note: This register field is sticky.
15:8	0x0	**CFG_TLP_PFX_LOG_4_SECOND_BYTE**: Byte 1 Error TLP Prefix Log 4. This field contains second byte of Fourth DW of TLP Prefix. Note: This register field is sticky.
7:0	0x0	**CFG_TLP_PFX_LOG_4_FIRST_BYTE**: Byte 0 Error TLP Prefix Log 4. This field contains first byte of Fourth DW of TLP Prefix. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_SPCIE_CAP_SPCIE_CAP_HEADER_REG_0

where <*j*> = 4, 8.

Description:This Register provides Capability Id, Capability Version, and next Offset of SPCIE Structure.

PCIE_X4_EP_PF0_SPCIE_CAP_SPCIE_CAP_HEADER_REG_0

PCIE_X8_EP_PF0_SPCIE_CAP_SPCIE_CAP_HEADER_REG_0

Offset: 0x148
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x16810019 (0b0001,0110,1000,0001,0000,0000,0001,1001)

Bit	Reset	Description
31:20	0x168	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Extended Capability structure or 000h if no other items exist in the linked list of Capabilities. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x19	**EXTENDED_CAP_ID**: Secondary PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. PCI Express Extended Capability ID for the Secondary PCI Express Extended Capability is 0019h. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_SPCIE_CAP_LINK_CONTROL3_REG_0

where <*j*> = 4, 8.

Description:This Register controls equilization and equilization interrupt.

PCIE_X4_EP_PF0_SPCIE_CAP_LINK_CONTROL3_REG_0

PCIE_X8_EP_PF0_SPCIE_CAP_LINK_CONTROL3_REG_0

Offset: 0x14c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:2	0x0	**RSVDP_2**: Reserved for future use.
1	0x0	**EQ_REQ_INT_EN**: Link Equalization Request Interrupt Enable. This bit is RW for Downstream Ports and for Upstream Ports when Crosslink Supported is 1b. This bit is not applicable and is RsvdP for Upstream Ports when the Crosslink Supported bit is 0b. If the Port does not support 8.0 GT/s, this bit is permitted to be hardwired to 0b by the controller. Note: The access attributes of this field are as follows: - Wire: RSVDP - Dbi: RSVDP
0	0x0	**PERFORM_EQ**: Perform Equalization. This bit is RW for Downstream Ports and for Upstream Ports when Crosslink Supported is 1b. This bit is not applicable and is RsvdP for Upstream Ports when the Crosslink Supported bit is 0b. If the Port does not support 8.0 GT/s, this bit is permitted to be hardwired to 0b by the controller. Note: The access attributes of this field are as follows: - Wire: RSVDP - Dbi: RSVDP

PCIE_X<*j*>_EP_PF0_SPCIE_CAP_LANE_ERR_STATUS_REG_0

where <*j*> = 4, 8.

Description:This Register contains Lane Error Status Bits per Lane.

PCIE_X4_EP_PF0_SPCIE_CAP_LANE_ERR_STATUS_REG_0

PCIE_X8_EP_PF0_SPCIE_CAP_LANE_ERR_STATUS_REG_0

Offset: 0x150
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:4	RO	0x0	**RSVDP_LANE_ERR_STATUS**: Reserved for future use.
3:0	RW	0x0	**LANE_ERR_STATUS**: Lane Error Status Bits per Lane. Each bit indicates if the corresponding Lane detected a Lane-based error. A value of 1b indicates that a Lane based-error was detected on the corresponding Lane Number. For Ports that are narrower than 32 Lanes, the unused upper bits [31: Maximum Link Width] are RsvdZ. For Ports that do not support 8.0 GT/s and do not set these bits based on 8b/10b errors , this field is permitted to be hardwired to 0 by the controller.

PCIE_X<*j*>_EP_PF0_SPCIE_CAP_SPCIE_CAP_OFF_0CH_REG_0

where <*j*> = 4, 8.

Description:This register provides Transmitter Preset and Receiver Preset Hint for Downstream Port and Upstream Port.

PCIE_X4_EP_PF0_SPCIE_CAP_SPCIE_CAP_OFF_0CH_REG_0

PCIE_X8_EP_PF0_SPCIE_CAP_SPCIE_CAP_OFF_0CH_REG_0

Offset: 0x154
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x74007400 (0b0111,0100,0000,0000,0111,0100,0000,0000)

Bit	Reset	Description
31	0x0	**RSVDP_31**: Reserved for future use.
30:28	0x7	**USP_RX_PRESET_HINT1**: Upstream Port 8.0 GT/s Receiver Preset Hint 1. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Receiver preset hint 1 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: This register field is sticky.
27:24	0x4	**USP_TX_PRESET1**: Upstream Port 8.0 GT/s Transmitter Preset 1. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Transmitter preset 1 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: This register field is sticky.
23	0x0	**RSVDP_23**: Reserved for future use.

Bit	Reset	Description
22:20	0x0	**DSP_RX_PRESET_HINT1**: Downstream Port 8.0 GT/s Receiver Preset Hint 1. Receiver preset hint 1 value that may be used as a suggested setting for 8.0 GT/s receiver equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit.
19:16	0x0	**DSP_TX_PRESET1**: Downstream Port 8.0 GT/s Transmitter Preset 1. Transmitter preset 1 value used for 8.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit.
15	0x0	**RSVDP_15**: Reserved for future use.
14:12	0x7	**USP_RX_PRESET_HINT0**: Upstream Port 8.0 GT/s Receiver Preset Hint 0. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Receiver preset hint 0 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: This register field is sticky.
11:8	0x4	**USP_TX_PRESET0**: Upstream Port 8.0 GT/s Transmitter Preset 0. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Transmitter preset 0 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: This register field is sticky.
7	0x0	**RSVDP_7**: Reserved for future use.
6:4	0x0	**DSP_RX_PRESET_HINT0**: Downstream Port 8.0 GT/s Receiver Preset Hint 0. Receiver preset hint 0 value that may be used as a suggested setting for 8.0 GT/s receiver equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit.
3:0	0x0	**DSP_TX_PRESET0**: Downstream Port 8.0 GT/s Transmitter Preset 0. Transmitter preset 0 value used for 8.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit.

PCIE_X<*j*>_EP_PF0_SPCIE_CAP_SPCIE_CAP_OFF_10H_REG_0

where <*j*> = 4, 8.

Description:The function of this register is dependent on your actual configuration. - Gen3: LEC or RSVD depending on the value of CX_NL. - Gen4: LEC or LEC2 or RSVD depending on the value of CX_NL. This register provides Transmitter Preset and Receiver Preset Hint for Downstream Port and Upstream Port.

PCIE_X4_EP_PF0_SPCIE_CAP_SPCIE_CAP_OFF_10H_REG_0

PCIE_X8_EP_PF0_SPCIE_CAP_SPCIE_CAP_OFF_10H_REG_0

Offset: 0x158
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x74007400 (0b0111,0100,0000,0000,0111,0100,0000,0000)

Bit	Reset	Description
31	0x0	**RSVDP_31**: Reserved for future use.
30:28	0x7	**USP_RX_PRESET_HINT3**: Upstream Port 8.0 GT/s Receiver Preset Hint3. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Receiver preset hint 3 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: This register field is sticky.
27:24	0x4	**USP_TX_PRESET3**: Upstream Port 8.0 GT/s Transmitter Preset3. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Transmitter preset 3 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: This register field is sticky.

Bit	Reset	Description
23	0x0	**RSVDP_23**: Reserved for future use.
22:20	0x0	**DSP_RX_PRESET_HINT3**: Downstream Port 8.0 GT/s Receiver Preset Hint3. Receiver preset hint 3 value that may be used as a suggested setting for 8.0 GT/s receiver equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit.
19:16	0x0	**DSP_TX_PRESET3**: Downstream Port 8.0 GT/s Transmitter Preset3. Transmitter preset 3 value used for 8.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit.
15	0x0	**RSVDP_15**: Reserved for future use.
14:12	0x7	**USP_RX_PRESET_HINT2**: Upstream Port 8.0 GT/s Receiver Preset Hint2. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Receiver preset hint 2 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: This register field is sticky.
11:8	0x4	**USP_TX_PRESET2**: Upstream Port 8.0 GT/s Transmitter Preset2. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Transmitter preset 2 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: This register field is sticky.
7	0x0	**RSVDP_7**: Reserved for future use.
6:4	0x0	**DSP_RX_PRESET_HINT2**: Downstream Port 8.0 GT/s Receiver Preset Hint2. Receiver preset hint 2 value that may be used as a suggested setting for 8.0 GT/s receiver equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit.

Bit	Reset	Description
3:0	0x0	**DSP_TX_PRESET2**: Downstream Port 8.0 GT/s Transmitter Preset2. Transmitter preset 2 value used for 8.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit.

PCIE_X8_EP_PF0_SPCIE_CAP_SPCIE_CAP_OFF_14H_REG_0

Description: The function of this register is dependent on your actual configuration. - Gen3: LEC or RSVD depending on the value of CX_NL. - Gen4: LEC or LEC2 or RSVD depending on the value of CX_NL. This register provides Transmitter Preset and Receiver Preset Hint for Downstream Port and Upstream Port.

Offset: 0x15c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x74007400 (0b0111,0100,0000,0000,0111,0100,0000,0000)

Bit	Reset	Description
Bit	Reset	Description
31	0x0	**RSVDP_31**: Reserved for future use.
30:28	0x7	**USP_RX_PRESET_HINT5**: Upstream Port 8.0 GT/s Receiver Preset Hint5. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Receiver preset hint 5 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: This register field is sticky.

Bit	Reset	Description
Bit	**Reset**	**Description**
27:24	0x4	**USP_TX_PRESET5**: Upstream Port 8.0 GT/s Transmitter Preset5. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Transmitter preset 5 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: This register field is sticky.
23	0x0	**RSVDP_23**: Reserved for future use.
22:20	0x0	**DSP_RX_PRESET_HINT5**: Downstream Port 8.0 GT/s Receiver Preset Hint5. Receiver preset hint 5 value that may be used as a suggested setting for 8.0 GT/s receiver equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit.
19:16	0x0	**DSP_TX_PRESET5**: Downstream Port 8.0 GT/s Transmitter Preset5. Transmitter preset 5 value used for 8.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit.
15	0x0	**RSVDP_15**: Reserved for future use.
14:12	0x7	**USP_RX_PRESET_HINT4**: Upstream Port 8.0 GT/s Receiver Preset Hint4. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Receiver preset hint 4 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: This register field is sticky.

Bit	Reset	Description
Bit	**Reset**	**Description**
11:8	0x4	**USP_TX_PRESET4**: Upstream Port 8.0 GT/s Transmitter Preset4. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Transmitter preset 4 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: This register field is sticky.
7	0x0	**RSVDP_7**: Reserved for future use.
6:4	0x0	**DSP_RX_PRESET_HINT4**: Downstream Port 8.0 GT/s Receiver Preset Hint4. Receiver preset hint 4 value that may be used as a suggested setting for 8.0 GT/s receiver equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit.
3:0	0x0	**DSP_TX_PRESET4**: Downstream Port 8.0 GT/s Transmitter Preset4. Transmitter preset 4 value used for 8.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit.

PCIE_X8_EP_PF0_SPCIE_CAP_SPCIE_CAP_OFF_18H_REG_0

Description: The function of this register is dependent on your actual configuration. - Gen3: LEC or RSVD depending on the value of CX_NL. - Gen4: LEC or LEC2 or RSVD depending on the value of CX_NL. This register provides Transmitter Preset and Receiver Preset Hint for Downstream Port and Upstream Port.

Offset: 0x160
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x74007400 (0b0111,0100,0000,0000,0111,0100,0000,0000)

Bit	Reset	Description
31	0x0	**RSVDP_31**: Reserved for future use.

Bit	Reset	Description
30:28	0x7	**USP_RX_PRESET_HINT7**: Upstream Port 8.0 GT/s Receiver Preset Hint7. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Receiver preset hint 7 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: This register field is sticky.
27:24	0x4	**USP_TX_PRESET7**: Upstream Port 8.0 GT/s Transmitter Preset7. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Transmitter preset 7 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: This register field is sticky.
23	0x0	**RSVDP_23**: Reserved for future use.
22:20	0x0	**DSP_RX_PRESET_HINT7**: Downstream Port 8.0 GT/s Receiver Preset Hint7. Receiver preset hint 7 value that may be used as a suggested setting for 8.0 GT/s receiver equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit.
19:16	0x0	**DSP_TX_PRESET7**: Downstream Port 8.0 GT/s Transmitter Preset7. Transmitter preset 7 value used for 8.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit.
15	0x0	**RSVDP_15**: Reserved for future use.

Bit	Reset	Description
14:12	0x7	**USP_RX_PRESET_HINT6**: Upstream Port 8.0 GT/s Receiver Preset Hint6. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Receiver preset hint 6 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: This register field is sticky.
11:8	0x4	**USP_TX_PRESET6**: Upstream Port 8.0 GT/s Transmitter Preset6. The write value is gated with the PCIE_CAP_CROSS_LINK_SUPPORT field of LINK_CAPABILITIES2_REG. Field contains the Transmitter preset 6 value sent or received during 8.0 GT/s Link Equalization. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field representss the value sent on Lane 0 during 8.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO. Note: This register field is sticky.
7	0x0	**RSVDP_7**: Reserved for future use.
6:4	0x0	**DSP_RX_PRESET_HINT6**: Downstream Port 8.0 GT/s Receiver Preset Hint6. Receiver preset hint 6 value that may be used as a suggested setting for 8.0 GT/s receiver equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit.
3:0	0x0	**DSP_TX_PRESET6**: Downstream Port 8.0 GT/s Transmitter Preset6. Transmitter preset 6 value used for 8.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit.

PCIE_X<*j*>_EP_PF0_PL16G_CAP_PL16G_EXT_CAP_HDR_REG_0

where <*j*> = 4, 8.

Description:Physical Layer 16.0 GT/s Extended Capability Header provides information about Capability ID, Version, and next offset.

Offset: 0x168
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X4_EP_PF0_PL16G_CAP_PL16G_EXT_CAP_HDR_REG_0

Reset: 0x18c10026 (0b0001,1000,1100,0001,0000,0000,0010,0110)

Bit	Reset	Description
31:20	0x18c	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. This bit depends on the version of the specification. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x26	**EXTENDED_CAP_ID**: PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. The Extended Capability ID for the Physical Layer 16.0 GT/s Capability is 0026h. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X8_EP_PF0_PL16G_CAP_PL16G_EXT_CAP_HDR_REG_0

Reset: 0x19010026 (0b0001,1001,0000,0001,0000,0000,0010,0110)

Bit	Reset	Description
31:20	0x190	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. This bit depends on the version of the specification. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

Bit	Reset	Description
15:0	0x26	**EXTENDED_CAP_ID**: PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. The Extended Capability ID for the Physical Layer 16.0 GT/s Capability is 0026h. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PL16G_CAP_PL16G_CAPABILITY_REG_0

where <*j*> = 4, 8.

Description:This register is reserved for the future update.

PCIE_X4_EP_PF0_PL16G_CAP_PL16G_CAPABILITY_REG_0

PCIE_X8_EP_PF0_PL16G_CAP_PL16G_CAPABILITY_REG_0

Offset: 0x16c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**RSVDP_0**: Reserved for future use.

PCIE_X<*j*>_EP_PF0_PL16G_CAP_PL16G_CONTROL_REG_0

where <*j*> = 4, 8.

Description:This register is reserved for the future update.

PCIE_X4_EP_PF0_PL16G_CAP_PL16G_CONTROL_REG_0

PCIE_X8_EP_PF0_PL16G_CAP_PL16G_CONTROL_REG_0

Offset: 0x170
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**RSVDP_0**: Reserved for future use.

PCIE_X<*j*>_EP_PF0_PL16G_CAP_PL16G_STATUS_REG_0

where <*j*> = 4, 8.

Description:16.0 GT/s Status Register provides status of equalization of 16.0 GT/s speed.

PCIE_X4_EP_PF0_PL16G_CAP_PL16G_STATUS_REG_0

PCIE_X8_EP_PF0_PL16G_CAP_PL16G_STATUS_REG_0

Offset: 0x174
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:5	RO	0x0	**RSVDP_5**: Reserved for future use.
4	RW	0x0	**LINK_EQ_16G_REQ**: Link Equalization Request 16.0GT/s. For a Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions.
3	RO	0x0	**EQ_16G_CPL_P3**: Equalization 16.0GT/s Phase 3 Successful. For a Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions. Note: This register field is sticky.
2	RO	0x0	**EQ_16G_CPL_P2**: Equalization 16.0GT/s Phase 2 Successful. For a Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions. Note: This register field is sticky.
1	RO	0x0	**EQ_16G_CPL_P1**: Equalization 16.0GT/s Phase 1 Successful. For a Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions. Note: This register field is sticky.
0	RO	0x0	**EQ_16G_CPL**: Equalization 16.0GT/s Complete. For a Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PL16G_CAP_PL16G_LC_DPAR_STATUS_REG_0

where <*j*> = 4, 8.

Description:The Local Data Parity Mismatch Status register is a 32-bit vector where each bit indicates if the local receiver detected a Data Parity mismatch on the Lane with the corresponding Lane number. This Lane number is the default Lane number which is invariant to Link width and Lane reversal negotiation that occurs during Link training.

PCIE_X4_EP_PF0_PL16G_CAP_PL16G_LC_DPAR_STATUS_REG_0

PCIE_X8_EP_PF0_PL16G_CAP_PL16G_LC_DPAR_STATUS_REG_0

Offset: 0x178
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:4	RO	0x0	**RSVDP_LC_DPAR_STATUS**: Reserved for future use.
3:0	RW	0x0	**LC_DPAR_STATUS**: Local Data Parity Mismatch Status. Each bit indicates if the corresponding Lane detected a Data Parity mismatch. A value of 1b indicates that a mismatch was detected on the corresponding Lane Number. The default value of each bit is 0b. For Ports that are narrower than 32 Lanes, the unused upper bits [31: MaximumLink Width] are RsvdZ.

PCIE_X<*j*>_EP_PF0_PL16G_CAP_PL16G_FIRST_RETIMER_DPAR_STATUS_REG_0

where <*j*> = 4, 8.

Description:The First Retimer Data Parity Status register is a 32-bit vector where each bit indicates if the first Retimer of a Path detected a Data Parity mismatch on the Lane with the corresponding Lane number. This Lane number is the default Lane number which is invariant to Link width and Lane reversal negotiation that occurs during Link training.

PCIE_X4_EP_PF0_PL16G_CAP_PL16G_FIRST_RETIMER_DPAR_STATUS_REG_0

PCIE_X8_EP_PF0_PL16G_CAP_PL16G_FIRST_RETIMER_DPAR_STATUS_REG_0

Offset: 0x17c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:4	RO	0x0	**RSVDP_FIRST_RETIMER_DPAR_STATUS**: Reserved for future use.
3:0	RW	0x0	**FIRST_RETIMER_DPAR_STATUS**: First Retimer Data Parity Mismatch Status. Each bit indicates if the corresponding Lane detected a Data Parity mismatch. A value of 1b indicates that a mismatch was detected on the corresponding Lane Number. The default value of each bit is 0b. The value of this field is undefined when no Retimers are present. For Ports that are narrower than 32 Lanes, the unused upper bits [31: Maximum Link Width] are RsvdZ.

PCIE_X<*j*>_EP_PF0_PL16G_CAP_PL16G_SECOND_RETIMER_DPAR_STATUS_REG_0

where <*j*> = 4, 8.

Description:The Second Retimer Data Parity Status register is a 32-bit vector where each bit indicates if the second Retimer of a Path detected a Data Parity mismatch on the Lane with the corresponding Lane number. This Lane number is the default Lane number which is invariant to Link width and Lane reversal negotiation that occurs during Link training.

PCIE_X4_EP_PF0_PL16G_CAP_PL16G_SECOND_RETIMER_DPAR_STATUS_REG_0

PCIE_X8_EP_PF0_PL16G_CAP_PL16G_SECOND_RETIMER_DPAR_STATUS_REG_0

Offset: 0x180
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:4	RO	0x0	**RSVDP_SECOND_RETIMER_DPAR_STATUS**: Reserved for future use.

Bit	R/W	Reset	Description
3:0	RW	0x0	**SECOND_RETIMER_DPAR_STATUS**: Second Retimer Data Parity Mismatch Status. Each bit indicates if the corresponding Lane detected a Data Parity mismatch. A value of 1b indicates that a mismatch was detected on the corresponding Lane Number. The default value of each bit is 0b. The value of this field is undefined when no Retimers are present or only one Retimer is present. For Ports that are narrower than 32 Lanes, the unused upper bits [31: Maximum Link Width] are RsvdZ.

PCIE_X<*j*>_EP_PF0_PL16G_CAP_PL16G_CAP_OFF_20H_REG_0

where <*j*> = 4, 8.

Description:This Equalization Control register consists of control fields required for Lane 0-3 16.0 GT/s equalization.

PCIE_X4_EP_PF0_PL16G_CAP_PL16G_CAP_OFF_20H_REG_0

PCIE_X8_EP_PF0_PL16G_CAP_PL16G_CAP_OFF_20H_REG_0

Offset: 0x188
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x40404040 (0b0100,0000,0100,0000,0100,0000,0100,0000)

Bit	Reset	Description
31:28	0x4	**USP_16G_TX_PRESET3**: Upstream Port 16.0 GT/s Transmitter Preset3. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field represents the value sent on Lane 3 during 16.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 3 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO.
27:24	0x0	**DSP_16G_TX_PRESET3**: Downstream Port 16.0 GT/s Transmitter Preset3. Transmitter Preset of Lane 3 used for 16.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit.

Bit	Reset	Description
23:20	0x4	**USP_16G_TX_PRESET2**: Upstream Port 16.0 GT/s Transmitter Preset2. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field represents the value sent on Lane 2 during 16.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 2 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO.
19:16	0x0	**DSP_16G_TX_PRESET2**: Downstream Port 16.0 GT/s Transmitter Preset2. Transmitter Preset of Lane 2 used for 16.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit.
15:12	0x4	**USP_16G_TX_PRESET1**: Upstream Port 16.0 GT/s Transmitter Preset1. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field represents the value sent on Lane 1 during 16.0 GT/s equalization. - CaseB: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 1 during Link Equalization. Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO.
11:8	0x0	**DSP_16G_TX_PRESET1**: Downstream Port 16.0 GT/s Transmitter Preset1. Transmitter Preset of Lane 1 used for 16.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit.
7:4	0x4	**USP_16G_TX_PRESET0**: Upstream Port 16.0 GT/s Transmitter Preset0. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field represents the value sent on Lane 0 during 16.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 0 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO.
3:0	0x0	**DSP_16G_TX_PRESET0**: Downstream Port 16.0 GT/s Transmitter Preset0. Transmitter Preset of Lane 0 used for 16.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit.

PCIE_X8_EP_PF0_PL16G_CAP_PL16G_CAP_OFF_24H_REG_0

Description: This Equalization Control register consists of control fields required for Lane 4-7 16.0 GT/s equalization.

Offset: 0x18c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x40404040 (0b0100,0000,0100,0000,0100,0000,0100,0000)

Bit	Reset	Description
31:28	0x4	**USP_16G_TX_PRESET7**: Upstream Port 16.0 GT/s Transmitter Preset7. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field represents the value sent on Lane 7 during 16.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 7 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO.
27:24	0x0	**DSP_16G_TX_PRESET7**: Downstream Port 16.0 GT/s Transmitter Preset7. Transmitter Preset of Lane 7 used for 16.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP . Otherwise, this field is HwInit.
23:20	0x4	**USP_16G_TX_PRESET6**: Upstream Port 16.0 GT/s Transmitter Preset6. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field represents the value sent on Lane 6 during 16.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 6 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO.
19:16	0x0	**DSP_16G_TX_PRESET6**: Downstream Port 16.0 GT/s Transmitter Preset6. Transmitter Preset of Lane 6 used for 16.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit.
15:12	0x4	**USP_16G_TX_PRESET5**: Upstream Port 16.0 GT/s Transmitter Preset5. - Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field represents the value sent on Lane 5 during 16.0 GT/s equalization. - Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 5 during Link Equalization. - Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO.

Bit	Reset	Description
11:8	0x0	**DSP_16G_TX_PRESET5**: Downstream Port 16.0 GT/s Transmitter Preset5. Transmitter Preset of Lane 5 used for 16.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit.
7:4	0x4	**USP_16G_TX_PRESET4**: Upstream Port 16.0 GT/s Transmitter Preset4. -Case A: When Operating port is Downstream port and whether the Crosslink is supported or not, this field represents the value sent on Lane 4 during 16.0 GT/s equalization. -Case B: When Operating port is Upstream port and Crosslink Supported bit is 0, this field is intended for debug and diagnostics. It contains the value captured from Lane 4 during Link Equalization. Case C: When Operating port is Upstream port and Crosslink Supported bit is 1, Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A applies. For case A and C, Field is HwInit. For case B, Field is RO.
3:0	0x0	**DSP_16G_TX_PRESET4**: Downstream Port 16.0 GT/s Transmitter Preset4. Transmitter Preset of Lane 4 used for 16.0 GT/s equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit.

PCIE_X<*j*>_EP_PF0_MARGIN_CAP_MARGIN_EXT_CAP_HDR_REG_0

where <*j*> = 4, 8.

Description:This register provides capbility ID, capability version and next offset value for Margining Extended Capability.

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X4_EP_PF0_MARGIN_CAP_MARGIN_EXT_CAP_HDR_REG_0

Offset: 0x18c
Reset: 0x1a410027 (0b0001,1010,0100,0001,0000,0000,0010,0111)

Bit	Reset	Description
31:20	0x1a4	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greaterthan 0FFh. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. This field is depends on version of the specification. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x27	**EXTENDED_CAP_ID**: PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. The Extended Capability ID for the Physical Layer 16.0 GT/s Margining Extended Capability is 0027h. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X8_EP_PF0_MARGIN_CAP_MARGIN_EXT_CAP_HDR_REG_0

Offset: 0x190
Reset: 0x1b810027 (0b0001,1011,1000,0001,0000,0000,0010,0111)

Bit	Reset	Description
31:20	0x1b8	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greaterthan 0FFh. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. This field is depends on version of the specification. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x27	**EXTENDED_CAP_ID**: PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. The Extended Capability ID for the Physical Layer 16.0 GT/s Margining Extended Capability is 0027h. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_MARGIN_CAP_MARGIN_PORT_CAPABILITIES_STATUS_REG_0

where <*j*> = 4, 8.

Description:This register indicates the status of the Margining feature.

PCIE_X4_EP_PF0_MARGIN_CAP_MARGIN_PORT_CAPABILITIES_STATUS_REG_0

Offset: 0x190

PCIE_X8_EP_PF0_MARGIN_CAP_MARGIN_PORT_CAPABILITIES_STATUS_REG_0

Offset: 0x194

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000001 (0b0000,0000,0000,0000,0000,0000,0000,0001)

Bit	R/W	Reset	Description
31:18	RO	0x0	**RSVDP_18**: Reserved for future use.
17	RO	0x0	**MARGINING_SOFTWARE_READY**: Margining Software Ready. When Margining uses Driver Software is Set, then this bit, when Set, indicates that the required software has performed the required initialization. The value of this bit is undefined if Margining uses Driver Software is Clear. The default value of this bit is implementation specific. If Margining uses Driver Software is Clear, Margining Ready must be Set no later than 100 ms after the Link trains to 16.0 GT/s. Default value is implementation specific.
16	RO	0x0	**MARGINING_READY**: Margining Ready. Indicates when the Margining feature is ready to accept margining commands. Behavior is undefined if this bit is Clear and, for any Lane, any of the Receiver Number , Margin Type , Usage Model , or Margin Payload fields are written. If Margining uses Driver Software is Set, Margining Ready must be Set no later than 100 ms after the later of Margining Software Ready becoming Set or the link training to 16.0 GT/s.
15:1	RO	0x0	**RSVDP_1**: Reserved for future use.
0	RW	0x1	**MARGINING_USES_DRIVER_SOFTWARE**: Margining uses Driver Software. If Set, indicates that Margining is partially implemented using Device Driver software. Margining Software Ready indicates when this software is initialized. If Clear, Margining does not require device driver software. In this case the value read from Margining Software Ready is undefined. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_MARGIN_CAP_MARGIN_LANE_CNTRL_STATUS0_REG_0

where <*j*> = 4, 8.

Description:The Margining Lane Control Register consists of control fields required for per-Lane margining. The number of entries in this register are sized by Maximum Link Width.

PCIE_X4_EP_PF0_MARGIN_CAP_MARGIN_LANE_CNTRL_STATUS0_REG_0

Offset: 0x194

PCIE_X8_EP_PF0_MARGIN_CAP_MARGIN_LANE_CNTRL_STATUS0_REG_0

Offset: 0x198

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00009c38 (0b0000,0000,0000,0000,1001,1100,0011,1000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**MARGIN_PAYLOAD_STATUS**: Margin Payload(Status) for Lane 0. This field is only meaningful, when the Margin Type is a defined encoding other than 'No Command'. This field must be reset to the default value if the Port goes to DL_Down status.
23	RO	0x0	**RSVDP_23**: Reserved for future use.
22	RO	0x0	**USAGE_MODEL_STATUS**: Usage Model(Status) for Lane 0. This field must be reset to the default value if the Port goes to DL_Down status.
21:19	RO	0x0	**MARGIN_TYPE_STATUS**: Margin Type(Status) for Lane 0. This field must be reset to the default value if the Port goes to DL_Down status.
18:16	RO	0x0	**RECEIVER_NUMBER_STATUS**: Receiver Number(Status) for Lane 0. For Downstream Ports, this field must be reset to the default value if the Port goes to DL_Down status.
15:8	RW	0x9c	**MARGIN_PAYLOAD**: Margin Payload for Lane 0. This field must be reset to the default value if the Port goes to DL_Down status.
7	RO	0x0	**RSVDP_7**: Reserved for future use.

Bit	R/W	Reset	Description
6	RW	0x0	**USAGE_MODEL**: Usage Model for Lane 0. This field must be reset to the default value if the Port goes to DL_Down status.
5:3	RW	0x7	**MARGIN_TYPE**: Margin Type for Lane 0. The default value is 111b. This field must be reset to the default value if the Port goes to DL_Down status.
2:0	RW	0x0	**RECEIVER_NUMBER**: Receiver Number for Lane 0. This field must be reset to the default value if the Port goes to DL_Down status.

PCIE_X<*j*>_EP_PF0_MARGIN_CAP_MARGIN_LANE_CNTRL_STATUS1_REG_0

where <*j*> = 4, 8.

Description:The Margining Lane Control Register consists of control fields required for per-Lane margining. The number of entries in this register are sized by Maximum Link Width.

PCIE_X4_EP_PF0_MARGIN_CAP_MARGIN_LANE_CNTRL_STATUS1_REG_0

Offset: 0x198

PCIE_X8_EP_PF0_MARGIN_CAP_MARGIN_LANE_CNTRL_STATUS1_REG_0

Offset: 0x19c

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00009c38 (0b0000,0000,0000,0000,1001,1100,0011,1000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**MARGIN_PAYLOAD_STATUS**: Margin Payload(Status) for Lane 1. This field is only meaningful, when the Margin Type is a defined encoding other than 'No Command'. This field must be reset to the default value if the Port goes to DL_Down status.
23	RO	0x0	**RSVDP_23**: Reserved for future use.
22	RO	0x0	**USAGE_MODEL_STATUS**: Usage Model(Status) for Lane 1. This field must be reset to the default value if the Port goes to DL_Down status.
21:19	RO	0x0	**MARGIN_TYPE_STATUS**: Margin Type(Status) for Lane 1. This field must be reset to the default value if the Port goes to DL_Down status.

Bit	R/W	Reset	Description
18:16	RO	0x0	**RECEIVER_NUMBER_STATUS**: Receiver Number(Status) for Lane 1. For Downstream Ports, this field must be reset to the default value if the Port goes to DL_Down status.
15:8	RW	0x9c	**MARGIN_PAYLOAD**: Margin Payload for Lane 1. This field must be reset to the default value if the Port goes to DL_Down status.
7	RO	0x0	**RSVDP_7**: Reserved for future use.
6	RW	0x0	**USAGE_MODEL**: Usage Model for Lane 1. This field must be reset to the default value if the Port goes to DL_Down status.
5:3	RW	0x7	**MARGIN_TYPE**: Margin Type for Lane 1. The default value is 111b. This field must be reset to the default value if the Port goes to DL_Down status.
2:0	RW	0x0	**RECEIVER_NUMBER**: Receiver Number for Lane 1. This field must be reset to the default value if the Port goes to DL_Down status.

PCIE_X<*j*>_EP_PF0_MARGIN_CAP_MARGIN_LANE_CNTRL_STATUS2_REG_0

where <*j*> = 4, 8.

Description:The Margining Lane Control Register consists of control fields required for per-Lane margining. The number of entries in this register are sized by Maximum Link Width.

PCIE_X4_EP_PF0_MARGIN_CAP_MARGIN_LANE_CNTRL_STATUS2_REG_0

Offset: 0x19c

PCIE_X8_EP_PF0_MARGIN_CAP_MARGIN_LANE_CNTRL_STATUS2_REG_0

Offset: 0x1a0

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00009c38 (0b0000,0000,0000,0000,1001,1100,0011,1000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**MARGIN_PAYLOAD_STATUS**: Margin Payload(Status) for Lane 2. This field is only meaningful, when the Margin Type is a defined encoding other than 'No Command'. This field must be reset to the default value if the Port goes to DL_Down status.
23	RO	0x0	**RSVDP_23**: Reserved for future use.
22	RO	0x0	**USAGE_MODEL_STATUS**: Usage Model(Status) for Lane 2. This field must be reset to the default value if the Port goes to DL_Down status.
21:19	RO	0x0	**MARGIN_TYPE_STATUS**: Margin Type(Status) for Lane 2. This field must be reset to the default value if the Port goes to DL_Down status.
18:16	RO	0x0	**RECEIVER_NUMBER_STATUS**: Receiver Number(Status) for Lane 2. For Downstream Ports, this field must be reset to the default value if the Port goes to DL_Down status.
15:8	RW	0x9c	**MARGIN_PAYLOAD**: Margin Payload for Lane 2. This field must be reset to the default value if the Port goes to DL_Down status.
7	RO	0x0	**RSVDP_7**: Reserved for future use.
6	RW	0x0	**USAGE_MODEL**: Usage Model for Lane 2. This field must be reset to the default value if the Port goes to DL_Down status.
5:3	RW	0x7	**MARGIN_TYPE**: Margin Type for Lane 2. The default value is 111b. This field must be reset to the default value if the Port goes to DL_Down status.
2:0	RW	0x0	**RECEIVER_NUMBER**: Receiver Number for Lane 2. This field must be reset to the default value if the Port goes to DL_Down status.

PCIE_X<*j*>_EP_PF0_MARGIN_CAP_MARGIN_LANE_CNTRL_STATUS3_REG_0

where <*j*> = 4, 8.

Description:The Margining Lane Control Register consists of control fields required for per-Lane margining. The number of entries in this register are sized by Maximum Link Width.

PCIE_X4_EP_PF0_MARGIN_CAP_MARGIN_LANE_CNTRL_STATUS3_REG_0

Offset: 0x1a0

PCIE_X8_EP_PF0_MARGIN_CAP_MARGIN_LANE_CNTRL_STATUS3_REG_0

Offset: 0x1a4

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00009c38 (0b0000,0000,0000,0000,1001,1100,0011,1000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**MARGIN_PAYLOAD_STATUS**: Margin Payload(Status) for Lane 3. This field is only meaningful, when the Margin Type is a defined encoding other than 'No Command'. This field must be reset to the default value if the Port goes to DL_Down status.
23	RO	0x0	**RSVDP_23**: Reserved for future use.
22	RO	0x0	**USAGE_MODEL_STATUS**: Usage Model(Status) for Lane 3. This field must be reset to the default value if the Port goes to DL_Down status.
21:19	RO	0x0	**MARGIN_TYPE_STATUS**: Margin Type(Status) for Lane 3. This field must be reset to the default value if the Port goes to DL_Down status.
18:16	RO	0x0	**RECEIVER_NUMBER_STATUS**: Receiver Number(Status) for Lane 3. For Downstream Ports, this field must be reset to the default value if the Port goes to DL_Down status.
15:8	RW	0x9c	**MARGIN_PAYLOAD**: Margin Payload for Lane 3. This field must be reset to the default value if the Port goes to DL_Down status.
7	RO	0x0	**RSVDP_7**: Reserved for future use.
6	RW	0x0	**USAGE_MODEL**: Usage Model for Lane 3. This field must be reset to the default value if the Port goes to DL_Down status.
5:3	RW	0x7	**MARGIN_TYPE**: Margin Type for Lane 3. The default value is 111b. This field must be reset to the default value if the Port goes to DL_Down status.
2:0	RW	0x0	**RECEIVER_NUMBER**: Receiver Number for Lane 3. This field must be reset to the default value if the Port goes to DL_Down status.

PCIE_X<*j*>_EP_PF0_LTR_CAP_LTR_CAP_HDR_REG_0

where <*j*> = 4, 8.

Description:This register provides capbility ID, capability version and next offset value for LTR (Latency Tolerance Reporting).

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X4_EP_PF0_LTR_CAP_LTR_CAP_HDR_REG_0

Offset: 0x1a4
Reset: 0x1ac10018 (0b0001,1010,1100,0001,0000,0000,0001,1000)

Bit	Reset	Description
31:20	0x1ac	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Extended Capability structure or 000h if no other items exist in the linked list of Capabilities. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. This field is depends on version of the specification. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x18	**CAP_ID**: LTR Extended Capacity ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. PCI Express Extended Capability for the LTR Extended Capability is 0018h. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X8_EP_PF0_LTR_CAP_LTR_CAP_HDR_REG_0

Offset: 0x1b8
Reset: 0x1c010018 (0b0001,1100,0000,0001,0000,0000,0001,1000)

Bit	Reset	Description
31:20	0x1c0	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Extended Capability structure or 000h if no other items exist in the linked list of Capabilities. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. This field is depends on version of the specification. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

Bit	Reset	Description
15:0	0x18	**CAP_ID**: LTR Extended Capacity ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. PCI Express Extended Capability for the LTR Extended Capability is 0018h. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_LTR_CAP_LTR_LATENCY_REG_0

where <*j*> = 4, 8.

Description:This register indicates Latency scale and vlaue for Max Snoop and No-Snoop.

PCIE_X4_EP_PF0_LTR_CAP_LTR_LATENCY_REG_0

Offset: 0x1a8

PCIE_X8_EP_PF0_LTR_CAP_LTR_LATENCY_REG_0

Offset: 0x1bc

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:29	RO	0x0	**RSVDP_29**: Reserved for future use.
28:26	RW	0x0	**MAX_NO_SNOOP_LAT_SCALE**: Max No-Snoop Latency Scale. This register provides a scale for the value contained within the Max No-Snoop LatencyValue field. Encoding is the same as the LatencyScale fields in the LTR Message. It is strongly recommended that any updates to this field are reflected in LTR Message(s) sent by the device within 1 ms. Hardware operation is undefined if software writes a Not Permitted value to this field.
25:16	RW	0x0	**MAX_NO_SNOOP_LAT**: Max No-Snoop Latency Value. Along with the Max No-Snoop LatencyScale field, this register specifies the maximum no-snoop latency that a device is permitted to request. Software should set this to the platform's maximum supported latency or less. It is strongly recommended that any updates to this field are reflected in LTR Message(s) sent by the device within 1 ms.
15:13	RO	0x0	**RSVDP_13**: Reserved for future use.

Bit	R/W	Reset	Description
12:10	RW	0x0	**MAX_SNOOP_LAT_SCALE**: Max Snoop Latency Scale. This register provides a scale for the value contained within the Max Snoop LatencyValue field. Encoding is the same as the LatencyScale fields in the LTR Message. It is strongly recommended that any updates to this field are reflected in LTR Message(s) sent by the device within 1 ms. Hardware operation is undefined if software writes a Not Permitted value to this field.
9:0	RW	0x0	**MAX_SNOOP_LAT**: Max Snoop Latency Value. Along with the Max Snoop LatencyScale field,this register specifies the maximum snoop latency that a device is permitted to request. Software should set this to the platform's maximum supported latency or less. It is strongly recommended that any updates to this field are reflected in LTR Message(s) sent by the device within 1 ms.

PCIE_X<*j*>_EP_PF0_L1SUB_CAP_L1SUB_CAP_HEADER_REG_0

where <*j*> = 4, 8.

Description:L1 Substates Extended Capability Header provides capability ID, capability version and next offset value for L1 Substates.

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X4_EP_PF0_L1SUB_CAP_L1SUB_CAP_HEADER_REG_0

Offset: 0x1ac
Reset: 0x1bc1001e (0b0001,1011,1100,0001,0000,0000,0001,1110)

Bit	Reset	Description
31:20	0x1bc	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h, if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. The bottom 2 bits of this offset are Reserved and must be implemented as 00b although software must mask them to allow for future uses of these bits. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. This field is depends on version of the specification. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

Bit	Reset	Description
15:0	0x1e	**EXTENDED_CAP_ID**: L1SUB Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. Extended Capability ID for L1 PM Substates is 001Eh. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X8_EP_PF0_L1SUB_CAP_L1SUB_CAP_HEADER_REG_0

Offset: 0x1c0
Reset: 0x1d01001e (0b0001,1101,0000,0001,0000,0000,0001,1110)

Bit	Reset	Description
31:20	0x1d0	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h, if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. The bottom 2 bits of this offset are Reserved and must be implemented as 00b although software must mask them to allow for future uses of these bits. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. This field is depends on version of the specification. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x1e	**EXTENDED_CAP_ID**: L1SUB Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. Extended Capability ID for L1 PM Substates is 001Eh. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_L1SUB_CAP_L1SUB_CAPABILITY_REG_0

where <*j*> = 4, 8.

Description:This register provides extended capability of L1 Substates.

PCIE_X4_EP_PF0_L1SUB_CAP_L1SUB_CAPABILITY_REG_0

Offset: 0x1b0

PCIE_X8_EP_PF0_L1SUB_CAP_L1SUB_CAPABILITY_REG_0

Offset: 0x1c4

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00a03c1f (0b0000,0000,1010,0000,0011,1100,0001,1111)

Bit	R/W	Reset	Description
31:24	RO	0x0	**RSVDP_24**: Reserved for future use.
23:19	RW	0x14	**PWR_ON_VALUE_SUPPORT**: Port T Power On Value. Along with the Port T_POWER_ON_SCALE field in the L1 PM Substates Capabilities register sets the time (in us) that this Port requires the port on the opposite side of Link to wait in L1.2.Exit after sampling CLKREQ# asserted before actively driving the interface. The value of Port T_POWER_ON is calculated by multiplying the value in this field by the scale value in the Port T_POWER_ON_SCALE field in the L1 PM Substates Capabilities register. Required for all Ports for which either the PCI-PM L1.2 Supported bit is Set, ASPM L1.2 Supported bit is Set, or both are Set, otherwise this field is of type RsvdP. Note: The access attributes of this field are as follows: - Wire: HWINIT - Dbi: L1SUB_CAPABILITY_REG.L1_2_ASPM_SUPPORT|| L1SUB_CAPABILITY_REG.L1_2_PCIPM_SUPPORT ? RWS : RSVDP
18	RO	0x0	**RSVDP_18**: Reserved for future use.
17:16	RW	0x0	**PWR_ON_SCALE_SUPPORT**: Port T Power On Scale. Specifies the scale used for the Port T_POWER_ON_VALUE field in the L1 PM Substates Capabilities register. Range of values are given below. Required for all Ports for which either the PCI-PM L1.2 Supported bit is Set, ASPM L1.2 Supported bit is Set, or both are Set, otherwise this field is of type RsvdP. Note: The access attributes of this field are as follows: - Wire: HWINIT - Dbi: L1SUB_CAPABILITY_REG.L1_2_ASPM_SUPPORT|| L1SUB_CAPABILITY_REG.L1_2_PCIPM_SUPPORT ? RWS : RSVDP
15:8	RW	0x3c	**COMM_MODE_SUPPORT**: Port Common Mode Restore Time. Time (in us) required for this Port to re-establish common mode. Required for all Ports for which either the PCI-PM L1.2 Supported bit is Set, ASPM L1.2 Supported bit is Set, or both are Set, otherwise this field is of type RsvdP. Note: The access attributes of this field are as follows: - Wire: HWINIT - Dbi: L1SUB_CAPABILITY_REG.L1_2_ASPM_SUPPORT|| L1SUB_CAPABILITY_REG.L1_2_PCIPM_SUPPORT ? RWS : RSVDP
7:5	RO	0x0	**RSVDP_5**: Reserved for future use.
4	RW	0x1	**L1_PMSUB_SUPPORT**: L1 PM Substates ECN Supported. When Set this field indicates that this Port supports L1 PM Substates. Note: The access attributes of this field are as follows: - Wire: HWINIT - Dbi: R/W (sticky)
3	RW	0x1	**L1_1_ASPM_SUPPORT**: ASPM L11 Supported. When Set this field indicates that ASPM L1.1 is supported. Note: The access attributes of this field are as follows: - Wire: HWINIT - Dbi: R/W (sticky)

Bit	R/W	Reset	Description
2	RW	0x1	**L1_2_ASPM_SUPPORT**: ASPM L12 Supported. When Set this field indicates that ASPM L1.2 is supported. Note: The access attributes of this field are as follows: - Wire: HWINIT - Dbi: R/W (sticky)
1	RW	0x1	**L1_1_PCIPM_SUPPORT**: PCI-PM L11 Supported. When Set this field indicates that PCI-PM L1.1 is supported, and must be Set by all Ports implementing L1 PM Substates. Note: The access attributes of this field are as follows: - Wire: HWINIT - Dbi: R/W (sticky)
0	RW	0x1	**L1_2_PCIPM_SUPPORT**: PCI-PM L12 Supported. When Set this field indicates that PCI-PM L1.2 is supported. Note: The access attributes of this field are as follows: - Wire: HWINIT - Dbi: R/W (sticky)

PCIE_X<*j*>_EP_PF0_L1SUB_CAP_L1SUB_CONTROL1_REG_0

where <*j*> = 4, 8.

Description:This register Controls that the individual extended capability is enabled ot not.

PCIE_X4_EP_PF0_L1SUB_CAP_L1SUB_CONTROL1_REG_0

Offset: 0x1b4

PCIE_X8_EP_PF0_L1SUB_CAP_L1SUB_CONTROL1_REG_0

Offset: 0x1c8

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:29	RW	0x0	**L1_2_TH_SCA**: LTR L12 Threshold Scale. This field provides a scale for the value contained within the LTR_L1.2_THRESHOLD_Value. Hardware operation is undefined if software writes a Not-Permitted value to this field. Required for all Ports Ports for which the ASPM L12 Supported bit is Set, otherwise this field is of type RsvdP. Note: The access attributes of this field are as follows: - Wire: L1SUB_CAPABILITY_REG.L1_2_ASPM_SUPPORT ? RW : RSVDP - Dbi: L1SUB_CAPABILITY_REG.L1_2_ASPM_SUPPORT ? RW : RSVDP
28:26	RO	0x0	**RSVDP_26**: Reserved for future use.

Bit	R/W	Reset	Description
25:16	RW	0x0	**L1_2_TH_VAL**: LTR L12 Threshold Value. Along with the LTR_L1.2_THRESHOLD_Scale, this field indicates the LTR threshold used to determine if entry into L1 results in L1.1 (if enabled) or L1.2 (if enabled). Required for all Ports for which the ASPM L12 Supported bit is Set, otherwise this field is of type RsvdP. Note: The access attributes of this field are as follows: - Wire: L1SUB_CAPABILITY_REG.L1_2_ASPM_SUPPORT ? RW : RSVDP - Dbi: L1SUB_CAPABILITY_REG.L1_2_ASPM_SUPPORT ? RW : RSVDP
15:8	RO	0x0	**T_COMMON_MODE**: Common Mode Restore Time. Sets value of TCOMMONMODE (in us), which must be used by the Downstream Ports for timing the re-establishment of common mode. This field is of type RsvdP for Upstream Ports.
7:4	RO	0x0	**RSVDP_4**: Reserved for future use.
3	RW	0x0	**L1_1_ASPM_EN**: ASPM L11 Enable. When Set this field, enables ASPM L1.1. For Ports for which the ASPM L1.1 Supported bit is clear, this bit is permitted to be hardwired to 0. Required for both Upstream and Downstream Ports. For compatibility with possible future extensions, software must not enable L1 PM Substates unless the L1 PM Substates Supported field in the L1 PM Substates Capabilities Register is Set.
2	RW	0x0	**L1_2_ASPM_EN**: ASPM L12 Enable. When Set this field, enables ASPM L1.2. For Ports for which the ASPM L1.2 Supported bit is clear, this bit is permitted to be hardwired to 0. Required for both Upstream and Downstream Ports. For compatibility with possible future extensions, software must not enable L1 PM Substates unless the L1 PM Substates Supported field in the L1 PM Substates Capabilities Register is Set.
1	RW	0x0	**L1_1_PCIPM_EN**: PCI-PM L11 Enable. When Set this field, enables PCI-PM L1.1. Required for both Upstream and Downstream Ports. For compatibility with possible future extensions, software must not enable L1 PM Substates unless the L1 PM Substates Supported field in the L1 PM Substates Capabilities Register is Set.
0	RW	0x0	**L1_2_PCIPM_EN**: PCI-PM L12 Enable. When Set this field, enables PCI-PM L1.2. For Ports for which the PCI-PM L1.2 Supported bit is clear, this bit is permitted to be hardwired to 0. Required for both Upstream and Downstream Ports. For compatibility with possible future extensions, software must not enable L1 PM Substates unless the L1 PM Substates Supported field in the L1 PM Substates Capabilities Register is Set.

PCIE_X<*j*>_EP_PF0_L1SUB_CAP_L1SUB_CONTROL2_REG_0

where <*j*> = 4, 8.

Description:This register Controls that the individual extended capability is enabled ot not.

PCIE_X4_EP_PF0_L1SUB_CAP_L1SUB_CONTROL2_REG_0

Offset: 0x1b8

PCIE_X8_EP_PF0_L1SUB_CAP_L1SUB_CONTROL2_REG_0

Offset: 0x1cc

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000028 (0b0000,0000,0000,0000,0000,0000,0010,1000)

Bit	R/W	Reset	Description
31:8	RO	0x0	**RSVDP_8**: Reserved for future use.
7:3	RW	0x5	**T_POWER_ON_VALUE**: T Power On Value. Along with the T_POWER_ON_SCALE sets the minimum amount of time (in us) that the Port must wait in L1.2.Exit after sampling CLKREQ# asserted before actively driving the interface. T_POWER_ON is calculated by multiplying the value in this field by the value in the T_POWER_ON_SCALE field. Required for all Ports that support L1.2, otherwise this field is of type RsvdP.This field must only be modified when the ASPM L1.2 Enable and PCI-PM L1.2 Enable bits are both Clear. The Port behavior is undefined if this field is modified when either the ASPM L1.2 Enable and/or PCI-PM L1.2 Enable bit(s) are set. Note: The access attributes of this field are as follows: - Wire: L1SUB_CAPABILITY_REG.L1_2_ASPM_SUPPORT|| L1SUB_CAPABILITY_REG.L1_2_PCIPM_SUPPORT ? RWS : RSVDP - Dbi: L1SUB_CAPABILITY_REG.L1_2_ASPM_SUPPORT|| L1SUB_CAPABILITY_REG.L1_2_PCIPM_SUPPORT ? RWS : RSVDP
2	RO	0x0	**RSVDP_2**: Reserved for future use.
1:0	RW	0x0	**T_POWER_ON_SCALE**: T Power On Scale. Specifies the scale used for T_POWER_ON_VALUE. Range of values are given below. Required for all Ports that support L1.2, otherwise this field is of type RsvdP. This field must only be modified when the ASPM L1.2 Enable and PCI-PM L1.2 Enable bits are both Clear. The Port behavior is undefined if this field is modified when either the ASPM L1.2 Enable and/or PCI-PM L1.2 Enable bit(s) are set. Note: The access attributes of this field are as follows: - Wire: L1SUB_CAPABILITY_REG.L1_2_ASPM_SUPPORT|| L1SUB_CAPABILITY_REG.L1_2_PCIPM_SUPPORT ? RWS : RSVDP - Dbi: L1SUB_CAPABILITY_REG.L1_2_ASPM_SUPPORT|| L1SUB_CAPABILITY_REG.L1_2_PCIPM_SUPPORT ? RWS : RSVDP

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_RAS_DES_CAP_HEADER_REG_0

where <*j*> = 4, 8.

Description:The Vendor-Specific Extended Capability (VSEC Capability) is an optional Extended Capability that is permitted to be implemented by any PCI Express Function or RCRB. This Register contains Capability Id, Capability Version and Next Offset value for Vendor-Specific Extended Capability.

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X4_EP_PF0_RAS_DES_CAP_RAS_DES_CAP_HEADER_REG_0

Offset: 0x1bc
Reset: 0x2bc1000b (0b0010,1011,1100,0001,0000,0000,0000,1011)

Bit	Reset	Description
31:20	0x2bc	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. Value of this field is depends on the version of the specification. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0xb	**EXTENDED_CAP_ID**: PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. Extended Capability ID for the Vendor-Specific Extended Capability is 000Bh. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X8_EP_PF0_RAS_DES_CAP_RAS_DES_CAP_HEADER_REG_0

Offset: 0x1d0
Reset: 0x2d01000b (0b0010,1101,0000,0001,0000,0000,0000,1011)

Bit	Reset	Description
31:20	0x2d0	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. Value of this field is depends on the version of the specification. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0xb	**EXTENDED_CAP_ID**: PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. Extended Capability ID for the Vendor-Specific Extended Capability is 000Bh. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_VENDOR_SPECIFIC_HEADER_REG_0

where <*j*> = 4, 8.

Description:This Register field provides VSEC Length, VSEC ID and VSEC Rev(Version Number). Vendor-specific software must qualify the associated Vendor ID of the PCI Express Function or RCRB before attempting to interpret the values in the VSEC ID or VSEC Rev fields.

PCIE_X4_EP_PF0_RAS_DES_CAP_VENDOR_SPECIFIC_HEADER_REG_0

Offset: 0x1c0

PCIE_X8_EP_PF0_RAS_DES_CAP_VENDOR_SPECIFIC_HEADER_REG_0

Offset: 0x1d4

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x10040002 (0b0001,0000,0000,0100,0000,0000,0000,0010)

Bit	Reset	Description
31:20	0x100	**VSEC_LENGTH**: VSEC Length. This field indicates the number of bytes in the entire VSEC structure, including the Vendor-Specific Extended Capability Header, the Vendor-Specific Header, and the vendor-specific registers.
19:16	0x4	**VSEC_REV**: VSEC Rev. This field is a vendor-defined version number that indicates the version of the VSEC structure. Software must qualify the Vendor ID and VSEC ID before interpreting this field.
15:0	0x2	**VSEC_ID**: VSEC ID. This field is a vendor-defined ID number that indicates the nature and format of the VSEC structure. Software must qualify the Vendor ID before interpreting this field.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_EVENT_COUNTER_CONTROL_REG_0

where <*j*> = 4, 8.

Description:This is a viewport control register.

- Setting the EVENT_COUNTER_EVENT_SELECT and EVENT_COUNTER_LANE_SELECT fields in this register determine the Event Counter data returned by the EVENT_COUNTER_DATA_REG viewport register.
- Setting the EVENT_COUNTER_ENABLE field in this register enables the Event Counter selected by the EVENT_COUNTER_EVENT_SELECT and EVENT_COUNTER_LANE_SELECT fields in this register.
- Setting the EVENT_COUNTER_CLEAR field in this register clears the Event Counter selected by the EVENT_COUNTER_EVENT_SELECT and EVENT_COUNTER_LANE_SELECT fields in this register.
- Reading the EVENT_COUNTER_STATUS field in this register returns the Enable status of the Event Counter selected by the EVENT_COUNTER_EVENT_SELECT and EVENT_COUNTER_LANE_SELECT fields in this register.

PCIE_X4_EP_PF0_RAS_DES_CAP_EVENT_COUNTER_CONTROL_REG_0

Offset: 0x1c4

PCIE_X8_EP_PF0_RAS_DES_CAP_EVENT_COUNTER_CONTROL_REG_0

Offset: 0x1d8

Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:28	RO	0x0	**RSVDP_28**: Reserved for future use.
27:16	RW	0x0	**EVENT_COUNTER_EVENT_SELECT**: Event Counter Data Select. This field in conjunction with the EVENT_COUNTER_LANE_SELECT field indexes the Event Counter data returned by the EVENT_COUNTER_DATA_REG register. - 27-24: Group number(4-bit: 0..0x7) - 23-16: Event number(8-bit: 0..0x13) within the Group - .. For detailed definitions of Group number and Event number, see the RAS DES chapter in the Databook. Note: This register field is sticky.
15:12	RO	0x0	**RSVDP_12**: Reserved for future use.
11:8	RW	0x0	**EVENT_COUNTER_LANE_SELECT**: Event Counter Lane Select. This field in conjunction with EVENT_COUNTER_EVENT_SELECT indexes the Event Counter data returned by the EVENT_COUNTER_DATA_REG register. Note: This register field is sticky.
7	RO	0x0	**EVENT_COUNTER_STATUS**: Event Counter Status. This register returns the current value of the Event Counter selected by the following fields: - EVENT_COUNTER_EVENT_SELECT - EVENT_COUNTER_LANE_SELECT Note: This register field is sticky.
6:5	RO	0x0	**RSVDP_5**: Reserved for future use.
4:2	WO	0x0	**EVENT_COUNTER_ENABLE**: Event Counter Enable. Enables/disables the Event Counter selected by the EVENT_COUNTER_EVENT_SELECT and EVENT_COUNTER_LANE_SELECT fields in this register. By default, all event counters are disabled. You can enable/disable a specific Event Counter by writing the 'per event off' or 'per event on' codes. You can enable/disable all event counters by writing the 'all on' or 'all off' codes. The read value is always '0'. For other values no change.
1:0	WO	0x0	**EVENT_COUNTER_CLEAR**: Event Counter Clear. Clears the Event Counter selected by the EVENT_COUNTER_EVENT_SELECT and EVENT_COUNTER_LANE_SELECT fields in this register. You can clear the value of a specific Event Counter by writing the 'per clear' code and you can clear all event counters at once by writing the 'all clear' code. The read value is always '0'. Other values are reserved.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_EVENT_COUNTER_DATA_REG_0

where <*j*> = 4, 8.

Description:This viewport register returns the data selected by the following fields:

- EVENT_COUNTER_EVENT_SELECT in EVENT_COUNTER_CONTROL_REG

- EVENT_COUNTER_LANE_SELECT in EVENT_COUNTER_CONTROL_REG.

For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X4_EP_PF0_RAS_DES_CAP_EVENT_COUNTER_DATA_REG_0

Offset: 0x1c8

PCIE_X8_EP_PF0_RAS_DES_CAP_EVENT_COUNTER_DATA_REG_0

Offset: 0x1dc

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EVENT_COUNTER_DATA**: Event Counter Data. This register returns the data selected by the following fields: - EVENT_COUNTER_EVENT_SELECT in EVENT_COUNTER_CONTROL_REG Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_TIME_BASED_ANALYSIS_CONTROL_REG_0

where <*j*> = 4, 8.

Description:Used for controlling the measurement of RX/TX data throughput and time spent in each low-power LTSSM state. For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X4_EP_PF0_RAS_DES_CAP_TIME_BASED_ANALYSIS_CONTROL_REG_0

Offset: 0x1cc

PCIE_X8_EP_PF0_RAS_DES_CAP_TIME_BASED_ANALYSIS_CONTROL_REG_0

Offset: 0x1e0

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000100 (0b0000,0000,0000,0000,0000,0001,0000,0000)

Bit	R/W	Reset	Description
31:24	RW	0x0	**TIME_BASED_REPORT_SELECT**: Time-based Report Select. Selects what type of data is measured for the selected duration (TIME_BASED_DURATION_SELECT), and returned in TIME_BASED_ANALYSIS_DATA. Each type of data is measured using one of three types of units: - Core_clk Cycles for 2.5GT/s, 5.0GT/s, 8.0GT/s, 16.0GT/s, 32GT/s. Total time in ps is [Value of TIME_BASED_ANALYSIS_DATA returned when TIME_BASED_REPORT_SELECT=0x00] * TIME_BASED_ANALYSIS_DATA. Values 0-4 and 7-8 correspond to Core_clk Cycles for 2.5GT/s, 5.0GT/s, 8.0GT/s, 16.0GT/s, 32GT/s. - Aux_clk Cycles. Total time in ps is [Period of platform specific clock] * TIME_BASED_ANALYSIS_DATA. Values 5, 6, and 9 correspond to aux_clk Cycles. - Core_clk Cycles for 20GT/s, 25GT/s (CCIX ESM data rate). Total time in ps is [Value of TIME_BASED_ANALYSIS_DATA returned when TIME_BASED_REPORT_SELECT=0x10] * TIME_BASED_ANALYSIS_DATA. Values 10-14 and 17-18 correspond to Core_clk Cycles for 2.5GT/s, 5.0GT/s, 8.0GT/s, 16.0GT/s, 32GT/s. - Data Bytes. Actual amount of bytes is 16 * TIME_BASED_ANALYSIS_DATA. Values 20-23 correspond to data bytes. All encodings other than the defined encodings are reserved. Note: This register field is sticky.
23:16	RO	0x0	**RSVDP_16**: Reserved for future use.
15:8	RW	0x1	**TIME_BASED_DURATION_SELECT**: Time-based Duration Select. Selects the duration of time-based analysis. When "manual control" is selected and TIMER_START is set to '1', this analysis never stops until TIMER_STOP is set to '0'. All encodings other than the defined encodings are reserved. Note: This register field is sticky.
7:1	RO	0x0	**RSVDP_1**: Reserved for future use.
0	RW	0x0	**TIMER_START**: Timer Start. This bit will be cleared automatically when the measurement is finished. Note: The app_ras_des_tba_ctrl input also sets the contents of this field and controls the measurement start/stop. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_TIME_BASED_ANALYSIS_DATA_REG_0

where <*j*> = 4, 8.

Description:Contains the measurement results of RX/TX data throughput and time spent in each low-power LTSSM state. This viewport register returns the data selected by the TIME_BASED_REPORT_SELECT field in TIME_BASED_ANALYSIS_CONTROL_REG. For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X4_EP_PF0_RAS_DES_CAP_TIME_BASED_ANALYSIS_DATA_REG_0

Offset: 0x1d0

PCIE_X8_EP_PF0_RAS_DES_CAP_TIME_BASED_ANALYSIS_DATA_REG_0

Offset: 0x1e4

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TIME_BASED_ANALYSIS_DATA**: Time Based Analysis Data. This register returns the data selected by the TIME_BASED_REPORT_SELECT field in TIME_BASED_ANALYSIS_CONTROL_REG. The results are cleared when next measurement starts. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_TIME_BASED_ANALYSIS_DATA_63_32_REG_0

where <*j*> = 4, 8.

Description:This viewport register returns the data selected by the TIME_BASED_REPORT_SELECT field in TIME_BASED_ANALYSIS_CONTROL_REG. For more details, see the "Reliability, Availability, and Serviceability (RAS)" section in the "Controller Operations" chapter of the Databook.

PCIE_X4_EP_PF0_RAS_DES_CAP_TIME_BASED_ANALYSIS_DATA_63_32_REG_0

Offset: 0x1d4

PCIE_X8_EP_PF0_RAS_DES_CAP_TIME_BASED_ANALYSIS_DATA_63_32_REG_0

Offset: 0x1e8

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TIME_BASED_ANALYSIS_DATA_63_32**: Upper 32 bits of Time Based Analysis Data. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_EINJ_ENABLE_REG_0

where <*j*> = 4, 8.

Description:Each type of error insertion is enabled by the corresponding bit in this register. The specific injection controls for each type of error are defined in the following registers: - 0: CRC Error: EINJ0_CRC_REG - 1: Sequence Number Error: EINJ1_SEQNUM_REG - 2: DLLP Error: EINJ2_DLLP_REG - 3: Symbol DataK Mask Error or Sync Header Error: EINJ3_SYMBOL_REG - 4: FC Credit Update Error: EINJ4_FC_REG - 5: TLP Duplicate/Nullify Error: EINJ5_SP_TLP_REG - 6: Specific TLP Error: EINJ6_COMPARE_*_REG/EINJ6_CHANGE_*_REG/EINJ6_TLP_REG After the errors have been inserted by controller, it will clear each bit here. For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X4_EP_PF0_RAS_DES_CAP_EINJ_ENABLE_REG_0

Offset: 0x1ec

PCIE_X8_EP_PF0_RAS_DES_CAP_EINJ_ENABLE_REG_0

Offset: 0x200

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:7	RO	0x0	**RSVDP_7**: Reserved for future use.
6	RW	0x0	**ERROR_INJECTION6_ENABLE**: Error Injection6 Enable (Specific TLP Error). Enables insertion of errors into the packets that you select. You can set this bit to '1' when you have disabled RAS datapath protection (DP) by setting CX_RASDP = CX_RASDP_RAM_PROT =0. You can set this bit to '1' when you have disabled the address translation by setting ADDR_TRANSLATION_SUPPORT_EN=0. For more details, see the EINJ6_COMPARE_*_REG/EINJ6_CHANGE_*_REG/EINJ6_TLP_REG registers. Note: This register field is sticky.
5	RW	0x0	**ERROR_INJECTION5_ENABLE**: Error Injection5 Enable (TLP Duplicate/Nullify Error). Enables insertion of duplicate/nullified TLPs. For more details, see the EINJ5_SP_TLP_REG register. Note: This register field is sticky.
4	RW	0x0	**ERROR_INJECTION4_ENABLE**: Error Injection4 Enable (FC Credit Update Error). Enables insertion of errors into UpdateFCs. For more details, see the EINJ4_FC_REG register. Note: This register field is sticky.
3	RW	0x0	**ERROR_INJECTION3_ENABLE**: Error Injection3 Enable (Symbol DataK Mask Error or Sync Header Error). Enables DataK masking of special symbols or the breaking of the sync header. For more details, see the EINJ3_SYMBOL_REG register. Note: This register field is sticky.

Bit	R/W	Reset	Description
2	RW	0x0	**ERROR_INJECTION2_ENABLE**: Error Injection2 Enable (DLLP Error). Enables insertion of DLLP errors. For more details, see the EINJ2_DLLP_REG register. Note: This register field is sticky.
1	RW	0x0	**ERROR_INJECTION1_ENABLE**: Error Injection1 Enable (Sequence Number Error). Enables insertion of errors into sequence numbers. For more details, see the EINJ1_SEQNUM_REG register. Note: This register field is sticky.
0	RW	0x0	**ERROR_INJECTION0_ENABLE**: Error Injection0 Enable (CRC Error). Enables insertion of errors into various CRC. For more details, see the EINJ0_CRC_REG register. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_EINJ0_CRC_REG_0

where <*j*> = 4, 8.

Description:Controls the insertion of errors into the CRC, and parity of ordered sets for the selected type of the packets as follows:

- LCRC. Bad TLP will be detected at the receiver side; receiver responds with NAK DLLP; Data Link Retry starts.
- 16-bit CRC of ACK/NAK DLLPs. Bad DLLP occurs at the receiver side; Replay NUM Rollover occurs.
- 16-bit CRC of UpdateFC DLLPs. Error insertion continues for the specific time; LTSSM transitions to the Recovery state because of the UpdateFC timeout (if the timeout is implemented at the receiver of the UpdateFCs).
- ECRC. If ECRC check is enabled, ECRC error is detected at the receiver side.
- FCRC. Framing error will be detected, TLP is discarded, and the LTSSM transitions to Recovery state.
- Parity of TSOS. Error insertion continues for the specific time; LTSSM Recovery/Configuration timeout will occur.
- Parity of SKPOS. Lane error will be detected at the receiver side.

PCIE_X4_EP_PF0_RAS_DES_CAP_EINJ0_CRC_REG_0

Offset: 0x1f0

PCIE_X8_EP_PF0_RAS_DES_CAP_EINJ0_CRC_REG_0

Offset: 0x204

Read/Write: See table below
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:12	RO	0x0	**RSVDP_12**: Reserved for future use.
11:8	RW	0x0	**EINJ0_CRC_TYPE**: Error injection type. Selects the type of CRC error to be inserted. All encodings other than the defined encodings are reserved. Note: This register field is sticky.
7:0	RW	0x0	**EINJ0_COUNT**: Error injection count. Indicates the number of errors. This register is decremented when the errors have been inserted. - If the counter value is 0x01 and error is inserted, ERROR_INJECTION0_ENABLE in EINJ_ENABLE_REG returns 0b. - If the counter value is 0x00 and ERROR_INJECTION0_ENABLE=1, the errors are inserted until ERROR_INJECTION0_ENABLE is set to '0'. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_EINJ1_SEQNUM_REG_0

where <*j*> = 4, 8.

Description:Controls the sequence number of the specific TLPs and ACK/NAK DLLPs.
Data Link Protocol Error will be detected at the Rx side of ACK/NAL DLLPs when one of these conditions is true:

- (((NEXT_TRANSMIT_SEQ -1) - AckNak_Seq_Num) mod 4096 > 2048)
- ((AckNak_Seq_Num - ACKD_SEQ) mod 4096 >= 2048)

TLP is treated as Duplicate TLP at the Rx side when all these conditions are true:

- (Sequence Number != NEXT_RCV_SEQ)
- ((NEXT_RCV_SEQ - Sequence Number) mod 4096 <= 2048)

TLP is treated as Bad TLP at the Rx side when all these conditions are true:

- (Sequence Number != NEXT_RCV_SEQ) and
- ((NEXT_RCV_SEQ - Sequence Number) mod 4096 > 2048)

PCIE_X4_EP_PF0_RAS_DES_CAP_EINJ1_SEQNUM_REG_0

Offset: 0x1f4

PCIE_X8_EP_PF0_RAS_DES_CAP_EINJ1_SEQNUM_REG_0

Offset: 0x208

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:29	RO	0x0	**RSVDP_29**: Reserved for future use.
28:16	RW	0x0	**EINJ1_BAD_SEQNUM**: Bad sequence number. Indicates the value to add/subtract from the naturally-assigned sequence numbers. This value is represented by two's complement. For example: - Set Type, SEQ# and Count -- EINJ1_SEQNUM_TYPE =0 (Insert errors into new TLPs) -- EINJ1_BAD_SEQNUM =0x1FFD (represents -3) -- EINJ1_COUNT =1 - Enable Error Injection -- ERROR_INJECTION1_ENABLE =1 - Send a TLP From the Core's Application Interface -- Assume SEQ#5 is given to the TLP. - The SEQ# is Changed to #2 by the Error Injection Function in Layer2. -- 5 + (-3) = 2 - The TLP with SEQ#2 is Transmitted to PCIe Link. Note: This register field is sticky.
15:9	RO	0x0	**RSVDP_9**: Reserved for future use.
8	RW	0x0	**EINJ1_SEQNUM_TYPE**: Sequence number type. Selects the type of sequence number. Note: This register field is sticky.
7:0	RW	0x0	**EINJ1_COUNT**: Error injection count. Indicates the number of errors. This register is decremented as the errors are being inserted. - If the counter value is 0x01 and error is inserted, ERROR_INJECTION1_ENABLE in EINJ_ENABLE_REG returns '0'. - If the counter value is 0x00 and ERROR_INJECTION1_ENABLE=1, the errors are inserted until ERROR_INJECTION1_ENABLE is set to '0'. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_EINJ2_DLLP_REG_0

where <*j*> = 4, 8.

Description:Controls the transmission of DLLPs and inserts the following errors: - If "ACK/NAK DLLP's transmission block" is selected, replay timeout error will occur at the transmitter of the TLPs and then Data Link Retry will occur. - If "Update FC DLLP's transmission block" is selected, LTSSM will transition to the Recovery state because of the UpdateFC timeout (if the timeout is implemented at the receiver of the UpdateFCs). - If "Always Transmission for NAK DLLP" is selected, Data Link Retry will occur at the transmitter of the TLPs. Furthermore, Replay NUM Rollover will occur when the transmitter has been requested four times to send the TLP with the same sequence number.

PCIE_X4_EP_PF0_RAS_DES_CAP_EINJ2_DLLP_REG_0

Offset: 0x1f8

PCIE_X8_EP_PF0_RAS_DES_CAP_EINJ2_DLLP_REG_0

Offset: 0x20c

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:10	RO	0x0	**RSVDP_10**: Reserved for future use.
9:8	RW	0x0	**EINJ2_DLLP_TYPE**: DLLP Type. Selects the type of DLLP errors to be inserted. Note: This register field is sticky.
7:0	RW	0x0	**EINJ2_COUNT**: Error injection count. Indicates the number of errors. This register is decremented as the errors are being inserted. - If the counter value is 0x01 and the error is inserted, ERROR_INJECTION2_ENABLE in EINJ_ENABLE_REG returns '0'. - If the counter value is 0x00 and ERROR_INJECTION2_ENABLE =1, the errors are inserted until ERROR_INJECTION2_ENABLE is set to '0'. This register is affected only when EINJ2_DLLP_TYPE =2'10b. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_EINJ3_SYMBOL_REG_0

where <*j*> = 4, 8.

Description:When 8b/10b encoding is used, this register controls error insertion into the special (K code) symbols. - If TS1/TS2/FTS/E-Idle/SKP is selected, it affects whole of the ordered set. It might cause timeout of the LTSSM. - If END/EDB/STP/SDP is selected, TLP/DLLP will be discarded at the receiver side. When 128b/130b encoding is used, this register controls error insertion into the sync-header.

PCIE_X4_EP_PF0_RAS_DES_CAP_EINJ3_SYMBOL_REG_0

Offset: 0x1fc

PCIE_X8_EP_PF0_RAS_DES_CAP_EINJ3_SYMBOL_REG_0

Offset: 0x210

Read/Write: See table below
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:11	RO	0x0	**RSVDP_11**: Reserved for future use.
10:8	RW	0x0	**EINJ3_SYMBOL_TYPE**: Error Type. 8b/10b encoding - Mask K symbol. It is not supported to insert errors into the first ordered-set after exiting from TxElecIdle when CX_FREQ_STEP_EN has been enabled. All encodings other than the defined encodings are reserved. Note: This register field is sticky.
7:0	RW	0x0	**EINJ3_COUNT**: Error injection count. Indicates the number of errors. This register is decremented as the errors are being inserted. - If the counter value is 0x01 and error is inserted, ERROR_INJECTION3_ENABLE in EINJ_ENABLE_REG returns '0'. - If the counter value is 0x00 and ERROR_INJECTION3_ENABLE =1, the errors are inserted until ERROR_INJECTION3_ENABLE is set to '0'. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_EINJ4_FC_REG_0

where <*j*> = 4, 8.

Description:Controls error insertion into the credit value in the UpdateFCs. It is possible to insert errors for any of the following types: - Posted TLP Header credit - Non-Posted TLP Header credit - Completion TLP Header credit - Posted TLP Data credit - Non-Posted TLP Data credit - Completion TLP Data credit These errors are not correctable while error insertion is enabled. Receiver buffer overflow error might occur.

PCIE_X4_EP_PF0_RAS_DES_CAP_EINJ4_FC_REG_0

Offset: 0x200

PCIE_X8_EP_PF0_RAS_DES_CAP_EINJ4_FC_REG_0

Offset: 0x214

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:29	RO	0x0	**RSVDP_29**: Reserved for future use.
28:16	RW	0x0	**EINJ4_BAD_UPDFC_VALUE**: Bad update-FC credit value. Indicates the value to add/subtract from the UpdateFC credit. This value is represented by two's complement. Note: This register field is sticky.
15	RO	0x0	**RSVDP_15**: Reserved for future use.
14:12	RW	0x0	**EINJ4_VC_NUMBER**: VC Number. Indicates target VC Number. Note: This register field is sticky.
11	RO	0x0	**RSVDP_11**: Reserved for future use.
10:8	RW	0x0	**EINJ4_UPDFC_TYPE**: Update-FC type. Selects the credit type. Note: This register field is sticky.
7:0	RW	0x0	**EINJ4_COUNT**: Error injection count. Indicates the number of errors. This register is decremented as the errors are being inserted. - If the counter value is 0x01 and error is inserted, ERROR_INJECTION4_ENABLE in EINJ_ENABLE_REG returns '0'. - If the counter value is 0x00 and ERROR_INJECTION4_ENABLE =1, the errors are inserted until ERROR_INJECTION4_ENABLE is set to '0'. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_EINJ5_SP_TLP_REG_0

where <*j*> = 4, 8.

Description:Controls the generation of specified TLPs. Correctable errors will occur which will be fixed by the PCIe protocol. - For Duplicate TLP, the controller initiates Data Link Retry by handling ACK DLLP as NAK DLLP. These TLPs will be duplicate TLPs at the receiver side. - For Nullified TLP, the TLPs that the controller transmits are changed into nullified TLPs and the original TLPs are stored in the retry buffer. The receiver of these TLPs will detect the lack of seq# and send NAK DLLP at the next TLP. Then the original TLPs are sent from retry buffer and the data controls are recovered. For 128 bit controller or more than 128 bit, the controller inserts errors the number of times of EINJ5_COUNT but doesn't ensure that the errors are continuously inserted into TLPs.

PCIE_X4_EP_PF0_RAS_DES_CAP_EINJ5_SP_TLP_REG_0

Offset: 0x204

PCIE_X8_EP_PF0_RAS_DES_CAP_EINJ5_SP_TLP_REG_0

Offset: 0x218

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:9	RO	0x0	**RSVDP_9**: Reserved for future use.
8	RW	0x0	**EINJ5_SPECIFIED_TLP**: Specified TLP. Selects the specified TLP to be inserted. Note: This register field is sticky.
7:0	RW	0x0	**EINJ5_COUNT**: Error injection count. Indicates the number of errors. This register is decremented as the errors are being inserted. - If the counter value is 0x01 and error is inserted, ERROR_INJECTION5_ENABLE in EINJ_ENABLE_REG returns '0'. - If the counter value is 0x00 and ERROR_INJECTION5_ENABLE =1, the errors are inserted until ERROR_INJECTION5_ENABLE is set to '0'. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_EINJ6_COMPARE_POINT_H0_REG_0

where <*j*> = 4, 8.

Description:Program this register for the 1st DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW0[7:0], TLP_DW0[15:8], TLP_DW0[23:16], TLP_DW0[31:24] The Packet Compare Point registers (EINJ6_COMPARE_POINT*) specify which Tx TLP header bits to compare with the corresponding bits in the Packet Compare Value registers (EINJ6_COMPARE_VALUE*). When all specified bits (in the Tx TLP header and EINJ6_COMPARE_VALUE*) match, the controller inserts errors into the TLP. The type and number of errors are specified by the EINJ6_TLP_REG register.

PCIE_X4_EP_PF0_RAS_DES_CAP_EINJ6_COMPARE_POINT_H0_REG_0

Offset: 0x208

PCIE_X8_EP_PF0_RAS_DES_CAP_EINJ6_COMPARE_POINT_H0_REG_0

Offset: 0x21c

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_COMPARE_POINT_H0**: Packet Compare Point: 1st DWORD. Specifies which Tx TLP header DWORD#0 bits to compare with the corresponding bits in the Packet Compare Value registers (EINJ6_COMPARE_VALUE*). When all specified bits (in the Tx TLP header and EINJ6_COMPARE_VALUE*) match, the controller inserts errors into the TLP. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_EINJ6_COMPARE_POINT_H1_REG_0

where <*j*> = 4, 8.

Description:Program this register for the 2nd DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW1[7:0], TLP_DW1[15:8], TLP_DW1[23:16], TLP_DW1[31:24] The Packet Compare Point registers (EINJ6_COMPARE_POINT*) specify which Tx TLP header bits to compare with the corresponding bits in the Packet Compare Value registers (EINJ6_COMPARE_VALUE*). When all specified bits (in the Tx TLP header and EINJ6_COMPARE_VALUE*) match, the controller inserts errors into the TLP. The type and number of errors are specified by the EINJ6_TLP_REG register.

PCIE_X4_EP_PF0_RAS_DES_CAP_EINJ6_COMPARE_POINT_H1_REG_0

Offset: 0x20c

PCIE_X8_EP_PF0_RAS_DES_CAP_EINJ6_COMPARE_POINT_H1_REG_0

Offset: 0x220

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_COMPARE_POINT_H1**: Packet Compare Point: 2nd DWORD. Specifies which Tx TLP header DWORD#1 bits to compare with the corresponding bits in the Packet Compare Value registers (EINJ6_COMPARE_VALUE*). When all specified bits (in the Tx TLP header and EINJ6_COMPARE_VALUE*) match, the controller inserts errors into the TLP. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_EINJ6_COMPARE_POINT_H2_REG_0

where <*j*> = 4, 8.

Description:Program this register for the 3rd DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW2[7:0], TLP_DW2[15:8], TLP_DW2[23:16], TLP_DW2[31:24] The Packet Compare Point registers (EINJ6_COMPARE_POINT*) specify which Tx TLP header bits to compare with the corresponding bits in the Packet Compare Value registers (EINJ6_COMPARE_VALUE*). When all specified bits (in the Tx TLP header and EINJ6_COMPARE_VALUE*) match, the controller inserts errors into the TLP. The type and number of errors are specified by the EINJ6_TLP_REG register.

PCIE_X4_EP_PF0_RAS_DES_CAP_EINJ6_COMPARE_POINT_H2_REG_0

Offset: 0x210

PCIE_X8_EP_PF0_RAS_DES_CAP_EINJ6_COMPARE_POINT_H2_REG_0

Offset: 0x224

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_COMPARE_POINT_H2**: Packet Compare Point: 3rd DWORD. Specifies which Tx TLP header DWORD#2 bits to compare with the corresponding bits in the Packet Compare Value registers (EINJ6_COMPARE_VALUE*). When all specified bits (in the Tx TLP header and EINJ6_COMPARE_VALUE*) match, the controller inserts errors into the TLP. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_EINJ6_COMPARE_POINT_H3_REG_0

where <*j*> = 4, 8.

Description:Program this register for the 4th DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW3[7:0], TLP_DW3[15:8], TLP_DW3[23:16], TLP_DW3[31:24] The Packet Compare Point registers (EINJ6_COMPARE_POINT*) specify which Tx TLP header bits to compare with the corresponding bits in the Packet Compare Value registers (EINJ6_COMPARE_VALUE*). When all specified bits (in the Tx TLP header and EINJ6_COMPARE_VALUE*) match, the controller inserts errors into the TLP. The type and number of errors are specified by the EINJ6_TLP_REG register.

PCIE_X4_EP_PF0_RAS_DES_CAP_EINJ6_COMPARE_POINT_H3_REG_0

Offset: 0x214

PCIE_X8_EP_PF0_RAS_DES_CAP_EINJ6_COMPARE_POINT_H3_REG_0

Offset: 0x228

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_COMPARE_POINT_H3**: Packet Compare Point: 4th DWORD. Specifies which Tx TLP header DWORD#3 bits to compare with the corresponding bits in the Packet Compare Value registers (EINJ6_COMPARE_VALUE*). When all specified bits (in the Tx TLP header and EINJ6_COMPARE_VALUE*) match, the controller inserts errors into the TLP. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_EINJ6_COMPARE_VALUE_H0_REG_0

where <*j*> = 4, 8.

Description:Program this register for the 1st DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW0[7:0], TLP_DW0[15:8], TLP_DW0[23:16], TLP_DW0[31:24] The Packet Compare Point registers (EINJ6_COMPARE_POINT*) specify which Tx TLP header bits to compare with the corresponding bits in the Packet Compare Value registers (EINJ6_COMPARE_VALUE*). When all specified bits (in the Tx TLP header and EINJ6_COMPARE_VALUE*) match, the controller inserts errors into the TLP. The type and number of errors are specified by the EINJ6_TLP_REG register.

PCIE_X4_EP_PF0_RAS_DES_CAP_EINJ6_COMPARE_VALUE_H0_REG_0

Offset: 0x218

PCIE_X8_EP_PF0_RAS_DES_CAP_EINJ6_COMPARE_VALUE_H0_REG_0

Offset: 0x22c

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_COMPARE_VALUE_H0**: Packet Compare Value: 1st DWORD. Specifies the value to compare against Tx the TLP header DWORD#0 bits specified in the Packet Compare Point registers (EINJ6_COMPARE_POINT*). Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_EINJ6_COMPARE_VALUE_H1_REG_0

where <*j*> = 4, 8.

Description:Program this register for the 2nd DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW1[7:0], TLP_DW1[15:8], TLP_DW1[23:16], TLP_DW1[31:24] The Packet Compare Point registers (EINJ6_COMPARE_POINT*) specify which Tx TLP header bits to compare with the corresponding bits in the Packet Compare Value registers (EINJ6_COMPARE_VALUE*). When all specified bits (in the Tx TLP header and EINJ6_COMPARE_VALUE*) match, the controller inserts errors into the TLP. The type and number of errors are specified by the EINJ6_TLP_REG register.

PCIE_X4_EP_PF0_RAS_DES_CAP_EINJ6_COMPARE_VALUE_H1_REG_0

Offset: 0x21c

PCIE_X8_EP_PF0_RAS_DES_CAP_EINJ6_COMPARE_VALUE_H1_REG_0

Offset: 0x230

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_COMPARE_VALUE_H1**: Packet Compare Value: 2nd DWORD. Specifies the value to compare against Tx the TLP header DWORD#1 bits specified in the Packet Compare Point registers (EINJ6_COMPARE_POINT*). Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_EINJ6_COMPARE_VALUE_H2_REG_0

where <*j*> = 4, 8.

Description:Program this register for the 3rd DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW2[7:0], TLP_DW2[15:8], TLP_DW2[23:16], TLP_DW2[31:24] The Packet Compare Point registers (EINJ6_COMPARE_POINT*) specify which Tx TLP header bits to compare with the corresponding bits in the Packet Compare Value registers (EINJ6_COMPARE_VALUE*). When all specified bits (in the Tx TLP header and EINJ6_COMPARE_VALUE*) match, the controller inserts errors into the TLP. The type and number of errors are specified by the EINJ6_TLP_REG register.

PCIE_X4_EP_PF0_RAS_DES_CAP_EINJ6_COMPARE_VALUE_H2_REG_0

Offset: 0x220

PCIE_X8_EP_PF0_RAS_DES_CAP_EINJ6_COMPARE_VALUE_H2_REG_0

Offset: 0x234

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_COMPARE_VALUE_H2**: Packet Compare Value: 3rd DWORD. Specifies the value to compare against Tx the TLP header DWORD#2 bits specified in the Packet Compare Point registers (EINJ6_COMPARE_POINT*). Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_EINJ6_COMPARE_VALUE_H3_REG_0

where <*j*> = 4, 8.

Description:Program this register for the 4th DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW3[7:0], TLP_DW3[15:8], TLP_DW3[23:16], TLP_DW3[31:24] The Packet Compare Point registers (EINJ6_COMPARE_POINT*) specify which Tx TLP header bits to compare with the corresponding bits in the Packet Compare Value registers (EINJ6_COMPARE_VALUE*). When all specified bits (in the Tx TLP header and EINJ6_COMPARE_VALUE*) match, the controller inserts errors into the TLP. The type and number of errors are specified by the EINJ6_TLP_REG register.

PCIE_X4_EP_PF0_RAS_DES_CAP_EINJ6_COMPARE_VALUE_H3_REG_0

Offset: 0x224

PCIE_X8_EP_PF0_RAS_DES_CAP_EINJ6_COMPARE_VALUE_H3_REG_0

Offset: 0x238

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_COMPARE_VALUE_H3**: Packet Compare Value: 4th DWORD. Specifies the value to compare against Tx the TLP header DWORD#3 bits specified in the Packet Compare Point registers (EINJ6_COMPARE_POINT*). Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_EINJ6_CHANGE_POINT_H0_REG_0

where <*j*> = 4, 8.

Description:Program this register for the 1st DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW0[7:0], TLP_DW0[15:8], TLP_DW0[23:16], TLP_DW0[31:24] The Packet Change Point registers (EINJ6_CHANGE_POINT*) specify which Tx TLP header bits to replace with the corresponding bits in the Packet Change Value registers (EINJ6_CHANGE_VALUE*). The type and number of errors are specified by the EINJ6_TLP_REG register. Only applies when EINJ6_INVERTED_CONTROL in EINJ6_TLP_REG =0.

PCIE_X4_EP_PF0_RAS_DES_CAP_EINJ6_CHANGE_POINT_H0_REG_0

Offset: 0x228

PCIE_X8_EP_PF0_RAS_DES_CAP_EINJ6_CHANGE_POINT_H0_REG_0

Offset: 0x23c

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_CHANGE_POINT_H0**: Packet Change Point: 1st DWORD. Specifies which Tx TLP header DWORD#0 bits to replace with the corresponding bits in the Packet Change Value registers (EINJ6_CHANGE_VALUE*). Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_EINJ6_CHANGE_POINT_H1_REG_0

where <*j*> = 4, 8.

Description:Program this register for the 2nd DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW1[7:0], TLP_DW1[15:8], TLP_DW1[23:16], TLP_DW1[31:24] The Packet Change Point registers (EINJ6_CHANGE_POINT*) specify which Tx TLP header bits to replace with the corresponding bits in the Packet Change Value registers (EINJ6_CHANGE_VALUE*). The type and number of errors are specified by the EINJ6_TLP_REG register. Only applies when EINJ6_INVERTED_CONTROL in EINJ6_TLP_REG =0.

PCIE_X4_EP_PF0_RAS_DES_CAP_EINJ6_CHANGE_POINT_H1_REG_0

Offset: 0x22c

PCIE_X8_EP_PF0_RAS_DES_CAP_EINJ6_CHANGE_POINT_H1_REG_0

Offset: 0x240

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_CHANGE_POINT_H1**: Packet Change Point: 2nd DWORD. Specifies which Tx TLP header DWORD#1 bits to replace with the corresponding bits in the Packet Change Value registers (EINJ6_CHANGE_VALUE*). Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_EINJ6_CHANGE_POINT_H2_REG_0

where <*j*> = 4, 8.

Description:Program this register for the 3rd DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW2[7:0], TLP_DW2[15:8], TLP_DW2[23:16], TLP_DW2[31:24] The Packet Change Point registers (EINJ6_CHANGE_POINT*) specify which Tx TLP header bits to replace with the corresponding bits in the Packet Change Value registers (EINJ6_CHANGE_VALUE*). The type and number of errors are specified by the EINJ6_TLP_REG register. Only applies when EINJ6_INVERTED_CONTROL in EINJ6_TLP_REG =0.

PCIE_X4_EP_PF0_RAS_DES_CAP_EINJ6_CHANGE_POINT_H2_REG_0

Offset: 0x230

PCIE_X8_EP_PF0_RAS_DES_CAP_EINJ6_CHANGE_POINT_H2_REG_0

Offset: 0x244

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_CHANGE_POINT_H2**: Packet Change Point: 3rd DWORD. Specifies which Tx TLP header DWORD#2 bits to replace with the corresponding bits in the Packet Change Value registers (EINJ6_CHANGE_VALUE*). Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_EINJ6_CHANGE_POINT_H3_REG_0

where <*j*> = 4, 8.

Description:Program this register for the 4th DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW3[7:0], TLP_DW3[15:8], TLP_DW3[23:16], TLP_DW3[31:24] The Packet Change Point registers (EINJ6_CHANGE_POINT*) specify which Tx TLP header bits to replace with the corresponding bits in the Packet Change Value registers (EINJ6_CHANGE_VALUE*). The type and number of errors are specified by the EINJ6_TLP_REG register. Only applies when EINJ6_INVERTED_CONTROL in EINJ6_TLP_REG =0.

PCIE_X4_EP_PF0_RAS_DES_CAP_EINJ6_CHANGE_POINT_H3_REG_0

Offset: 0x234

PCIE_X8_EP_PF0_RAS_DES_CAP_EINJ6_CHANGE_POINT_H3_REG_0

Offset: 0x248

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_CHANGE_POINT_H3**: Packet Change Point: 4th DWORD. Specifies which Tx TLP header DWORD#3 bits to replace with the corresponding bits in the Packet Change Value registers (EINJ6_CHANGE_VALUE*). Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_EINJ6_CHANGE_VALUE_H0_REG_0

where <*j*> = 4, 8.

Description:Program this register for the 1st DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW0[7:0], TLP_DW0[15:8], TLP_DW0[23:16], TLP_DW0[31:24] The Packet Change Point registers (EINJ6_CHANGE_POINT*) specify which Tx TLP header bits to replace with the corresponding bits in the Packet Change Value registers (EINJ6_CHANGE_VALUE*). The type and number of errors are specified by the EINJ6_TLP_REG register. Only applies when EINJ6_INVERTED_CONTROL in EINJ6_TLP_REG =0.

PCIE_X4_EP_PF0_RAS_DES_CAP_EINJ6_CHANGE_VALUE_H0_REG_0

Offset: 0x238

PCIE_X8_EP_PF0_RAS_DES_CAP_EINJ6_CHANGE_VALUE_H0_REG_0

Offset: 0x24c

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_CHANGE_VALUE_H0**: Packet Change Value: 1st DWORD. Specifies replacement values for the Tx TLP header DWORD#0 bits defined in the Packet Change Point registers (EINJ6_CHANGE_POINT*). Only applies when the EINJ6_INVERTED_CONTROL field in EINJ6_TLP_REG is '0'. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_EINJ6_CHANGE_VALUE_H1_REG_0

where <*j*> = 4, 8.

Description:Program this register for the 2nd DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW1[7:0], TLP_DW1[15:8], TLP_DW1[23:16], TLP_DW1[31:24] The Packet Change Point registers (EINJ6_CHANGE_POINT*) specify which Tx TLP header bits to replace with the corresponding bits in the Packet Change Value registers (EINJ6_CHANGE_VALUE*). The type and number of errors are

specified by the EINJ6_TLP_REG register. Only applies when EINJ6_INVERTED_CONTROL in EINJ6_TLP_REG =0.

PCIE_X4_EP_PF0_RAS_DES_CAP_EINJ6_CHANGE_VALUE_H1_REG_0

Offset: 0x23c

PCIE_X8_EP_PF0_RAS_DES_CAP_EINJ6_CHANGE_VALUE_H1_REG_0

Offset: 0x250

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_CHANGE_VALUE_H1**: Packet Change Value: 2nd DWORD. Specifies replacement values for the Tx TLP header DWORD#1 bits defined in the Packet Change Point registers (EINJ6_CHANGE_POINT*). Only applies when the EINJ6_INVERTED_CONTROL field in EINJ6_TLP_REG is '0'. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_EINJ6_CHANGE_VALUE_H2_REG_0

where <*j*> = 4, 8.

Description:Program this register for the 3rd DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW2[7:0], TLP_DW2[15:8], TLP_DW2[23:16], TLP_DW2[31:24] The Packet Change Point registers (EINJ6_CHANGE_POINT*) specify which Tx TLP header bits to replace with the corresponding bits in the Packet Change Value registers (EINJ6_CHANGE_VALUE*). The type and number of errors are specified by the EINJ6_TLP_REG register. Only applies when EINJ6_INVERTED_CONTROL in EINJ6_TLP_REG =0.

PCIE_X4_EP_PF0_RAS_DES_CAP_EINJ6_CHANGE_VALUE_H2_REG_0

Offset: 0x240

PCIE_X8_EP_PF0_RAS_DES_CAP_EINJ6_CHANGE_VALUE_H2_REG_0

Offset: 0x254

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_CHANGE_VALUE_H2**: Packet Change Value: 3rd DWORD. Specifies replacement values for the Tx TLP header DWORD#2 bits defined in the Packet Change Point registers (EINJ6_CHANGE_POINT*). Only applies when the EINJ6_INVERTED_CONTROL field in EINJ6_TLP_REG is '0'. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_EINJ6_CHANGE_VALUE_H3_REG_0

where <*j*> = 4, 8.

Description:Program this register for the 4th DWORD of TLP header/prefix. It is necessary to carefully consider the endianness when you program this register. Bits [31:0] = TLP_DW3[7:0], TLP_DW3[15:8], TLP_DW3[23:16], TLP_DW3[31:24] The Packet Change Point registers (EINJ6_CHANGE_POINT*) specify which Tx TLP header bits to replace with the corresponding bits in the Packet Change Value registers (EINJ6_CHANGE_VALUE*). The type and number of errors are specified by the EINJ6_TLP_REG register. Only applies when EINJ6_INVERTED_CONTROL in EINJ6_TLP_REG =0.

PCIE_X4_EP_PF0_RAS_DES_CAP_EINJ6_CHANGE_VALUE_H3_REG_0

Offset: 0x244

PCIE_X8_EP_PF0_RAS_DES_CAP_EINJ6_CHANGE_VALUE_H3_REG_0

Offset: 0x258

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**EINJ6_CHANGE_VALUE_H3**: Packet Change Value: 4th DWORD. Specifies replacement values for the Tx TLP header DWORD#3 bits defined in the Packet Change Point registers (EINJ6_CHANGE_POINT*). Only applies when the EINJ6_INVERTED_CONTROL field in EINJ6_TLP_REG is '0'. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_EINJ6_TLP_REG_0

where <*j*> = 4, 8.

Description:The Packet Compare Point registers (EINJ6_COMPARE_POINT*) specify which Tx TLP header bits to compare with the corresponding bits in the Packet Compare Value registers (EINJ6_COMPARE_VALUE*). When all specified bits (in the Tx TLP header and EINJ6_COMPARE_VALUE*) match, the controller inserts errors into the TLP. The type and number of errors are specified by the this register. The Packet Change Point registers (EINJ6_CHANGE_POINT*) specify which Tx TLP header bits to replace with the corresponding bits in the Packet Change Value registers (EINJ6_CHANGE_VALUE*). The type and number of errors are specified by the this register. Only applies when EINJ6_INVERTED_CONTROL in this register =0. The TLP into that errors are injected will not arrive at the transaction layer of the remote device when all of the following conditions are true. - Using 128b/130b encoding - Injecting errors into TLP Length field / TLP digest bit.

PCIE_X4_EP_PF0_RAS_DES_CAP_EINJ6_TLP_REG_0

Offset: 0x248

PCIE_X8_EP_PF0_RAS_DES_CAP_EINJ6_TLP_REG_0

Offset: 0x25c

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:12	RO	0x0	**RSVDP_12**: Reserved for future use.
11:9	RW	0x0	**EINJ6_PACKET_TYPE**: Packet type. Selects the TLP packets to inject errors into. All encodings other than the specified encodings are reserved. Note: This register field is sticky.
8	RW	0x0	**EINJ6_INVERTED_CONTROL**: Inverted Error Injection Control. Encodded vlues given as above. Note: This register field is sticky.

Bit	R/W	Reset	Description
7:0	RW	0x0	**EINJ6_COUNT**: Error Injection Count. Indicates the number of errors to insert. This counter is decremented while errors are been inserted. - If the counter value is 0x01 and error is inserted, ERROR_INJECTION6_ENABLE in EINJ_ENABLE_REG returns '0'. - If the counter value is 0x00 and ERROR_INJECTION6_ENABLE=1, errors are inserted until ERROR_INJECTION6_ENABLE is set to '0'. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_SD_CONTROL1_REG_0

where <*j*> = 4, 8.

Description:For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X4_EP_PF0_RAS_DES_CAP_SD_CONTROL1_REG_0

Offset: 0x25c

PCIE_X8_EP_PF0_RAS_DES_CAP_SD_CONTROL1_REG_0

Offset: 0x270

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**RSVDP_24**: Reserved for future use.
23:22	RW	0x0	**LOW_POWER_INTERVAL**: Low Power Entry Interval Time. Interval Time that the controller starts monitoring RXELECIDLE signal after L0s/L1/L2 entry. You should set the value according to the latency from receiving EIOS to, RXELECIDLE assertion at the PHY. Note: This register field is sticky.
21:20	RW	0x0	**TX_EIOS_NUM**: Number of Tx EIOS. This register sets the number of transmit EIOS for L0s/L1 entry and Disable/Loopback/Hot-reset exit. The controller selects the greater value between this register and the value defined by the PCI-SIG specification. Note: This register field is sticky.
19:17	RO	0x0	**RSVDP_17**: Reserved for future use.

Bit	R/W	Reset	Description
16	RW	0x0	**FORCE_DETECT_LANE_EN**: Force Detect Lane Enable. Note: This register field is sticky.
15:0	RW	0x0	**FORCE_DETECT_LANE**: Force Detect Lane. When the FORCE_DETECT_LANE_EN field is set, the controller ignores receiver detection from PHY during LTSSM Detect state and uses this value instead. Value represents lane number. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_SD_CONTROL2_REG_0

where <*j*> = 4, 8.

Description:For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X4_EP_PF0_RAS_DES_CAP_SD_CONTROL2_REG_0

Offset: 0x260

PCIE_X8_EP_PF0_RAS_DES_CAP_SD_CONTROL2_REG_0

Offset: 0x274

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:17	RO	0x0	**RSVDP_17**: Reserved for future use.
16	RW	0x0	**FRAMING_ERR_RECOVERY_DISABLE**: Framing Error Recovery Disable. This bit disables a transition to Recovery state when a Framing Error is occurred. Note: This register field is sticky.
15:11	RO	0x0	**RSVDP_11**: Reserved for future use.
10	RW	0x0	**DIRECT_LPBKSLV_TO_EXIT**: Direct Loopback Slave To Exit. Note: This register field is sticky.
9	RW	0x0	**DIRECT_POLCOMP_TO_DETECT**: Direct Polling.Compliance to Detect. Note: This register field is sticky.

Bit	R/W	Reset	Description
8	RW	0x0	**DIRECT_RECIDLE_TO_CONFIG**: Direct Recovery.Idle to Configuration. Note: This register field is sticky.
7:3	RO	0x0	**RSVDP_3**: Reserved for future use.
2	RW	0x0	**NOACK_FORCE_LINKDOWN**: Force LinkDown. Note: This register field is sticky.
1	WO	0x0	**RECOVERY_REQUEST**: Recovery Request.
0	RW	0x0	**HOLD_LTSSM**: Hold and Release LTSSM. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_SD_STATUS_L1LANE_REG_0

where <*j*> = 4, 8.

Description:This viewport register returns the data selected by the following field: - LANE_SELECT in SD_STATUS_L1LANE_REG For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X4_EP_PF0_RAS_DES_CAP_SD_STATUS_L1LANE_REG_0

Offset: 0x26c

PCIE_X8_EP_PF0_RAS_DES_CAP_SD_STATUS_L1LANE_REG_0

Offset: 0x280

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00180000 (0b0000,0000,0001,1000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**DESKEW_POINTER**: Deskew Pointer. Indicates Deskew pointer of internal Deskew buffer of selected lane number(LANE_SELECT). Note: This register field is sticky.
23:21	RO	0x0	**RSVDP_21**: Reserved for future use.

Bit	R/W	Reset	Description
20	RO	0x1	**PIPE_TXELECIDLE**: PIPE:TxElecIdle. Indicates PIPE TXELECIDLE signal of selected lane number(LANE_SELECT). Note: This register field is sticky.
19	RO	0x1	**PIPE_RXELECIDLE**: PIPE:RxElecIdle. Indicates PIPE RXELECIDLE signal of selected lane number(LANE_SELECT). Note: This register field is sticky.
18	RO	0x0	**PIPE_RXVALID**: PIPE:RxValid. Indicates PIPE RXVALID signal of selected lane number(LANE_SELECT). Note: This register field is sticky.
17	RO	0x0	**PIPE_DETECT_LANE**: PIPE:Detect Lane. Indicates whether PHY indicates receiver detection or not on selected lane number(LANE_SELECT). Note: This register field is sticky.
16	RO	0x0	**PIPE_RXPOLARITY**: PIPE:RxPolarity. Indicates PIPE RXPOLARITY signal of selected lane number(LANE_SELECT). Note: This register field is sticky.
15:4	RO	0x0	**RSVDP_4**: Reserved for future use.
3:0	RW	0x0	**LANE_SELECT**: Lane Select. Lane Select register for Silicon Debug Status Register of Layer1-PerLane. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_SD_STATUS_L1LTSSM_REG_0

where <*j*> = 4, 8.

Description:For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X4_EP_PF0_RAS_DES_CAP_SD_STATUS_L1LTSSM_REG_0

Offset: 0x270

PCIE_X8_EP_PF0_RAS_DES_CAP_SD_STATUS_L1LTSSM_REG_0

Offset: 0x284

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000200 (0b0000,0000,0000,0000,0000,0010,0000,0000)

Bit	R/W	Reset	Description
31:16	RO	0x0	**LTSSM_VARIABLE**: LTSSM Variable. Indicates internal LTSSM variables defined in the PCI Express Base Specification. For other value idle_to_rlock_transitioned. Note: This register field is sticky.
15	RO	0x0	**LANE_REVERSAL**: Lane Reversal Operation. Receiver detected lane reversal. This field is only valid in the L0 LTSSM state. Note: This register field is sticky.
14:11	RO	0x0	**RSVDP_11**: Reserved for future use.
10:8	RO	0x2	**PIPE_POWER_DOWN**: PIPE:PowerDown. Indicates PIPE PowerDown signal. Note: This register field is sticky.
7	RW	0x0	**FRAMING_ERR**: Framing Error. Indicates Framing Error detection status.
6:0	RO	0x0	**FRAMING_ERR_PTR**: First Framing Error Pointer. Identifies the first Framing Error using the following encoding. The field contents are only valid value when FRAMING_ERR =1. - Received Unexpected Framing Token (Values 01h to 06h) - Received Unexpected STP Token (Values 11h to 13h) - Received Unexpected Block (Values 21h to 2Eh) All encodings other than the defined encodings are reserved. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_SD_STATUS_PM_REG_0

where <*j*> = 4, 8.

Description:For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X4_EP_PF0_RAS_DES_CAP_SD_STATUS_PM_REG_0

Offset: 0x274

PCIE_X8_EP_PF0_RAS_DES_CAP_SD_STATUS_PM_REG_0

Offset: 0x288

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**RSVDP_24**: Reserved for future use.
23:16	RO	0x0	**LATCHED_NFTS**: Latched N_FTS. Indicates the value of N_FTS in the received TS Ordered Sets from the Link Partner. Note: This register field is sticky.
15:13	RO	0x0	**L1SUB_STATE**: L1Sub State. Indicates internal state machine of L1Sub state. Note: This register field is sticky.
12	RW	0x0	**PME_RESEND_FLAG**
11:8	RO	0x0	**INTERNAL_PM_SSTATE**: Internal PM State(Slave). Indicates internal state machine of Power Management Slave controller. Note: This register field is sticky.
7:5	RO	0x0	**RSVDP_5**: Reserved for future use.
4:0	RO	0x0	**INTERNAL_PM_MSTATE**: Internal PM State(Master). Indicates internal state machine of Power Management Master controller. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_SD_STATUS_L2_REG_0

where <*j*> = 4, 8.

Description:For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X4_EP_PF0_RAS_DES_CAP_SD_STATUS_L2_REG_0

Offset: 0x278

PCIE_X8_EP_PF0_RAS_DES_CAP_SD_STATUS_L2_REG_0

Offset: 0x28c

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00fff000 (0b0000,0000,1111,1111,1111,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**RSVDP_28**: Reserved for future use.
27	0x0	**FC_INIT2**: FC_INIT2. Indicates the controller is in FC_INIT2(VC0) state. Note: This register field is sticky.
26	0x0	**FC_INIT1**: FC_INIT1. Indicates the controller is in FC_INIT1(VC0) state. Note: This register field is sticky.
25:24	0x0	**DLCMSM**: DLCMSM. Indicates the current DLCMSM. Note: This register field is sticky.
23:12	0xfff	**RX_ACK_SEQ_NO**: Tx Ack Sequence Number. Indicates ACKD_SEQ which is updated by receiving ACK/NAK DLLP. Note: This register field is sticky.
11:0	0x0	**TX_TLP_SEQ_NO**: Tx Tlp Sequence Number. Indicates next transmit sequence number for transmit TLP. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_SD_STATUS_L3FC_REG_0

where <*j*> = 4, 8.

Description:The CREDIT_DATA[0/1] fields in this viewport register return the data for the VC and TLP Type selected by the following fields: - CREDIT_SEL_VC - CREDIT_SEL_CREDIT_TYPE - CREDIT_SEL_TLP_TYPE - CREDIT_SEL_HD For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X4_EP_PF0_RAS_DES_CAP_SD_STATUS_L3FC_REG_0

Offset: 0x27c

PCIE_X8_EP_PF0_RAS_DES_CAP_SD_STATUS_L3FC_REG_0

Offset: 0x290

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:20	RO	0x0	**CREDIT_DATA1**: Credit Data1. Current FC credit data selected by the CREDIT_SEL_VC, CREDIT_SEL_CREDIT_TYPE, CREDIT_SEL_TLP_TYPE, and CREDIT_SEL_HD viewport-select fields. - Rx: Credit Allocated Value - Tx: Credit Limit Value. This value is valid when DLCMSM=0x3(DL_ACTIVE). Note: This register field is sticky.
19:8	RO	0x0	**CREDIT_DATA0**: Credit Data0. Current FC credit data selected by the CREDIT_SEL_VC, CREDIT_SEL_CREDIT_TYPE, CREDIT_SEL_TLP_TYPE, and CREDIT_SEL_HD viewport-select fields. - Rx: Credit Received Value - Tx: Credit Consumed Value Note: This register field is sticky.
7	RO	0x0	**RSVDP_7**: Reserved for future use.
6	RW	0x0	**CREDIT_SEL_HD**: Credit Select(HeaderData). This field in conjunction with the CREDIT_SEL_VC, CREDIT_SEL_CREDIT_TYPE, and CREDIT_SEL_TLP_TYPE viewport-select fields determines that data that is returned by the CREDIT_DATA0 and CREDIT_DATA1 data fields. Note: This register field is sticky.
5:4	RW	0x0	**CREDIT_SEL_TLP_TYPE**: Credit Select(TLP Type). This field in conjunction with the CREDIT_SEL_VC, CREDIT_SEL_CREDIT_TYPE, and CREDIT_SEL_HD viewport-select fields determines that data that is returned by the CREDIT_DATA0 and CREDIT_DATA1 data fields. Note: This register field is sticky.
3	RW	0x0	**CREDIT_SEL_CREDIT_TYPE**: Credit Select(Credit Type). This field in conjunction with the CREDIT_SEL_VC, CREDIT_SEL_TLP_TYPE, and CREDIT_SEL_HD viewport-select fields determines that data that is returned by the CREDIT_DATA0 and CREDIT_DATA1 data fields. Note: This register field is sticky.
2:0	RW	0x0	**CREDIT_SEL_VC**: Credit Select(VC). This field in conjunction with the CREDIT_SEL_CREDIT_TYPE, CREDIT_SEL_TLP_TYPE, and CREDIT_SEL_HD viewport-select fields determines that data that is returned by the CREDIT_DATA0 and CREDIT_DATA1 data fields. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_SD_STATUS_L3_REG_0

where <*j*> = 4, 8.

Description:Silicon Debug Status (Layer3). For more details, see the RAS DES section in the Core Operations chapter of the Databook.)

PCIE_X4_EP_PF0_RAS_DES_CAP_SD_STATUS_L3_REG_0

Offset: 0x280

PCIE_X8_EP_PF0_RAS_DES_CAP_SD_STATUS_L3_REG_0

Offset: 0x294

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:8	RO	0x0	**RSVDP_8**: Reserved for future use.
7	RW	0x0	**MFTLP_STATUS**: Malformed TLP Status. Indicates malformed TLP has occurred.
6:0	RO	0x0	**MFTLP_POINTER**: First Malformed TLP Error Pointer. Indicates the element of the received first malformed TLP. This pointer is validated by MFTLP_STATUS. All encodings other than the defined encodings are reserved. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_SD_EQ_CONTROL1_REG_0

where <*j*> = 4, 8.

Description:This is a viewport control register. Setting the EQ_RATE_SEL and EQ_LANE_SEL fields in this register determine the per-lane Silicon Debug EQ Status data returned by the SD_EQ_STATUS[1/2/3] viewport registers. For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X4_EP_PF0_RAS_DES_CAP_SD_EQ_CONTROL1_REG_0

Offset: 0x28c

PCIE_X8_EP_PF0_RAS_DES_CAP_SD_EQ_CONTROL1_REG_0

Offset: 0x2a0

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:24	RW	0x0	**FOM_TARGET**: FOM Target. Indicates figure of merit target criteria value of EQ Master(DSP in EQ Phase3/USP in EQ Phase2). This field is only valid when GEN3_EQ_FB_MODE is 0001b(Figure Of Merit). Note: This register field is sticky.
23	RW	0x0	**FOM_TARGET_ENABLE**: FOM Target Enable. Enables the FOM_TARGET fields. Note: This register field is sticky.
22:18	RO	0x0	**RSVDP_18**: Reserved for future use.
17:16	RW	0x0	**EVAL_INTERVAL_TIME**: Eval Interval Time. Indicates interval time of RxEqEval assertion. This field is used for EQ Master(DSP in EQ Phase3/USP in EQ Phase2). Note: This register field is sticky.
15:10	RO	0x0	**RSVDP_10**: Reserved for future use.
9:8	RW	0x0	**EXT_EQ_TIMEOUT**: Extends EQ Phase2/3 Timeout. This field is used when the LTSSM is in Recovery.EQ2/3. When this field is set, the value of EQ2/3 timeout is extended. Note: This register field is sticky.
7:6	RO	0x0	**RSVDP_6**: Reserved for future use.
5:4	RW	0x0	**EQ_RATE_SEL**: EQ Status Rate Select. Setting this field in conjunction with the EQ_LANE_SEL field determines the per-lane Silicon Debug EQ Status data returned by the SD_EQ_CONTROL[2/3] and SD_EQ_STATUS[1/2/3] viewport registers. Note: This register field is sticky.
3:0	RW	0x0	**EQ_LANE_SEL**: EQ Status Lane Select. Setting this field in conjunction with the EQ_RATE_SEL field determines the per-lane Silicon Debug EQ Status data returned by the SD_EQ_CONTROL[2/3] and SD_EQ_STATUS[1/2/3] viewport registers. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_SD_EQ_CONTROL2_REG_0

where <*j*> = 4, 8.

Description:This viewport register returns the value for the rate and lane selected by the EQ_RATE_SEL and EQ_LANE_SEL fields in the SD_EQ_CONTROL1_REG register. For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X4_EP_PF0_RAS_DES_CAP_SD_EQ_CONTROL2_REG_0

Offset: 0x290

PCIE_X8_EP_PF0_RAS_DES_CAP_SD_EQ_CONTROL2_REG_0

Offset: 0x2a4

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RO	0x0	**RSVDP_31**: Reserved for future use.
30	RW	0x0	**FORCE_LOCAL_TX_PRESET_ENABLE**: Force Local Transmitter Preset Enable. Enables the FORCE_LOCAL_TX_PRESET field. If select rate in the EQ_RATE_SEL field is 32.0GT/s Speed, this feature is not available. Note: This register field is sticky.
29	RW	0x0	**FORCE_LOCAL_RX_HINT_ENABLE**: Force Local Receiver Preset Hint Enable. Enables the FORCE_LOCAL_RX_HINT field. If select rate in the EQ_RATE_SEL field is other than 8.0GT/s Speed, this feature is not available. Note: This register field is sticky.
28	RW	0x0	**FORCE_LOCAL_TX_COEF_ENABLE**: Force Local Transmitter Coefficient Enable. Enables the following fields: - FORCE_LOCAL_TX_PRE_CURSOR - FORCE_LOCAL_TX_CURSOR - FORCE_LOCAL_TX_POST_CURSOR Note: This register field is sticky.
27:24	RW	0x0	**FORCE_LOCAL_TX_PRESET**: Force Local Transmitter Preset. Indicates initial preset value of USP in EQ Slave(EQ Phase2) instead of receiving EQ TS2. If select rate in the EQ_RATE_SEL field is 32.0GT/s Speed, this feature is not available. Note: This register field is sticky.
23:21	RO	0x0	**RSVDP_21**: Reserved for future use.
20:18	RW	0x0	**FORCE_LOCAL_RX_HINT**: Force Local Receiver Preset Hint. Indicates the RxPresetHint value of EQ Slave(DSP in EQ Phase2/USP in EQ Phase3), instead of received or set value. If select rate in the EQ_RATE_SEL field is other than 8.0GT/s Speed, this feature is not available. Note: This register field is sticky.
17:12	RW	0x0	**FORCE_LOCAL_TX_POST_CURSOR**: Force Local Transmitter Post-Cursor. Indicates the coefficient value of EQ Slave(DSP in EQ Phase2/USP in EQ Phase3), instead of the value instructed from link partner. Note: This register field is sticky.
11:6	RW	0x0	**FORCE_LOCAL_TX_CURSOR**: Force Local Transmitter Cursor. Indicates the coefficient value of EQ Slave(DSP in EQ Phase2/USP in EQ Phase3), instead of the value instructed from link partner. Note: This register field is sticky.

Bit	R/W	Reset	Description
5:0	RW	0x0	**FORCE_LOCAL_TX_PRE_CURSOR**: Force Local Transmitter Pre-cursor. Indicates the coefficient value of EQ Slave(DSP in EQ Phase2/USP in EQ Phase3), instead of the value instructed from link partner. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_SD_EQ_CONTROL3_REG_0

where <*j*> = 4, 8.

Description:This viewport register returns the value for the rate and lane selected by the EQ_RATE_SEL and EQ_LANE_SEL fields in the SD_EQ_CONTROL1_REG register. For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X4_EP_PF0_RAS_DES_CAP_SD_EQ_CONTROL3_REG_0

Offset: 0x294

PCIE_X8_EP_PF0_RAS_DES_CAP_SD_EQ_CONTROL3_REG_0

Offset: 0x2a8

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:29	RO	0x0	**RSVDP_29**: Reserved for future use.
28	RW	0x0	**FORCE_REMOTE_TX_COEF_ENABLE**: Force Remote Transmitter Coefficient Enable. Enables the following fields: - FORCE_REMOTE_TX_PRE_CURSOR - FORCE_REMOTE_TX_CURSOR - FORCE_REMOTE_TX_POST_CURSOR This function can only be used when GEN3_EQ_FB_MODE = 0000b(Direction Change) Note: This register field is sticky.
27:18	RO	0x0	**RSVDP_18**: Reserved for future use.
17:12	RW	0x0	**FORCE_REMOTE_TX_POST_CURSOR**: Force Remote Transmitter Post-Cursor. Indicates the coefficient value of EQ Master(DSP in EQ Phase3/USP in EQ Phase2), instead of the value instructed from local phy in dirchange mode. Note: This register field is sticky.

Bit	R/W	Reset	Description
11:6	RW	0x0	**FORCE_REMOTE_TX_CURSOR**: Force Remote Transmitter Cursor. Indicates the coefficient value of EQ Master(DSP in EQ Phase3/USP in EQ Phase2), instead of the value instructed from local phy in dirchange mode. Note: This register field is sticky.
5:0	RW	0x0	**FORCE_REMOTE_TX_PRE_CURSOR**: Force Remote Transmitter Pre-Cursor. Indicates the coefficient value of EQ Master(DSP in EQ Phase3/USP in EQ Phase2), instead of the value instructed from local phy in dirchange mode. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_SD_EQ_STATUS1_REG_0

where <*j*> = 4, 8.

Description:This viewport register returns the first of three words of Silicon Debug EQ Status data for the rate and lane selected by the EQ_RATE_SEL and EQ_LANE_SEL fields in the SD_EQ_CONTROL1_REG register. The following fields are available when Equalization finished unsuccessfully(EQ_CONVERGENCE_INFO=2). - EQ_RULEA_VIOLATION - EQ_RULEB_VIOLATION - EQ_RULEC_VIOLATION - EQ_REJECT_EVENT For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X4_EP_PF0_RAS_DES_CAP_SD_EQ_STATUS1_REG_0

Offset: 0x29c

PCIE_X8_EP_PF0_RAS_DES_CAP_SD_EQ_STATUS1_REG_0

Offset: 0x2b0

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:8	0x0	**RSVDP_8**: Reserved for future use.
7	0x0	**EQ_REJECT_EVENT**: EQ Reject Event. Indicates that the controller receives two consecutive TS1 OS w/Reject=1b during EQ Master phase(DSP in EQ Phase3/USP in EQ Phase2). This bit is automatically cleared when the controller starts EQ Master phase again. Note: This register field is sticky.

Bit	Reset	Description
6	0x0	**EQ_RULEC_VIOLATION**: EQ Rule C Violation. Indicates that coefficients rule C violation is detected in the values provided by PHY using direction change method during EQ Master phase(DSP in EQ Phase3/USP in EQ Phase2). The coefficients rule C correspond to the rules c) from section "Rules for Transmitter Coefficents" in the PCI Express Base Specification. This bit is automatically cleared when the controller starts EQ Master phase again. Note: This register field is sticky.
5	0x0	**EQ_RULEB_VIOLATION**: EQ Rule B Violation. Indicates that coefficients rule B violation is detected in the values provided by PHY using direction change method during EQ Master phase(DSP in EQ Phase3/USP in EQ Phase2). The coefficients rules B correspond to the rules b) from section "Rules for Transmitter Coefficents" in the PCI Express Base Specification. This bit is automatically cleared when the controller starts EQ Master phase again. Note: This register field is sticky.
4	0x0	**EQ_RULEA_VIOLATION**: EQ Rule A Violation. Indicates that coefficients rule A violation is detected in the values provided by PHY using direction change method during EQ Master phase(DSP in EQ Phase3/USP in EQ Phase2). The coefficients rules A correspond to the rules a) from section "Rules for Transmitter Coefficents" in the PCI Express Base Specification. This bit is automatically cleared when the controller starts EQ Master phase again. Note: This register field is sticky.
3	0x0	**RSVDP_3**: Reserved for future use.
2:1	0x0	**EQ_CONVERGENCE_INFO**: EQ Convergence Info. Indicates equalization convergence information. This bit is automatically cleared when the controller starts EQ Master phase again. Note: This register field is sticky.
0	0x0	**EQ_SEQUENCE**: EQ Sequence. Indicates that the controller is starting the equalization sequence. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_SD_EQ_STATUS2_REG_0

where <*j*> = 4, 8.

Description:This viewport register returns the second of three words of Silicon Debug EQ Status data for the rate and lane selected by the EQ_RATE_SEL and EQ_LANE_SEL fields in the SD_EQ_CONTROL1_REG register. Each field is available when Equalization finished successfully(EQ_CONVERGENCE_INFO=1). For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X4_EP_PF0_RAS_DES_CAP_SD_EQ_STATUS2_REG_0

Offset: 0x2a0

PCIE_X8_EP_PF0_RAS_DES_CAP_SD_EQ_STATUS2_REG_0

Offset: 0x2b4

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**EQ_LOCAL_FOM_VALUE**: EQ Local Figure of Merit. Indicates Local maximum Figure of Merit value. Note: This register field is sticky.
23:21	0x0	**RSVDP_21**: Reserved for future use.
20:18	0x0	**EQ_LOCAL_RX_HINT**: EQ Local Receiver Preset Hint. Indicates Local Receiver Preset Hint value. If select rate in the EQ_RATE_SEL field is other than 8.0GT/s Speed, this feature is not available. Note: This register field is sticky.
17:12	0x0	**EQ_LOCAL_POST_CURSOR**: EQ Local Post-Cursor. Indicates Local post cursor coefficient value. Note: This register field is sticky.
11:6	0x0	**EQ_LOCAL_CURSOR**: EQ Local Cursor. Indicates Local cursor coefficient value. Note: This register field is sticky.
5:0	0x0	**EQ_LOCAL_PRE_CURSOR**: EQ Local Pre-Cursor. Indicates Local pre cursor coefficient value. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_RAS_DES_CAP_SD_EQ_STATUS3_REG_0

where <*j*> = 4, 8.

Description:This viewport register returns the third of three words of Silicon Debug EQ Status data for the rate and lane selected by the EQ_RATE_SEL and EQ_LANE_SEL fields in the SD_EQ_CONTROL1_REG register. Each field is available when Equalization finished successfully(EQ_CONVERGENCE_INFO=1). For more details, see the RAS DES section in the Core Operations chapter of the Databook.

PCIE_X4_EP_PF0_RAS_DES_CAP_SD_EQ_STATUS3_REG_0

Offset: 0x2a4

PCIE_X8_EP_PF0_RAS_DES_CAP_SD_EQ_STATUS3_REG_0

Offset: 0x2b8

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:30	0x0	**RSVDP_30**: Reserved for future use.
29:24	0x0	**EQ_REMOTE_FS**: EQ Remote FS. Indicates Remote FS value. Note: This register field is sticky.
23:18	0x0	**EQ_REMOTE_LF**: EQ Remote LF. Indicates Remote LF value. Note: This register field is sticky.
17:12	0x0	**EQ_REMOTE_POST_CURSOR**: EQ Remote Post-Cursor. Indicates Remote post cursor coefficient value. Note: This register field is sticky.
11:6	0x0	**EQ_REMOTE_CURSOR**: EQ Remote Cursor. Indicates Remote cursor coefficient value. Note: This register field is sticky.
5:0	0x0	**EQ_REMOTE_PRE_CURSOR**: EQ Remote Pre-Cursor. Indicates Remote pre cursor coefficient value. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_VSECRAS_CAP_RASDP_EXT_CAP_HDR_OFF_0

where <*j*> = 4, 8.

Description:This Register provides capability ID, Capability version, and Next capability offset for PCIe Extended capability stucture.

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X4_EP_PF0_VSECRAS_CAP_RASDP_EXT_CAP_HDR_OFF_0

Offset: 0x2bc
Reset: 0x2f41000b (0b0010,1111,0100,0001,0000,0000,0000,1011)

Bit	Reset	Description
31:20	0x2f4	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. The bottom 2 bits of this offset are Reserved and must be implemented as 00b although software must mask them to allow for future uses of these bits. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**CAP**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. A version of the specification that changes the Extended Capability in a way that is not otherwise identifiable (e.g., through a new Capability field) is permitted to increment this field. All such changes to the Capability structure must be software-compatible. Software must check for Capability Version numbers that are greater than or equal to the highest number defined when the software is written, as Functions reporting any such Capability Version numbers will contain a Capability structure that is compatible with that piece of software. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0xb	**ID**: PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X8_EP_PF0_VSECRAS_CAP_RASDP_EXT_CAP_HDR_OFF_0

Offset: 0x2d0
Reset: 0x3081000b (0b0011,0000,1000,0001,0000,0000,0000,1011)

Bit	Reset	Description
31:20	0x308	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. The bottom 2 bits of this offset are Reserved and must be implemented as 00b although software must mask them to allow for future uses of these bits. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

Bit	Reset	Description
19:16	0x1	**CAP**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. A version of the specification that changes the Extended Capability in a way that is not otherwise identifiable (e.g., through a new Capability field) is permitted to increment this field. All such changes to the Capability structure must be software-compatible. Software must check for Capability Version numbers that are greater than or equal to the highest number defined when the software is written, as Functions reporting any such Capability Version numbers will contain a Capability structure that is compatible with that piece of software. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0xb	**ID**: PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_VSECRAS_CAP_RASDP_VENDOR_SPECIFIC_HDR_OFF_0

where <*j*> = 4, 8.

Description:This Register provides VSEC Length, VSEC ID and VSEC Rev (Version Number). Vendor-specific software must qualify the associated Vendor ID of the PCI Express Function or RCRB before attempting to interpret the values in the VSEC ID or VSEC Rev fields.

PCIE_X4_EP_PF0_VSECRAS_CAP_RASDP_VENDOR_SPECIFIC_HDR_OFF_0

Offset: 0x2c0

PCIE_X8_EP_PF0_VSECRAS_CAP_RASDP_VENDOR_SPECIFIC_HDR_OFF_0

Offset: 0x2d4

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x03810001 (0b0000,0011,1000,0001,0000,0000,0000,0001)

Bit	Reset	Description
31:20	0x38	**VSEC_LENGTH**: VSEC Length. This field indicates the number of bytes in the entire VSEC structure, including the Vendor-Specific Extended Capability Header, the Vendor-Specific Header, and the vendor-specific registers. Note: This register field is sticky.

Bit	Reset	Description
19:16	0x1	**VSEC_REV**: VSEC Rev. This field is a vendor-defined version number that indicates the version of the VSEC structure. Software must qualify the Vendor ID and VSEC ID before interpreting this field. Note: This register field is sticky.
15:0	0x1	**VSEC_ID**: VSEC ID. This field is a vendor-defined ID number that indicates the nature and format of the VSEC structure. Software must qualify the Vendor ID before interpreting this field. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_VSECRAS_CAP_RASDP_ERROR_PROT_CTRL_OFF_0

where <*j*> = 4, 8.

Description:Allows you to disable ECC error correction for RAMs and datapath. When the AXI Bridge Module is implemented and the master / slave clocks are asynchronous to the PCIe native controller clock (core_clk), you must not write this register while operations are in progress in the AXI master / slave interface.

PCIE_X4_EP_PF0_VSECRAS_CAP_RASDP_ERROR_PROT_CTRL_OFF_0

Offset: 0x2c4

PCIE_X8_EP_PF0_VSECRAS_CAP_RASDP_ERROR_PROT_CTRL_OFF_0

Offset: 0x2d8

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**RSVDP_24**: Reserved for future use.
23	RW	0x0	**ERROR_PROT_DISABLE_CXS_RX**: Error correction disable for CXS Tx path (PCIe Rx path). Note: This register field is sticky.
22	RW	0x0	**ERROR_PROT_DISABLE_ADM_RX**: Error correction disable for ADM Rx path. Note: This register field is sticky.
21	RW	0x0	**ERROR_PROT_DISABLE_LAYER3_RX**: Error correction disable for layer 3 Rx path. Note: This register field is sticky.

Bit	R/W	Reset	Description
20	RW	0x0	**ERROR_PROT_DISABLE_LAYER2_RX**: Error correction disable for layer 2 Rx path. Note: This register field is sticky.
19	RW	0x0	**ERROR_PROT_DISABLE_DMA_READ**: Error correction disable for DMA read engine. Note: This register field is sticky.
18	RW	0x0	**ERROR_PROT_DISABLE_AXI_BRIDGE_INBOUND_REQUEST**: Error correction disable for AXI bridge inbound request path. Note: This register field is sticky.
17	RW	0x0	**ERROR_PROT_DISABLE_AXI_BRIDGE_INBOUND_COMPLETION**: Error correction disable for AXI bridge inbound completion composer. Does not disable the error detection reporting for 1-bit and 2-bit ECC errors. Note: This register field is sticky.
16	RW	0x0	**ERROR_PROT_DISABLE_RX**: Global error correction disable for all Rx layers. Note: This register field is sticky.
15:9	RO	0x0	**RSVDP_9**: Reserved for future use.
8	RW	0x0	**ERROR_PROT_DISABLE_DTIM_TX**: Error correction disable for DTIM Tx path. Note: This register field is sticky.
7	RW	0x0	**ERROR_PROT_DISABLE_CXS_TX**: Error correction disable for CXS Rx path (PCIe Tx path). Note: This register field is sticky.
6	RW	0x0	**ERROR_PROT_DISABLE_ADM_TX**: Error correction disable for Adm Tx path. Note: This register field is sticky.
5	RW	0x0	**ERROR_PROT_DISABLE_LAYER3_TX**: Error correction disable for layer 3 Tx path. Note: This register field is sticky.
4	RW	0x0	**ERROR_PROT_DISABLE_LAYER2_TX**: Error correction disable for layer 2 Tx path. Note: This register field is sticky.
3	RW	0x0	**ERROR_PROT_DISABLE_DMA_WRITE**: Error correction disable for DMA write engine. Note: This register field is sticky.
2	RW	0x0	**ERROR_PROT_DISABLE_AXI_BRIDGE_OUTBOUND**: Error correction disable for AXI bridge outbound request path. Note: This register field is sticky.
1	RW	0x0	**ERROR_PROT_DISABLE_AXI_BRIDGE_MASTER**: Error correction disable for AXI bridge master completion buffer. Note: This register field is sticky.
0	RW	0x0	**ERROR_PROT_DISABLE_TX**: Global error correction disable for all Tx layers. Does not disable the error detection reporting for 1-bit and 2-bit ECC errors. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_VSECRAS_CAP_RASDP_CORR_COUNTER_CTRL_OFF_0

where <*j*> = 4, 8.

Description:This is a viewport control register. Setting the CORR_COUNTER_SELECTION_REGION and CORR_COUNTER_SELECTION fields in this register determine the counter data returned by the RASDP_CORR_COUNT_REPORT_OFF viewport data register.

PCIE_X4_EP_PF0_VSECRAS_CAP_RASDP_CORR_COUNTER_CTRL_OFF_0

Offset: 0x2c8

PCIE_X8_EP_PF0_VSECRAS_CAP_RASDP_CORR_COUNTER_CTRL_OFF_0

Offset: 0x2dc

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000010 (0b0000,0000,0000,0000,0000,0000,0001,0000)

Bit	R/W	Reset	Description
31:24	RW	0x0	**CORR_COUNTER_SELECTION**: Counter selection. This field selects the counter ID (within the region defined by CORR_COUNTER_SELECTION_REGION) whose contents can be read from the RASDP_CORR_COUNT_REPORT_OFF register. You can cycle this field value from 0 to 255 to access all counters.
23:20	RW	0x0	**CORR_COUNTER_SELECTION_REGION**: Select correctable counter region. All encodings other than the defined encodings are reserved.
19:5	RO	0x0	**RSVDP_5**: Reserved for future use.
4	RW	0x1	**CORR_EN_COUNTERS**: Enable correctable errors counters. The counters are enabled by default.
3:1	RO	0x0	**RSVDP_1**: Reserved for future use.
0	WO	0x0	**CORR_CLEAR_COUNTERS**: Clear all correctable error counters.

PCIE_X<*j*>_EP_PF0_VSECRAS_CAP_RASDP_CORR_COUNT_REPORT_OFF_0

where <*j*> = 4, 8.

Description:This viewport register returns the counter data selected by the CORR_COUNTER_SELECTION_REGION and CORR_COUNTER_SELECTION fields in the RASDP_CORR_COUNTER_CTRL_OFF register.

PCIE_X4_EP_PF0_VSECRAS_CAP_RASDP_CORR_COUNT_REPORT_OFF_0

Offset: 0x2cc

PCIE_X8_EP_PF0_VSECRAS_CAP_RASDP_CORR_COUNT_REPORT_OFF_0

Offset: 0x2e0

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**CORR_COUNTER_SELECTED**: Counter selection. Returns the value set in the CORR_COUNTER_SELECTION field of the RASDP_CORR_COUNTER_CTRL_OFF register.
23:20	0x0	**CORR_COUNTER_SELECTED_REGION**: Selected correctable counter region. All encodings other than the defined encodings are reserved.
19:8	0x0	**RSVDP_8**: Reserved for future use.
7:0	0x0	**CORR_COUNTER**: Current corrected error count for the selected counter.

PCIE_X<*j*>_EP_PF0_VSECRAS_CAP_RASDP_UNCORR_COUNTER_CTRL_OFF_0

where <*j*> = 4, 8.

Description:This is a viewport control register. Setting the UNCORR_COUNTER_SELECTION_REGION and UNCORR_COUNTER_SELECTION fields in this register determine the counter data returned by the RASDP_UNCORR_COUNT_REPORT_OFF viewport data register.

PCIE_X4_EP_PF0_VSECRAS_CAP_RASDP_UNCORR_COUNTER_CTRL_OFF_0

Offset: 0x2d0

PCIE_X8_EP_PF0_VSECRAS_CAP_RASDP_UNCORR_COUNTER_CTRL_OFF_0

Offset: 0x2e4

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000010 (0b0000,0000,0000,0000,0000,0000,0001,0000)

Bit	R/W	Reset	Description
31:24	RW	0x0	**UNCORR_COUNTER_SELECTION**: Counter selection. This field selects the counter ID (within the region defined by UNCORR_COUNTER_SELECTION_REGION) whose contents can be read from the RASDP_UNCORR_COUNT_REPORT_OFF register. You can cycle this field value from 0 to 255 to access all counters.
23:20	RW	0x0	**UNCORR_COUNTER_SELECTION_REGION**: Select uncorrectable counter region. All encodings other than the defined encodings are reserved.
19:5	RO	0x0	**RSVDP_5**: Reserved for future use.
4	RW	0x1	**UNCORR_EN_COUNTERS**: Enable uncorrectable errors counters. The counters are enabled by default.
3:1	RO	0x0	**RSVDP_1**: Reserved for future use.
0	WO	0x0	**UNCORR_CLEAR_COUNTERS**: Clear uncorrectable errors counters. When asserted causes all counters tracking the uncorrectable errors to be cleared.

PCIE_X<*j*>_EP_PF0_VSECRAS_CAP_RASDP_UNCORR_COUNT_REPORT_OFF_0

where <*j*> = 4, 8.

Description:This viewport register returns the counter data selected by the UNCORR_COUNTER_SELECTION_REGION and UNCORR_COUNTER_SELECTION fields in the RASDP_UNCORR_COUNTER_CTRL_OFF register.

PCIE_X4_EP_PF0_VSECRAS_CAP_RASDP_UNCORR_COUNT_REPORT_OFF_0

Offset: 0x2d4

PCIE_X8_EP_PF0_VSECRAS_CAP_RASDP_UNCORR_COUNT_REPORT_OFF_0

Offset: 0x2e8

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**UNCORR_COUNTER_SELECTED**: Counter selection. Returns the value set in the UNCORR_COUNTER_SELECTION field of the RASDP_UNCORR_COUNTER_CTRL_OFF register.
23:20	0x0	**UNCORR_COUNTER_SELECTED_REGION**: Selected uncorrectable counter region. All encodings other than the defined encodings are reserved.
19:8	0x0	**RSVDP_8**: Reserved for future use.
7:0	0x0	**UNCORR_COUNTER**: Current uncorrected error count for the selected counter

PCIE_X<*j*>_EP_PF0_VSECRAS_CAP_RASDP_ERROR_INJ_CTRL_OFF_0

where <*j*> = 4, 8.

Description:Error injection control for the following features: - 1-bit or 2-bit injection - Continuous or fixed-number (n) injection modes - Global enable/disable - Selectable location where injection occurs.

PCIE_X4_EP_PF0_VSECRAS_CAP_RASDP_ERROR_INJ_CTRL_OFF_0

Offset: 0x2d8

PCIE_X8_EP_PF0_VSECRAS_CAP_RASDP_ERROR_INJ_CTRL_OFF_0

Offset: 0x2ec

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**RSVDP_24**: Reserved for future use.

Bit	R/W	Reset	Description
23:16	RW	0x0	**ERROR_INJ_LOC**: Error injection location. Selects where error injection takes place. You can cycle this field value from 0 to 255 to access all locations.
15:8	RW	0x0	**ERROR_INJ_COUNT**: Error injection count. If value is n, n amount of errors injected.
7:6	RO	0x0	**RSVDP_6**: Reserved for future use.
5:4	RW	0x0	**ERROR_INJ_TYPE**: Error injection type.
3:1	RO	0x0	**RSVDP_1**: Reserved for future use.
0	RW	0x0	**ERROR_INJ_EN**: Error injection global enable. When set, enables the error insertion logic.

PCIE_X<*j*>_EP_PF0_VSECRAS_CAP_RASDP_CORR_ERROR_LOCATION_OFF_0

where <*j*> = 4, 8.

Description:Corrected errors locations. For more details, see the RAS Data Protection (DP) section in the Core Operations chapter of the Databook.

PCIE_X4_EP_PF0_VSECRAS_CAP_RASDP_CORR_ERROR_LOCATION_OFF_0

Offset: 0x2dc

PCIE_X8_EP_PF0_VSECRAS_CAP_RASDP_CORR_ERROR_LOCATION_OFF_0

Offset: 0x2f0

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00e000e0 (0b0000,0000,1110,0000,0000,0000,1110,0000)

Bit	Reset	Description
31:24	0x0	**LOC_LAST_CORR_ERROR**: Location/ID of the last corrected error within the region defined by REG_LAST_CORR_ERROR. You can cycle this field value from 0 to 255 to access all counters.

Bit	Reset	Description
23:20	0xe	**REG_LAST_CORR_ERROR**: Region of the last corrected error. All encodings other than the defined encodings are reserved.
19:16	0x0	**RSVDP_16**: Reserved for future use.
15:8	0x0	**LOC_FIRST_CORR_ERROR**: Location/ID of the first corrected error within the region defined by REG_FIRST_CORR_ERROR. You can cycle this field value from 0 to 255 to access all counters.
7:4	0xe	**REG_FIRST_CORR_ERROR**: Region of the first corrected error. All encodings other than the defined encodings are reserved.
3:0	0x0	**RSVDP_0**: Reserved for future use.

PCIE_X<*j*>_EP_PF0_VSECRAS_CAP_RASDP_UNCORR_ERROR_LOCATION_OFF_0

where <*j*> = 4, 8.

Description:Uncorrected errors locations. For more details, see the RAS Data Protection (DP) section in the Core Operations chapter of the Databook.

PCIE_X4_EP_PF0_VSECRAS_CAP_RASDP_UNCORR_ERROR_LOCATION_OFF_0

Offset: 0x2e0

PCIE_X8_EP_PF0_VSECRAS_CAP_RASDP_UNCORR_ERROR_LOCATION_OFF_0

Offset: 0x2f4

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00e000e0 (0b0000,0000,1110,0000,0000,0000,1110,0000)

Bit	Reset	Description
31:24	0x0	**LOC_LAST_UNCORR_ERROR**: Location/ID of the last uncorrected error within the region defined by REG_LAST_UNCORR_ERROR. You can cycle this field value from 0 to 255 to access all counters.
23:20	0xe	**REG_LAST_UNCORR_ERROR**: Region of the last uncorrected error. All encodings other than the defined encodings are reserved.

Bit	Reset	Description
19:16	0x0	**RSVDP_16**: Reserved for future use.
15:8	0x0	**LOC_FIRST_UNCORR_ERROR**: Location/ID of the first uncorrected error within the region defined by REG_FIRST_UNCORR_ERROR. You can cycle this field value from 0 to 255 to access all counters.
7:4	0xe	**REG_FIRST_UNCORR_ERROR**: Region of the first uncorrected error. All encodings other than the defined encodings are reserved.
3:0	0x0	**RSVDP_0**: Reserved for future use.

PCIE_X<*j*>_EP_PF0_VSECRAS_CAP_RASDP_ERROR_MODE_EN_OFF_0

where <*j*> = 4, 8.

Description:RASDP error mode enable. The controller enters RASDP error mode (if ERROR_MODE_EN =1) upon detection of the first uncorrectable error. During this mode: - Rx TLPs that are forwarded to your application are not guaranteed to be correct; you must discard them. For more details, see the RAS Data Protection (DP) section in the Core Operations chapter of the Databook.

PCIE_X4_EP_PF0_VSECRAS_CAP_RASDP_ERROR_MODE_EN_OFF_0

Offset: 0x2e4

PCIE_X8_EP_PF0_VSECRAS_CAP_RASDP_ERROR_MODE_EN_OFF_0

Offset: 0x2f8

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000001 (0b0000,0000,0000,0000,0000,0000,0000,0001)

Bit	R/W	Reset	Description
31:2	RO	0x0	**RSVDP_2**: Reserved for future use.
1	RW	0x0	**AUTO_LINK_DOWN_EN**: Write '1' to enable the controller to bring the link down when the controller enters RASDP error mode. Note: This register field is sticky.

Bit	R/W	Reset	Description
0	RW	0x1	**ERROR_MODE_EN**: Write '1' to enable the controller enter RASDP error mode when it detects an uncorrectable error. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_VSECRAS_CAP_RASDP_ERROR_MODE_CLEAR_OFF_0

where <*j*> = 4, 8.

Description:Exit RASDP error mode. For more details, see the RAS Data Protection (DP) section in the Core Operations chapter of the Databook.

PCIE_X4_EP_PF0_VSECRAS_CAP_RASDP_ERROR_MODE_CLEAR_OFF_0

Offset: 0x2e8

PCIE_X8_EP_PF0_VSECRAS_CAP_RASDP_ERROR_MODE_CLEAR_OFF_0

Offset: 0x2fc

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:1	RO	0x0	**RSVDP_1**: Reserved for future use.
0	WO	0x0	**ERROR_MODE_CLEAR**: Write '1' to take the controller out of RASDP error mode. The controller will then report uncorrectable errors (through AER internal error reporting) and also stop nullifying/discarding TLPs.

PCIE_X<*j*>_EP_PF0_VSECRAS_CAP_RASDP_RAM_ADDR_CORR_ERROR_OFF_0

where <*j*> = 4, 8.

Description:For more details, see the RAS Data Protection (DP) section in the Core Operations chapter of the Databook.

PCIE_X4_EP_PF0_VSECRAS_CAP_RASDP_RAM_ADDR_CORR_ERROR_OFF_0

Offset: 0x2ec

PCIE_X8_EP_PF0_VSECRAS_CAP_RASDP_RAM_ADDR_CORR_ERROR_OFF_0

Offset: 0x300

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**RAM_INDEX_CORR_ERROR**: RAM index where a corrected error (1-bit ECC) is detected.
27	0x0	**RSVDP_27**: Reserved for future use.
26:0	0x0	**RAM_ADDR_CORR_ERROR**: RAM Address where a corrected error (1-bit ECC) is detected.

PCIE_X<*j*>_EP_PF0_VSECRAS_CAP_RASDP_RAM_ADDR_UNCORR_ERROR_OFF_0

where <*j*> = 4, 8.

Description:For more details, see the RAS Data Protection (DP) section in the Core Operations chapter of the Databook.

PCIE_X4_EP_PF0_VSECRAS_CAP_RASDP_RAM_ADDR_UNCORR_ERROR_OFF_0

Offset: 0x2f0

PCIE_X8_EP_PF0_VSECRAS_CAP_RASDP_RAM_ADDR_UNCORR_ERROR_OFF_0

Offset: 0x304

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**RAM_INDEX_UNCORR_ERROR**: RAM index where an uncorrected error (2-bit ECC) is detected.

Bit	Reset	Description
27	0x0	**RSVDP_27**: Reserved for future use.
26:0	0x0	**RAM_ADDR_UNCORR_ERROR**: RAM Address where an uncorrected error (2-bit ECC) is detected.

PCIE_X<*j*>_EP_PF0_DLINK_CAP_DATA_LINK_FEATURE_EXT_HDR_OFF_0

where <*j*> = 4, 8.

Description:This register provides capability ID, capability version, and next offset value.

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X4_EP_PF0_DLINK_CAP_DATA_LINK_FEATURE_EXT_HDR_OFF_0

Offset: 0x2f4
Reset: 0x30010025 (0b0011,0000,0000,0001,0000,0000,0010,0101)

Bit	Reset	Description
31:20	0x300	**DLINK_NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h, if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. The bottom 2 bits of this offset are Reserved and must be implemented as 00b although software must mask them to allow for future uses of these bits. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**DLINK_CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. This field depends on the version of the specification. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x25	**DLINK_EXT_CAP_ID**: Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. Extended Capability ID for Data Link Feature is 0025h. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X8_EP_PF0_DLINK_CAP_DATA_LINK_FEATURE_EXT_HDR_OFF_0

Offset: 0x308
Reset: 0x31410025 (0b0011,0001,0100,0001,0000,0000,0010,0101)

Bit	Reset	Description
31:20	0x314	**DLINK_NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h, if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. The bottom 2 bits of this offset are Reserved and must be implemented as 00b although software must mask them to allow for future uses of these bits. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**DLINK_CAP_VERSION**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. This field depends on the version of the specification. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x25	**DLINK_EXT_CAP_ID**: Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. Extended Capability ID for Data Link Feature is 0025h. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_DLINK_CAP_DATA_LINK_FEATURE_CAP_OFF_0

where <*j*> = 4, 8.

Description:This register provides description about extended feature.

PCIE_X4_EP_PF0_DLINK_CAP_DATA_LINK_FEATURE_CAP_OFF_0

Offset: 0x2f8

PCIE_X8_EP_PF0_DLINK_CAP_DATA_LINK_FEATURE_CAP_OFF_0

Offset: 0x30c

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x80000001 (0b1000,0000,0000,0000,0000,0000,0000,0001)

Bit	R/W	Reset	Description
31	RW	0x1	**DL_FEATURE_EXCHANGE_EN**: Data Link Feature Exchange Enable. If Set, this bit indicates that this Port will enter the DL_Feature negotiation state. Note: The access attributes of this field are as follows: - Wire: HWINIT - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky)
30:23	RO	0x0	**RSVDP_23**: Reserved for future use.
22:1	RW	0x0	**FUTURE_FEATURE_SUPPORTED**: Local Future Data Link Feature Supported. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
0	RW	0x1	**SCALED_FLOW_CNTL_SUPPORTED**: Local Scaled Flow Control Supported. - Bit 0: Local Scaled Flow Control Supported - Bits [22:1]: RsvdP Bits associated with features that this Port is capable of supporting are HwInit, defaulting to 1b. Note: The access attributes of this field are as follows: - Wire: HWINIT - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky)

PCIE_X<*j*>_EP_PF0_DLINK_CAP_DATA_LINK_FEATURE_STATUS_OFF_0

where <*j*> = 4, 8.

Description:This register provides status of the capability of data link feature.

PCIE_X4_EP_PF0_DLINK_CAP_DATA_LINK_FEATURE_STATUS_OFF_0

Offset: 0x2fc

PCIE_X8_EP_PF0_DLINK_CAP_DATA_LINK_FEATURE_STATUS_OFF_0

Offset: 0x310

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**DATA_LINK_FEATURE_STATUS_VALID**: Remote Data Link Feature Supported Valid. This field indicates that the Port has received a Data Link Feature DLLP in state DL_Feature and that the Remote Data Link Feature Supported and Remote Data Link Feature Ack fields are meaningful. This field is Cleared on entry to state DL_Inactive.

Bit	Reset	Description
30:23	0x0	**RSVDP_23**: Reserved for future use.
22:0	0x0	**REMOTE_DATA_LINK_FEATURE_SUPPORTED**: Remote Data Link Feature Supported. - Bit 0: Remote Scaled Flow Control Supported - Bits [22:1]: Undefined

PCIE_X<*j*>_EP_PF0_PTM_CAP_PTM_EXT_CAP_HDR_OFF_0

where <*j*> = 4, 8.

Description:This Register provides Capability ID, Cabability Version, and Next Offset of Precision Time Measurement Capability structure.

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X4_EP_PF0_PTM_CAP_PTM_EXT_CAP_HDR_OFF_0

Offset: 0x300
Reset: 0x30c1001f (0b0011,0000,1100,0001,0000,0000,0001,1111)

Bit	Reset	Description
31:20	0x30c	**PTM_NEXT_OFFSET**: Precision Time Measurement PCI Express Extended Capability Next Offset. This field contains the offset to the next PCI Express Extended Capability structure or 000h if no other items exist in the linked list of Capabilities. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**PTM_CAP_VERSION**: Precision Time Measurement PCI Express Extended Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. This bit depends on the version of the specification. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x1f	**PTM_CAP_ID**: Precision Time Measurement PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. PCI Express Extended Capability ID for the Precision Time Measurement Capability is 001Fh. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X8_EP_PF0_PTM_CAP_PTM_EXT_CAP_HDR_OFF_0

Offset: 0x314
Reset: 0x3201001f (0b0011,0010,0000,0001,0000,0000,0001,1111)

Bit	Reset	Description
31:20	0x320	**PTM_NEXT_OFFSET**: Precision Time Measurement PCI Express Extended Capability Next Offset. This field contains the offset to the next PCI Express Extended Capability structure or 000h if no other items exist in the linked list of Capabilities. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**PTM_CAP_VERSION**: Precision Time Measurement PCI Express Extended Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. This bit depends on the version of the specification. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x1f	**PTM_CAP_ID**: Precision Time Measurement PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. PCI Express Extended Capability ID for the Precision Time Measurement Capability is 001Fh. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PTM_CAP_PTM_CAP_OFF_0

where <*j*> = 4, 8.

Description:This register describes a Function's support for Precision Time Measurement. Not all fields within this register apply to all Functions capable of implementing PTM.

PCIE_X4_EP_PF0_PTM_CAP_PTM_CAP_OFF_0

Offset: 0x304

PCIE_X8_EP_PF0_PTM_CAP_PTM_CAP_OFF_0

Offset: 0x318

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00001007 (0b0000,0000,0000,0000,0001,0000,0000,0111)

Bit	R/W	Reset	Description
31:16	RO	0x0	**RSVDP_16**: Reserved for future use.
15:8	RW	0x10	**PTM_CLK_GRAN**: PTM Local Clock Granularity. For other than this value (between b00000001-b11111110) indicates the period of this Time Source's local clock in ns. If the PTM Root Select bit is Set, this local clock is used to provide PTM Master Time. Otherwise, the Time Source uses this local clock to locally track PTM Master Time received from further Upstream within a PTM Hierarchy. This field is reserved for Functions that do not implement the PTM Time Source role. Note: The access attributes of this field are as follows: - Wire: RSVDP - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky)
7:3	RO	0x0	**RSVDP_3**: Reserved for future use.
2	RW	0x1	**PTM_ROOT_CAPABLE**: PTM Root Capable. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
1	RW	0x1	**PTM_RES_CAPABLE**: PTM Responder Capable. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
0	RW	0x1	**PTM_REQ_CAPABLE**: PTM Requester Capable. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PTM_CAP_PTM_CONTROL_OFF_0

where <*j*> = 4, 8.

Description:This register controls a Function's participation in the Precision Time Measurement mechanism. Not all fields within this register apply to all Functions capable of implementing PTM.

PCIE_X4_EP_PF0_PTM_CAP_PTM_CONTROL_OFF_0

Offset: 0x308

PCIE_X8_EP_PF0_PTM_CAP_PTM_CONTROL_OFF_0

Offset: 0x31c

Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:16	RO	0x0	**RSVDP_16**: Reserved for future use.
15:8	RO	0x0	**EFF_GRAN**: PTM Effective Granularity. For Functions implementing the PTM Requester Role, this field provides information relating to the expected accuracy of the PTM clock, but does not otherwise affect the PTM mechanism. For Endpoints, system software must program this field to the value representing the maximum Local Clock Granularity reported by the PTM Root and all intervening PTM Time Sources. For RCiEPs, system software must set this field to the value reported in the Local Clock Granularity field by the associated PTM Time Source. For other than this value (between b00000001-b11111110) indicates the effective PTM granularity in ns. Note: The access attributes of this field are as follows: - Wire: HWINIT - Dbi: HWINIT
7:2	RO	0x0	**RSVDP_2**: Reserved for future use.
1	RO	0x0	**ROOT_SELECT**: PTM Root Select. If the value of the PTM Root Capable bit is 0b, this bit is permitted to be hardwired to 0b by the controller. Within each PTM Hierarchy, it is recommended that system software select only the furthest Upstream Time Source to be the PTM Root. Note: The access attributes of this field are as follows: - Wire: HWINIT - Dbi: HWINIT
0	RW	0x0	**PTM_ENABLE**: PTM Enable.

PCIE_X<*j*>_EP_PF0_PTM_REQ_CAP_PTM_REQ_CAP_HDR_OFF_0

where <*j*> = 4, 8.

Description:This Register provides Capability ID, Cabability Version, and Next Offset of Precision Time Measurement Capability structure.

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X4_EP_PF0_PTM_REQ_CAP_PTM_REQ_CAP_HDR_OFF_0

Offset: 0x30c
Reset: 0x3741000b (0b0011,0111,0100,0001,0000,0000,0000,1011)

Bit	Reset	Description
31:20	0x374	**PTM_REQ_EXT_CAP_NEXT_OFFS**: Precision Time Measurement Requester VSEC Next Pointer. For more details, see the PTM section in the Databook. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**PTM_REQ_EXT_CAP_VER**: Precision Time Measurement Requester VSEC Version. For more details, see the PTM section in the Databook. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0xb	**PTM_REQ_EXT_CAP_ID**: Precision Time Measurement Requester VSEC ID. For more details, see the PTM section in the Databook. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X8_EP_PF0_PTM_REQ_CAP_PTM_REQ_CAP_HDR_OFF_0

Offset: 0x320
Reset: 0x3881000b (0b0011,1000,1000,0001,0000,0000,0000,1011)

Bit	Reset	Description
31:20	0x388	**PTM_REQ_EXT_CAP_NEXT_OFFS**: Precision Time Measurement Requester VSEC Next Pointer. For more details, see the PTM section in the Databook. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**PTM_REQ_EXT_CAP_VER**: Precision Time Measurement Requester VSEC Version. For more details, see the PTM section in the Databook. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0xb	**PTM_REQ_EXT_CAP_ID**: Precision Time Measurement Requester VSEC ID. For more details, see the PTM section in the Databook. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PTM_REQ_CAP_PTM_REQ_HDR_OFF_0

where <*j*> = 4, 8.

Description:For more details, see the PTM section in the Databook.

PCIE_X4_EP_PF0_PTM_REQ_CAP_PTM_REQ_HDR_OFF_0

Offset: 0x310

PCIE_X8_EP_PF0_PTM_REQ_CAP_PTM_REQ_HDR_OFF_0

Offset: 0x324

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x05410003 (0b0000,0101,0100,0001,0000,0000,0000,0011)

Bit	Reset	Description
31:20	0x54	**PTM_REQ_VSEC_LENGTH**: PTM Requester VSEC Length. For more details, see the PTM section in the Databook. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19:16	0x1	**PTM_REQ_VSEC_REV**: PTM Requester VSEC Revision. For more details, see the PTM section in the Databook. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0x3	**PTM_REQ_VSEC_ID**: PTM Requester VSEC ID. For more details, see the PTM section in the Databook. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PTM_REQ_CAP_PTM_REQ_CONTROL_OFF_0

where <*j*> = 4, 8.

Description:For more details, see the PTM section in the Databook.

PCIE_X4_EP_PF0_PTM_REQ_CAP_PTM_REQ_CONTROL_OFF_0

Offset: 0x314

PCIE_X8_EP_PF0_PTM_REQ_CAP_PTM_REQ_CONTROL_OFF_0

Offset: 0x328

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000900 (0b0000,0000,0000,0000,0000,1001,0000,0000)

Bit	R/W	Reset	Description
31:17	RO	0x0	**RSVDP_17**: Reserved for future use.
16	RW	0x0	**PTM_REQ_PDEL_BYTE_REV**: PTM Requester Propagation Delay Byte Reversal. Reverse the order of bytes in the PTM Propagation Delay data word of the PTM ResponseD Message, for compatibility with previous revisions. For more details, see the PTM section in the Databook. Note: This register field is sticky.
15:8	RW	0x9	**PTM_REQ_LONG_TIMER**: PTM Requester Long Timer. Determines the period between each auto update PTM Dialogue in miliseconds. Update period is the register value +1 milisecond. For the Switch product this value must not be set larger than 0x9 for spec compliance. For more details, see the PTM section in the Databook. Note: This register field is sticky.
7:3	RO	0x0	**RSVDP_3**: Reserved for future use.
2	RW	0x0	**PTM_REQ_FAST_TIMERS**: PTM Fast Timers (Debug mode for PTM Timers). The 100us timer output will go high at 30us and the 10ms timer output will go high at 100us (the Long Timer Value is ignored). There is no change to the 1us timer. The requester operation will otherwise remain the same. For more details, see the PTM section in the Databook. Note: This register field is sticky.
1	RW	0x0	**PTM_REQ_START_UPDATE**: PTM Requester Start Update. For more details, see the PTM section in the Databook.
0	RW	0x0	**PTM_REQ_AUTO_UPDATE_ENABLED**: PTM Requester Auto Update Enabled. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PTM_REQ_CAP_PTM_REQ_STATUS_OFF_0

where <*j*> = 4, 8.

Description:For more details, see the PTM section in the Databook.

PCIE_X4_EP_PF0_PTM_REQ_CAP_PTM_REQ_STATUS_OFF_0

Offset: 0x318

PCIE_X8_EP_PF0_PTM_REQ_CAP_PTM_REQ_STATUS_OFF_0

Offset: 0x32c

Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:2	0x0	**RSVDP_2**: Reserved for future use.
1	0x0	**PTM_REQ_MANUAL_UPDATE_ALLOWED**: PTM Requester Manual Update Allowed. Indicates whether or not a Manual Update can be signalled. For more details, see the PTM section in the Databook.
0	0x0	**PTM_REQ_CONTEXT_VALID**: PTM Requester Context Valid. Indicate that the Timing Context is valid. For more details, see the PTM section in the Databook.

PCIE_X<*j*>_EP_PF0_PTM_REQ_CAP_PTM_REQ_LOCAL_LSB_OFF_0

where <*j*> = 4, 8.

Description:For more details, see the PTM section in the Databook.

PCIE_X4_EP_PF0_PTM_REQ_CAP_PTM_REQ_LOCAL_LSB_OFF_0

Offset: 0x31c

PCIE_X8_EP_PF0_PTM_REQ_CAP_PTM_REQ_LOCAL_LSB_OFF_0

Offset: 0x330

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PTM_REQ_LOCAL_LSB**: PTM Requester Local Clock LSB. Lower 32 bits of local timer value. For more details, see the PTM section in the Databook. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PTM_REQ_CAP_PTM_REQ_LOCAL_MSB_OFF_0

where <*j*> = 4, 8.

Description:For more details, see the PTM section in the Databook.

PCIE_X4_EP_PF0_PTM_REQ_CAP_PTM_REQ_LOCAL_MSB_OFF_0

Offset: 0x320

PCIE_X8_EP_PF0_PTM_REQ_CAP_PTM_REQ_LOCAL_MSB_OFF_0

Offset: 0x334

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PTM_REQ_LOCAL_MSB**: PTM Requester Local Clock MSB. Upper 32 bits of local timer value. For more details, see the PTM section in the Databook. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PTM_REQ_CAP_PTM_REQ_T1_LSB_OFF_0

where <*j*> = 4, 8.

Description:For more details, see the PTM section in the Databook.

PCIE_X4_EP_PF0_PTM_REQ_CAP_PTM_REQ_T1_LSB_OFF_0

Offset: 0x324

PCIE_X8_EP_PF0_PTM_REQ_CAP_PTM_REQ_T1_LSB_OFF_0

Offset: 0x338

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PTM_REQ_T1_LSB**: PTM Requester T1 Timestamp LSB. Lower 32 bits of the T1 Timestamp value. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PTM_REQ_CAP_PTM_REQ_T1_MSB_OFF_0

where <*j*> = 4, 8.

Description:For more details, see the PTM section in the Databook.

PCIE_X4_EP_PF0_PTM_REQ_CAP_PTM_REQ_T1_MSB_OFF_0

Offset: 0x328

PCIE_X8_EP_PF0_PTM_REQ_CAP_PTM_REQ_T1_MSB_OFF_0

Offset: 0x33c

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PTM_REQ_T1_MSB**: PTM Requester T1 Timestamp MSB. Upper 32 bits of the T1 Timestamp value. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PTM_REQ_CAP_PTM_REQ_T1P_LSB_OFF_0

where <*j*> = 4, 8.

Description:For more details, see the PTM section in the Databook.

PCIE_X4_EP_PF0_PTM_REQ_CAP_PTM_REQ_T1P_LSB_OFF_0

Offset: 0x32c

PCIE_X8_EP_PF0_PTM_REQ_CAP_PTM_REQ_T1P_LSB_OFF_0

Offset: 0x340

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PTM_REQ_T1P_LSB**: PTM Requester T1 Previous Timestamp LSB. Lower 32 bits of the previously stored T1 Timestamp value. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PTM_REQ_CAP_PTM_REQ_T1P_MSB_OFF_0

where <*j*> = 4, 8.

Description:For more details, see the PTM section in the Databook.

PCIE_X4_EP_PF0_PTM_REQ_CAP_PTM_REQ_T1P_MSB_OFF_0

Offset: 0x330

PCIE_X8_EP_PF0_PTM_REQ_CAP_PTM_REQ_T1P_MSB_OFF_0

Offset: 0x344

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PTM_REQ_T1P_MSB**: PTM Requester T1 Previous Timestamp MSB. Upper 32 bits of the previously stored T1 Timestamp value. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PTM_REQ_CAP_PTM_REQ_T4_LSB_OFF_0

where <*j*> = 4, 8.

Description:For more details, see the PTM section in the Databook.

PCIE_X4_EP_PF0_PTM_REQ_CAP_PTM_REQ_T4_LSB_OFF_0

Offset: 0x334

PCIE_X8_EP_PF0_PTM_REQ_CAP_PTM_REQ_T4_LSB_OFF_0

Offset: 0x348

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PTM_REQ_T4_LSB**: PTM Requester T4 Timestamp LSB. Lower 32 bits of the T4 Timestamp value. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PTM_REQ_CAP_PTM_REQ_T4_MSB_OFF_0

where <*j*> = 4, 8.

Description:For more details, see the PTM section in the Databook.

PCIE_X4_EP_PF0_PTM_REQ_CAP_PTM_REQ_T4_MSB_OFF_0

Offset: 0x338

PCIE_X8_EP_PF0_PTM_REQ_CAP_PTM_REQ_T4_MSB_OFF_0

Offset: 0x34c

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PTM_REQ_T4_MSB**: PTM Requester T4 Timestamp MSB. Upper 32 bits of the T4 Timestamp value. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PTM_REQ_CAP_PTM_REQ_T4P_LSB_OFF_0

where <*j*> = 4, 8.

Description:For more details, see the PTM section in the Databook.

PCIE_X4_EP_PF0_PTM_REQ_CAP_PTM_REQ_T4P_LSB_OFF_0

Offset: 0x33c

PCIE_X8_EP_PF0_PTM_REQ_CAP_PTM_REQ_T4P_LSB_OFF_0

Offset: 0x350

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PTM_REQ_T4P_LSB**: PTM Requester T4 Previous Timestamp LSB. Lower 32 bits of the previously stored T4 Timestamp value. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PTM_REQ_CAP_PTM_REQ_T4P_MSB_OFF_0

where <*j*> = 4, 8.

Description:For more details, see the PTM section in the Databook.

PCIE_X4_EP_PF0_PTM_REQ_CAP_PTM_REQ_T4P_MSB_OFF_0

Offset: 0x340

PCIE_X8_EP_PF0_PTM_REQ_CAP_PTM_REQ_T4P_MSB_OFF_0

Offset: 0x354

Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PTM_REQ_T4P_MSB**: PTM Requester T4 Previous Timestamp MSB. Upper 32 bits of the previously stored T4 Timestamp value. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PTM_REQ_CAP_PTM_REQ_MASTER_LSB_OFF_0

where <*j*> = 4, 8.

Description:For more details, see the PTM section in the Databook.

PCIE_X4_EP_PF0_PTM_REQ_CAP_PTM_REQ_MASTER_LSB_OFF_0

Offset: 0x344

PCIE_X8_EP_PF0_PTM_REQ_CAP_PTM_REQ_MASTER_LSB_OFF_0

Offset: 0x358

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PTM_REQ_MASTER_LSB**: PTM Requester Master Time LSB. Lower 32 bits of Master Time value sent from PTM Responder. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PTM_REQ_CAP_PTM_REQ_MASTER_MSB_OFF_0

where <*j*> = 4, 8.

Description:For more details, see the PTM section in the Databook.

PCIE_X4_EP_PF0_PTM_REQ_CAP_PTM_REQ_MASTER_MSB_OFF_0

Offset: 0x348

PCIE_X8_EP_PF0_PTM_REQ_CAP_PTM_REQ_MASTER_MSB_OFF_0

Offset: 0x35c

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PTM_REQ_MASTER_MSB**: PTM Requester Master Time MSB. Upper 32 bits of Master Time value sent from PTM Responder. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PTM_REQ_CAP_PTM_REQ_PROP_DELAY_OFF_0

where <*j*> = 4, 8.

Description:For more details, see the PTM section in the Databook.

PCIE_X4_EP_PF0_PTM_REQ_CAP_PTM_REQ_PROP_DELAY_OFF_0

Offset: 0x34c

PCIE_X8_EP_PF0_PTM_REQ_CAP_PTM_REQ_PROP_DELAY_OFF_0

Offset: 0x360

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PTM_REQ_PROP_DELAY**: PTM Requester Propagation Delay. Propagation Delay sent from the PTM Responder. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PTM_REQ_CAP_PTM_REQ_MASTERT1_LSB_OFF_0

where <*j*> = 4, 8.

Description:For more details, see the PTM section in the Databook.

PCIE_X4_EP_PF0_PTM_REQ_CAP_PTM_REQ_MASTERT1_LSB_OFF_0

Offset: 0x350

PCIE_X8_EP_PF0_PTM_REQ_CAP_PTM_REQ_MASTERT1_LSB_OFF_0

Offset: 0x364

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PTM_REQ_MASTERT1_LSB**: PTM Requester Master Time at T1 LSB. Lower 32 bits of the calculated Master Time at T1 prime. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PTM_REQ_CAP_PTM_REQ_MASTERT1_MSB_OFF_0

where <*j*> = 4, 8.

Description:For more details, see the PTM section in the Databook.

PCIE_X4_EP_PF0_PTM_REQ_CAP_PTM_REQ_MASTERT1_MSB_OFF_0

Offset: 0x354

PCIE_X8_EP_PF0_PTM_REQ_CAP_PTM_REQ_MASTERT1_MSB_OFF_0

Offset: 0x368

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PTM_REQ_MASTERT1_MSB**: PTM Requester Master Time at T1 MSB. Upper 32 bits of the calculated Master Time at T1 prime. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PTM_REQ_CAP_PTM_REQ_TX_LATENCY_OFF_0

where <*j*> = 4, 8.

Description:For more details, see the PTM section in the Databook.

PCIE_X4_EP_PF0_PTM_REQ_CAP_PTM_REQ_TX_LATENCY_OFF_0

Offset: 0x358

PCIE_X8_EP_PF0_PTM_REQ_CAP_PTM_REQ_TX_LATENCY_OFF_0

Offset: 0x36c

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000018 (0b0000,0000,0000,0000,0000,0000,0001,1000)

Bit	R/W	Reset	Description
31:12	RO	0x0	**RSVDP_12**: Reserved for future use.
11:0	RW	0x18	**PTM_REQ_TX_LATENCY**: PTM Requester TX Latency. Requester Transmit path latency viewport register. A register is provided for each supported link speed, and the value used for timestamp adjustment is automatically selected based on the current speed. The PTM Requester Latency Register Select register specifies the register to be written or read. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PTM_REQ_CAP_PTM_REQ_RX_LATENCY_OFF_0

where <*j*> = 4, 8.

Description:For more details, see the PTM section in the Databook.

PCIE_X4_EP_PF0_PTM_REQ_CAP_PTM_REQ_RX_LATENCY_OFF_0

Offset: 0x35c

PCIE_X8_EP_PF0_PTM_REQ_CAP_PTM_REQ_RX_LATENCY_OFF_0

Offset: 0x370

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000001f (0b0000,0000,0000,0000,0000,0000,0001,1111)

Bit	R/W	Reset	Description
31:12	RO	0x0	**RSVDP_12**: Reserved for future use.
11:0	RW	0x1f	**PTM_REQ_RX_LATENCY**: PTM Requester RX Latency - Requester Receive path latency viewport register. A register is provided for each supported link speed, and the value used for timestamp adjustment is automatically selected based on the current speed. The PTM Requester Latency Register Select register specifies the register to be written or read. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PTM_REQ_CAP_PTM_REQ_CLOCK_CORR_LSB_OFF_0

where <*j*> = 4, 8.

Description:For more details, see the PTM section in the Databook.

PCIE_X4_EP_PF0_PTM_REQ_CAP_PTM_REQ_CLOCK_CORR_LSB_OFF_0

Offset: 0x360

PCIE_X8_EP_PF0_PTM_REQ_CAP_PTM_REQ_CLOCK_CORR_LSB_OFF_0

Offset: 0x374

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PTM_REQ_CLOCK_CORR_LSB**: PTM Requester Clock Correction LSB. Lower 32 bits of amount by which PTM Requester Local Clock has been corrected. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PTM_REQ_CAP_PTM_REQ_CLOCK_CORR_MSB_OFF_0

where <*j*> = 4, 8.

Description:For more details, see the PTM section in the Databook.

PCIE_X4_EP_PF0_PTM_REQ_CAP_PTM_REQ_CLOCK_CORR_MSB_OFF_0

Offset: 0x364

PCIE_X8_EP_PF0_PTM_REQ_CAP_PTM_REQ_CLOCK_CORR_MSB_OFF_0

Offset: 0x378

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PTM_REQ_CLOCK_CORR_MSB**: PTM Requester Clock Correction MSB. Upper 32 bits of amount by which PTM Requester Local Clock has been corrected. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PTM_REQ_CAP_PTM_REQ_NOM_CLOCK_T_OFF_0

where <*j*> = 4, 8.

Description:For more details, see the PTM section in the Databook.

PCIE_X4_EP_PF0_PTM_REQ_CAP_PTM_REQ_NOM_CLOCK_T_OFF_0

Offset: 0x368

PCIE_X8_EP_PF0_PTM_REQ_CAP_PTM_REQ_NOM_CLOCK_T_OFF_0

Offset: 0x37c

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**RSVDP_24**: Reserved for future use.
23:16	0x0	**PTM_REQ_NOM_CLOCK_T_INT**: PTM Requester Nominal Clock Period Integral. Integral part of the nominal PTM local clock period. LSB is 1/(2^16) nanoseconds. For more details, see the PTM section in the Databook. Note: This register field is sticky.
15:0	0x0	**PTM_REQ_NOM_CLOCK_T_FRAC**: PTM Requester Nominal Clock Period Fractional. Fractional part of the nominal PTM local clock period, in nanoseconds. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PTM_REQ_CAP_PTM_REQ_SCALED_CLOCK_T_OFF_0

where <*j*> = 4, 8.

Description:For more details, see the PTM section in the Databook.

PCIE_X4_EP_PF0_PTM_REQ_CAP_PTM_REQ_SCALED_CLOCK_T_OFF_0

Offset: 0x36c

PCIE_X8_EP_PF0_PTM_REQ_CAP_PTM_REQ_SCALED_CLOCK_T_OFF_0

Offset: 0x380

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PTM_REQ_SCALED_CLOCK_T_EN**: PTM Requester Scaled Clock Period Enable. This bit is cleared when the core_clk rate starts to change and can only be set when the clock period change is complete. For more details, see the PTM section in the Databook. Note: This register field is sticky.

Bit	R/W	Reset	Description
30:24	RO	0x0	**RSVDP_24**: Reserved for future use.
23:16	RW	0x0	**PTM_REQ_SCALED_CLOCK_T_INT**: PTM Requester Scaled Clock Period Integral. Integral part of the nominal PTM local clock period, in nanoseconds. For more details, see the PTM section in the Databook. Note: This register field is sticky.
15:0	RW	0x0	**PTM_REQ_SCALED_CLOCK_T_FRAC**: PTM Requester Scaled Clock Period Fractional. Fractional part of the nominal PTM local clock period. LSB is 1/(2^16) nanoseconds. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PTM_REQ_CAP_PTM_REQ_LATENCY_REG_SEL_OFF_0

where <*j*> = 4, 8.

Description:For more details, see the PTM section in the Databook.

PCIE_X4_EP_PF0_PTM_REQ_CAP_PTM_REQ_LATENCY_REG_SEL_OFF_0

Offset: 0x370

PCIE_X8_EP_PF0_PTM_REQ_CAP_PTM_REQ_LATENCY_REG_SEL_OFF_0

Offset: 0x384

Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:8	RO	0x0	**RSVDP_8**: Reserved for future use.
7	RW	0x0	**PTM_REQ_ESM_SEL**: Selection Mode for PTM Requester Tx and Rx Latency Viewport register to be written or read. For more details, see the PTM section in the Databook. Note: This register field is sticky.
6:4	RO	0x0	**RSVDP_4**: Reserved for future use.

Bit	R/W	Reset	Description
3:0	RW	0x0	**PTM_REQ_LATENCY_REG_SEL**: Selects the PTM Requester Tx and Rx Latency Viewport register to be written or read, together with the PTM_REQ_ESM_SEL field. For more details, see the PTM section in the Databook. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_VSECDMA_CAP_VSECDMA_EXT_CAP_HDR_OFF_0

where <*j*> = 4, 8.

Description:This register provides capability ID, capability version, and next capability offset for PCIe extended capability structure.

PCIE_X4_EP_PF0_VSECDMA_CAP_VSECDMA_EXT_CAP_HDR_OFF_0

Offset: 0x374

PCIE_X8_EP_PF0_VSECDMA_CAP_VSECDMA_EXT_CAP_HDR_OFF_0

Offset: 0x388

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0001000b (0b0000,0000,0000,0001,0000,0000,0000,1011)

Bit	Reset	Description
31:20	0x0	**NEXT_OFFSET**: Next Capability Offset. This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. The bottom 2 bits of this offset are Reserved and must be implemented as 00b although software must mask them to allow for future uses of these bits. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

Bit	Reset	Description
19:16	0x1	**CAP**: Capability Version. This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. A version of the specification that changes the Extended Capability in a way that is not otherwise identifiable (that is, through a new Capability field) is permitted to increment this field. All such changes to the Capability structure must be software-compatible. Software must check for Capability Version numbers that are greater than or equal to the highest number defined when the software is written, as Functions reporting any such Capability Version numbers will contain a Capability structure that is compatible with that piece of software. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15:0	0xb	**ID**: PCI Express Extended Capability ID. This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_VSECDMA_CAP_VSECDMA_VENDOR_SPECIFIC_HDR_OFF_0

where <*j*> = 4, 8.

Description:This Register provides VSEC Length, VSEC ID, and VSEC Rev(Version Number). Vendor-specific software must qualify the associated Vendor ID of the PCI Express Function or RCRB before attempting to interpret the values in the VSEC ID or VSEC Rev fields.

PCIE_X4_EP_PF0_VSECDMA_CAP_VSECDMA_VENDOR_SPECIFIC_HDR_OFF_0

Offset: 0x378

PCIE_X8_EP_PF0_VSECDMA_CAP_VSECDMA_VENDOR_SPECIFIC_HDR_OFF_0

Offset: 0x38c

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x01800006 (0b0000,0001,1000,0000,0000,0000,0000,0110)

Bit	Reset	Description
31:20	0x18	**VSEC_LENGTH**: VSEC Length. This field indicates the number of bytes in the entire VSEC structure, including the Vendor-Specific Extended Capability Header, the Vendor-Specific Header, and the vendor-specific registers.

Bit	Reset	Description
19:16	0x0	**VSEC_REV**: VSEC Rev. This field is a vendor-defined version number that indicates the version of the VSEC structure. Software must qualify the Vendor ID and VSEC ID before interpreting this field.
15:0	0x6	**VSEC_ID**: VSEC ID. This field is a vendor-defined ID number that indicates the nature and format of the VSEC structure. Software must qualify the Vendor ID before interpreting this field.

PCIE_X<*j*>_EP_PF0_VSECDMA_CAP_VSECDMA_DEVICE_INFORMATION_OFF_0

where <*j*> = 4, 8.

Description:This register provides DMA and AXI Bridge implementation-specific information: - DMA Architecture: Legacy DMA or Hyper DMA (HDMA). - Register Location: Port-logic or Mapped to a Function and BAR. - Register Map: Legacy DMA or HDMA. - Channel Separation: Address distance between read and write channels. - AXI Bridge: Used or Not Used - AXI Master Bus Specification: AXI Master Bus Width, Burst Length, and Boundary Pointer Width You must use this information along with the IP version registers: - PCIE_VERSION_NUMBER_OFF - PCIE_VERSION_TYPE_OFF.

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X4_EP_PF0_VSECDMA_CAP_VSECDMA_DEVICE_INFORMATION_OFF_0

Offset: 0x37c
Reset: 0x34a80401 (0b0011,0100,1010,1000,0000,0100,0000,0001)

Bit	Reset	Description
31:30	0x0	**RSVDP_30**: Reserved for future use.
29:26	0xd	**MASTER_PAGE_BOUNDARY_POINTER_WIDTH**: This field provides address page boundary information. It reports the value of CC_MSTR_PAGE_BOUNDARY_PW configuration parameter.
25:23	0x1	**MASTER_BURST_LENGTH**: Reports the CC_MSTR_BURST_LEN configuration parameter.
22:20	0x2	**MASTER_BUS_WIDTH**: This field provides information regarding the AXI master data bus width. It reports the value of MASTER_BUS_DATA_WIDTH configuration parameter.
19	0x1	**AXI**: This field provides information about AXI interface usage. It reports the value of AXI_POPULATED configuration parameter.

Bit	Reset	Description
18:16	0x0	**CHANNEL_SEPARATION**: If the MAP_FORMAT is set to HDMA_NATIVE, this field specifies the read write channel address separation. Other values are reserved.
15:11	0x0	**PFN**: Physical Function Number. This field provides information regarding the DMA register and physical function mapping.
10:8	0x4	**BARN**: Bar Number. This field provides information regarding the DMA register and BAR number mapping.
7:3	0x0	**RSVDP_3**: Reserved for future use.
2:0	0x1	**MAP_FORMAT**: Defines the register map format and features to be one of the following values: Other values are reserved.

PCIE_X8_EP_PF0_VSECDMA_CAP_VSECDMA_DEVICE_INFORMATION_OFF_0

Offset: 0x390
Reset: 0x34380401 (0b0011,0100,0011,1000,0000,0100,0000,0001)

Bit	Reset	Description
31:30	0x0	**RSVDP_30**: Reserved for future use.
29:26	0xd	**MASTER_PAGE_BOUNDARY_POINTER_WIDTH**: This field provides address page boundary information. It reports the value of CC_MSTR_PAGE_BOUNDARY_PW configuration parameter.
25:23	0x0	**MASTER_BURST_LENGTH**: Reports the CC_MSTR_BURST_LEN configuration parameter.
22:20	0x3	**MASTER_BUS_WIDTH**: This field provides information regarding the AXI master data bus width. It reports the value of MASTER_BUS_DATA_WIDTH configuration parameter.
19	0x1	**AXI**: This field provides information about AXI interface usage. It reports the value of AXI_POPULATED configuration parameter.
18:16	0x0	**CHANNEL_SEPARATION**: If the MAP_FORMAT is set to HDMA_NATIVE, this field specifies the read write channel address separation. Other values are reserved.
15:11	0x0	**PFN**: Physical Function Number. This field provides information regarding the DMA register and physical function mapping.
10:8	0x4	**BARN**: Bar Number. This field provides information regarding the DMA register and BAR number mapping.
7:3	0x0	**RSVDP_3**: Reserved for future use.

Bit	Reset	Description
2:0	0x1	**MAP_FORMAT**: Defines the register map format and features to be one of the following values: Other values are reserved.

PCIE_X<*j*>_EP_PF0_VSECDMA_CAP_VSECDMA_NUM_CHAN_OFF_0

where <*j*> = 4, 8.

Description:This register specifies the number of read and write channels implemented.

PCIE_X4_EP_PF0_VSECDMA_CAP_VSECDMA_NUM_CHAN_OFF_0

Offset: 0x380

PCIE_X8_EP_PF0_VSECDMA_CAP_VSECDMA_NUM_CHAN_OFF_0

Offset: 0x394

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00020004 (0b0000,0000,0000,0010,0000,0000,0000,0100)

Bit	Reset	Description
31:26	0x0	**RSVDP_26**: Reserved for future use.
25:16	0x2	**NUM_DMA_RD_CHAN**: This field provides information regarding the number of implemented read channels. It reports the value of CC_NUM_DMA_RD_CHAN parameter.
15:10	0x0	**RSVDP_10**: Reserved for future use.
9:0	0x4	**NUM_DMA_WR_CHAN**: This field provides information regarding the number of implemented write channels. It reports the value of CC_NUM_DMA_WR_CHAN parameter.

PCIE_X<*j*>_EP_PF0_VSECDMA_CAP_VSECDMA_UNROLL_ADDR_OFFSET_LOW_OFF_0

where <*j*> = 4, 8.

Description:This register specifies the lower 32 bits of the offset of the start address of the DMA register map. Applicable only if MAP_FORMAT >0, that is, all map formats other than EDMA_LEGACY_PL.

PCIE_X4_EP_PF0_VSECDMA_CAP_VSECDMA_UNROLL_ADDR_OFFSET_LOW_OFF_0

Offset: 0x384

PCIE_X8_EP_PF0_VSECDMA_CAP_VSECDMA_UNROLL_ADDR_OFFSET_LOW_OFF_0

```
 VSECDMA_UNROLL_ADDR_OFFSET_LOW_OFF offset(decimal)=920/0x398 size=32    (32'bit reset value:
0x0 Description: This register specifies the lower 32 bits of the offset of the start address
of the DMA register map. Applicable only if MAP_FORMAT >0, that is, all map formats other than
EDMA_LEGACY_PL. )
```

Offset: 0x398

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UNROLL_ADDR_OFFSET_LOW**: BAR address offset, 32-bit LSB.

PCIE_X<*j*>_EP_PF0_VSECDMA_CAP_VSECDMA_UNROLL_ADDR_OFFSET_HIGH_OFF_0

where <*j*> = 4, 8.

Description:This register specifies the upper 32 bits of the offset of the start address of the DMA register map. Applicable only if MAP_FORMAT >0, that is, all map formats other than EDMA_LEGACY_PL.

PCIE_X4_EP_PF0_VSECDMA_CAP_VSECDMA_UNROLL_ADDR_OFFSET_HIGH_OFF_0

Offset: 0x388

PCIE_X8_EP_PF0_VSECDMA_CAP_VSECDMA_UNROLL_ADDR_OFFSET_HIGH_OFF_0

```
 VSECDMA_UNROLL_ADDR_OFFSET_HIGH_OFF offset(decimal)=924/0x39c size=32    (32'bit reset value:
0x0 Description: This register specifies the upper 32 bits of the offset of the start address
of the DMA register map. Applicable only if MAP_FORMAT >0, that is, all map formats other than
EDMA_LEGACY_PL. )
```

Offset: 0x39c

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UNROLL_ADDR_OFFSET_HIGH**: BAR address offset, 32-bit MSB.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_ACK_LATENCY_TIMER_OFF_0

where <*j*> = 4, 8.

Description:This register holds the ack latency timer limit and replay timer limit values.

PCIE_X4_EP_PF0_PORT_LOGIC_ACK_LATENCY_TIMER_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_ACK_LATENCY_TIMER_OFF_0

Offset: 0x700
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0c23040b (0b0000,1100,0010,0011,0000,0100,0000,1011)

Bit	Reset	Description
31:16	0xc23	**REPLAY_TIME_LIMIT**: Replay Timer Limit. The replay timer expires when it reaches this limit. The controller initiates a replay upon reception of a NAK or when the replay timer expires. For more details, see "Transmit Replay" in the Databook. - You can modify the effective timer limit through the TIMER_MOD_REPLAY_TIMER field of the TIMER_CTRL_MAX_FUNC_NUM_OFF register. - After reset, the controller updates the default according to the Negotiated Link Width, Max_Payload_Size, and speed. The value is determined from Tables 3-4, 3-5, and 3-6 of the PCI Express Base Specification. - If there is a change in the payload size or link speed, the controller overrides any value that you have written to this register field, and resets the field back to the specification-defined value. The controller does not change the value in the TIMER_MOD_REPLAY_TIMER field of the TIMER_CTRL_MAX_FUNC_NUM_OFF register.

Bit	Reset	Description
15:0	0x40b	**ROUND_TRIP_LATENCY_TIME_LIMIT**: Ack Latency Timer Limit. The Ack latency timer expires when it reaches this limit. For more details, see "ACK/NAK Scheduling" in the Databook. - You can modify the effective timer limit through the TIMER_MOD_ACK_NAK field of the TIMER_CTRL_MAX_FUNC_NUM_OFF register. - After reset, the controller updates the default according to the Negotiated Link Width, Max_Payload_Size, and speed. The value is determined from Tables 3-7, 3-8, and 3-9 of the PCI Express Base Specification. - The limit must reflect the round trip latency from requester to completer. - If there is a change in the payload size or link width, the controller overrides any value that you have written to this register field, and resets the field back to the specification-defined value. The controller does not change the value in the TIMER_MOD_ACK_NAK field of the TIMER_CTRL_MAX_FUNC_NUM_OFF register.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_VENDOR_SPEC_DLLP_OFF_0

where <*j*> = 4, 8.

Description:This register holds the vendor specific DLLP.

PCIE_X4_EP_PF0_PORT_LOGIC_VENDOR_SPEC_DLLP_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_VENDOR_SPEC_DLLP_OFF_0

Offset: 0x704
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:0	0xffffffff	**VENDOR_SPEC_DLLP**: Vendor Specific DLLP Register. You can use this register to send a specific PCI Express DLLP. Your application can write 8-bit DLLP Type and 24-bit Payload data into this register, and set the VENDOR_SPECIFIC_DLLP_REQ field of the PORT_LINK_CTRL_OFF, to send the DLLP. - Bits[7:0]: DLLP Type - Bits[31:8]: Vendor Defined Payload (24 bits) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_PORT_FORCE_OFF_0

where <*j*> = 4, 8.

Description:This register can be used for testing and debuggong the link.

PCIE_X4_EP_PF0_PORT_LOGIC_PORT_FORCE_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_PORT_FORCE_OFF_0

Offset: 0x708
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00400000 (0b0000,0000,0100,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:24	RO	0x0	**RSVDP_24**: Reserved for future use.
23	RW	0x0	**DO_DESKEW_FOR_SRIS**: Use the transitions from TS2 to Logical Idle Symbol, SKP OS to Logical Idle Symbol, EIEOS to Logical Idle Symbol, and FTS Sequence to SKP OS to do deskew instead of using received SKP OS or TS1 to TS2 transition if DO_DESKEW_FOR_SRIS is set to '1'. Note: This register field is sticky.
22	RW	0x1	**SUPPORT_PART_LANES_RXEI_EXIT**: Support LTSSM transition from Polling.Active to Polling.Config based on Rx 8 TSs on any lanes which are Rx EI exit too from base spec after 24ms timeout. This prevents some lanes detected but not Rx EI exit and LTSSM cannot move to Polling.Config. You must set the parameter CX_AUTO_LANE_FLIP_CTRL_EN true for the auto lanes reversal. Note: This register field is sticky.
21:16	RW	0x0	**LINK_STATE**: Forced LTSSM State. The LTSSM state that the controller is forced to when you set the FORCE_EN bit (Force Link). LTSSM state encoding is defined by the lts_state variable in workspace/src/Layer1/smlh_ltssm.v. Note: This register field is sticky.
15	WO	0x0	**FORCE_EN**: Force Link. The controller supports a testing and debug capability to allow your software to force the LTSSM state machine into a specific state, and to force the controller to transmit a specific Link Command. Asserting this bit triggers the following actions: - Forces the LTSSM to the state specified by the Forced LTSSM State field. - Forces the controller to transmit the command specified by the Forced Link Command field. This is a self-clearing register field. Reading from this register field always returns a '0'.
14:12	RO	0x0	**RSVDP_12**: Reserved for future use.
11:8	RW	0x0	**FORCED_LTSSM**: Forced Link Command. The link command that the controller is forced to transmit when you set FORCE_EN bit (Force Link). Link command encoding is defined by the ltssm_cmd variable in workspace/src/Layer1/smlh_ltssm.v. Note: This register field is sticky.
7:0	RW	0x0	**LINK_NUM**: Link Number. Not used for endpoint. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_ACK_F_ASPM_CTRL_OFF_0

where <*j*> = 4, 8.

Description:This register is used to control ack frequency and L0-L1 ASPM behaviour.

PCIE_X4_EP_PF0_PORT_LOGIC_ACK_F_ASPM_CTRL_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_ACK_F_ASPM_CTRL_OFF_0

Offset: 0x70c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x23343400 (0b0010,0011,0011,0100,0011,0100,0000,0000)

Bit	R/W	Reset	Description
31	RO	0x0	**RSVDP_31**: Reserved for future use.
30	RW	0x0	**ENTER_ASPM**: ASPM L1 Entry Control. Note: This register field is sticky.
29:27	RW	0x4	**L1_ENTRANCE_LATENCY**: L1 Entrance Latency. Note: Programming this timer with a value greater that 32us has no effect unless extended sync is used, or all of the credits are infinite. Note: This register field is sticky.
26:24	RW	0x3	**L0S_ENTRANCE_LATENCY**: L0s Entrance Latency. Note: This register field is sticky.
23:16	RO	0x34	**COMMON_CLK_N_FTS**: Common Clock N_FTS. This is the N_FTS when common clock is used. The number of Fast Training Sequence ordered sets to be transmitted when transitioning from L0s to L0. This field is only writable (sticky) when all of the following configuration parameter equations are true: - CX_NFTS !=CX_COMM_NFTS - DEFAULT_L0S_EXIT_LATENCY ! =DEFAULT_COMM_L0S_EXIT_LATENCY - DEFAULT_L1_EXIT_LATENCY ! =DEFAULT_COMM_L1_EXIT_LATENCY Note: The access attributes of this field are as follows: - Wire: R - Dbi: R
15:8	RW	0x34	**ACK_N_FTS**: The number of Fast Training Sequence(N_FTS) ordered sets to be transmitted when transitioning from L0s to L0. Note: This register field is sticky.
7:0	RW	0x0	**ACK_FREQ**: Ack Frequency. The controller accumulates the number of pending ACKs specified here (up to 255) before scheduling an ACK DLLP. - 0: Indicates that this Ack Frequency Counter feature is turned off. The controller generates a low-priority ACK request for every TLP that it receives. The controller waits until the ACK Latency Timer expires, then converts the current low-priority ACK request to a high-priority ACK request and schedules the DLLP for transmission to the remote link partner. - 1-255: Indicates that the controller will schedule a high-priority ACK after receiving this number of TLPs. It might schedule the ACK before receiving this number of TLPs if the ACK Latency Timer expires, but never later. For a typical system, you do not have to modify the default setting. For more details, see "ACK/NAK Scheduling" in the Databook. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_PORT_LINK_CTRL_OFF_0

where <*j*> = 4, 8.

Description:Using this register you can control the port link behaviour.

Offset: 0x710
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X4_EP_PF0_PORT_LOGIC_PORT_LINK_CTRL_OFF_0

Reset: 0x00070120 (0b0000,0000,xx00,0111,0000,0001,0010,0000)

Bit	R/W	Reset	Description
31:28	RO	0x0	**RSVDP_28**: Reserved for future use.
27	RW	0x0	**TRANSMIT_LANE_REVERSALE_ENABLE**: TRANSMIT_LANE_REVERSALE_ENABLE is an internally reserved field. Do not use. Note: This register field is sticky.
26	RW	0x0	**EXTENDED_SYNCH**: EXTENDED_SYNCH is an internally reserved field. Do not use. Note: This register field is sticky.
25	RW	0x0	**CORRUPT_LCRC_ENABLE**: CORRUPT_LCRC_ENABLE is an internally reserved field. Do not use. Note: This register field is sticky.
24	RW	0x0	**BEACON_ENABLE**: BEACON_ENABLE is an internally reserved field. Do not use. Note: This register field is sticky.
21:16	RW	0x7	**LINK_CAPABLE**: Link Mode Enable. Sets the number of lanes in the link that you want to connect to the link partner. When you have unused lanes in your system, then you must change the value in this register to reflect the number of lanes. You must also change the value in the "Predetermined Number of Lanes" field of the "Link Width and Speed Change Control Register". For more information, see "How to Tie Off Unused Lanes". For information on upsizing and downsizing the link width, see "Link Establishment". Note: This register field is sticky.
15:12	RO	0x0	**RSVDP_12**: Reserved for future use.
11:8	RW	0x1	**LINK_RATE**: LINK_RATE is an internally reserved field. Do not use. Note: This register field is sticky.

Bit	R/W	Reset	Description
7	RW	0x0	**FAST_LINK_MODE**: Fast Link Mode. Sets all internal LTSSM millisecond timers to Fast Mode for speeding up simulation. Forces the LTSSM training (link initialization) to use shorter time-outs and to link up faster. - The default scaling factor can be changed using the DEFAULT_FAST_LINK_SCALING_FACTOR parameter or through the FAST_LINK_SCALING_FACTOR field in the TIMER_CTRL_MAX_FUNC_NUM_OFF register. - Fast Link Mode can also be activated by setting the diag_ctrl_bus[2] pin to '1'. For more details, see the "Fast Link Simulation Mode" section in the "Integrating the Controller with the PHY or Application RTL or Verification IP" chapter of the User Guide. Note: This register field is sticky.
6	RW	0x0	**LINK_DISABLE**: LINK_DISABLE is an internally reserved field. Do not use. Note: This register field is sticky.
5	RW	0x1	**DLL_LINK_EN**: DLL Link Enable. Note: This register field is sticky.
4	RO	0x0	**RSVDP_4**: Reserved for future use.
3	RW	0x0	**RESET_ASSERT**: Reset Assert. Triggers a recovery and forces the LTSSM to the hot reset state (downstream port only). Note: This register field is sticky.
2	RW	0x0	**LOOPBACK_ENABLE**: Loopback Enable. Turns on loopback. For more details, see "Loopback" in the Databook. Note: This register field is sticky.
1	RW	0x0	**SCRAMBLE_DISABLE**: Scramble Disable. Turns off data scrambling. Note: This register field is sticky.
0	RW	0x0	**VENDOR_SPECIFIC_DLLP_REQ**: Vendor Specific DLLP Request. Reading from this self-clearing register field always returns a '0'.

PCIE_X8_EP_PF0_PORT_LOGIC_PORT_LINK_CTRL_OFF_0

Reset: 0x000f0120 (0b0000,0000,xx00,1111,0000,0001,0010,0000)

Bit	R/W	Reset	Description
31:28	RO	0x0	**RSVDP_28**: Reserved for future use.
27	RW	0x0	**TRANSMIT_LANE_REVERSALE_ENABLE**: TRANSMIT_LANE_REVERSALE_ENABLE is an internally reserved field. Do not use. Note: This register field is sticky.
26	RW	0x0	**EXTENDED_SYNCH**: EXTENDED_SYNCH is an internally reserved field. Do not use. Note: This register field is sticky.
25	RW	0x0	**CORRUPT_LCRC_ENABLE**: CORRUPT_LCRC_ENABLE is an internally reserved field. Do not use. Note: This register field is sticky.

Bit	R/W	Reset	Description
24	RW	0x0	**BEACON_ENABLE**: BEACON_ENABLE is an internally reserved field. Do not use. Note: This register field is sticky.
21:16	RW	0xf	**LINK_CAPABLE**: Link Mode Enable. Sets the number of lanes in the link that you want to connect to the link partner. When you have unused lanes in your system, then you must change the value in this register to reflect the number of lanes. You must also change the value in the "Predetermined Number of Lanes" field of the "Link Width and Speed Change Control Register". For more information, see "How to Tie Off Unused Lanes". For information on upsizing and downsizing the link width, see "Link Establishment". Note: This register field is sticky.
15:12	RO	0x0	**RSVDP_12**: Reserved for future use.
11:8	RW	0x1	**LINK_RATE**: LINK_RATE is an internally reserved field. Do not use. Note: This register field is sticky.
7	RW	0x0	**FAST_LINK_MODE**: Fast Link Mode. Sets all internal LTSSM millisecond timers to Fast Mode for speeding up simulation. Forces the LTSSM training (link initialization) to use shorter time-outs and to link up faster. - The default scaling factor can be changed using the DEFAULT_FAST_LINK_SCALING_FACTOR parameter or through the FAST_LINK_SCALING_FACTOR field in the TIMER_CTRL_MAX_FUNC_NUM_OFF register. - Fast Link Mode can also be activated by setting the diag_ctrl_bus[2] pin to '1'. For more details, see the "Fast Link Simulation Mode" section in the "Integrating the Controller with the PHY or Application RTL or Verification IP" chapter of the User Guide. Note: This register field is sticky.
6	RW	0x0	**LINK_DISABLE**: LINK_DISABLE is an internally reserved field. Do not use. Note: This register field is sticky.
5	RW	0x1	**DLL_LINK_EN**: DLL Link Enable. Note: This register field is sticky.
4	RO	0x0	**RSVDP_4**: Reserved for future use.
3	RW	0x0	**RESET_ASSERT**: Reset Assert. Triggers a recovery and forces the LTSSM to the hot reset state (downstream port only). Note: This register field is sticky.
2	RW	0x0	**LOOPBACK_ENABLE**: Loopback Enable. Turns on loopback. For more details, see "Loopback" in the Databook. Note: This register field is sticky.
1	RW	0x0	**SCRAMBLE_DISABLE**: Scramble Disable. Turns off data scrambling. Note: This register field is sticky.
0	RW	0x0	**VENDOR_SPECIFIC_DLLP_REQ**: Vendor Specific DLLP Request. Reading from this self-clearing register field always returns a '0'.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_LANE_SKEW_OFF_0

where <*j*> = 4, 8.

Description:This register is used to control the lane skew behaviour.

Offset: 0x714
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X4_EP_PF0_PORT_LOGIC_LANE_SKEW_OFF_0

Reset: 0x18000000 (0b0001,1000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**DISABLE_LANE_TO_LANE_DESKEW**: Disable Lane-to-Lane Deskew. Causes the controller to disable the internal Lane-to-Lane deskew logic. Note: This register field is sticky.
30:27	0x3	**IMPLEMENT_NUM_LANES**: Implementation-specific Number of Lanes. Set the implementation-specific number of lanes. The number of lanes to be used when in Loopback Master. The number of lanes programmed must be equal to or less than the valid number of lanes set in LINK_CAPABLE field. You must configure this field before initiating Loopback by writing in the LOOPBACK_ENABLE field. The controller will transition from Loopback.Entry to Loopback.Active after receiving two consecutive TS1 Ordered Sets with the Loopback bit asserted on the implementation specific number of lanes configured in this field. Note: This register field is sticky.
26	0x0	**ELASTIC_BUFFER_MODE**: Selects Elasticity Buffer operating mode: Note: This register field is sticky.
25	0x0	**ACK_NAK_DISABLE**: Ack/Nak Disable. Prevents the controller from sending ACK and NAK DLLPs. Note: This register field is sticky.
24	0x0	**FLOW_CTRL_DISABLE**: Flow Control Disable. Prevents the controller from sending FC DLLPs. Note: This register field is sticky.
23:0	0x0	**INSERT_LANE_SKEW**: INSERT_LANE_SKEW is an internally reserved field. Do not use. Note: This register field is sticky.

PCIE_X8_EP_PF0_PORT_LOGIC_LANE_SKEW_OFF_0

Reset: 0x38000000 (0b0011,1000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**DISABLE_LANE_TO_LANE_DESKEW**: Disable Lane-to-Lane Deskew. Causes the controller to disable the internal Lane-to-Lane deskew logic. Note: This register field is sticky.
30:27	0x7	**IMPLEMENT_NUM_LANES**: Implementation-specific Number of Lanes. Set the implementation-specific number of lanes. The number of lanes to be used when in Loopback Master. The number of lanes programmed must be equal to or less than the valid number of lanes set in LINK_CAPABLE field. You must configure this field before initiating Loopback by writing in the LOOPBACK_ENABLE field. The controller will transition from Loopback.Entry to Loopback.Active after receiving two consecutive TS1 Ordered Sets with the Loopback bit asserted on the implementation specific number of lanes configured in this field. Note: This register field is sticky.
26	0x0	**ELASTIC_BUFFER_MODE**: Selects Elasticity Buffer operating mode: Note: This register field is sticky.
25	0x0	**ACK_NAK_DISABLE**: Ack/Nak Disable. Prevents the controller from sending ACK and NAK DLLPs. Note: This register field is sticky.
24	0x0	**FLOW_CTRL_DISABLE**: Flow Control Disable. Prevents the controller from sending FC DLLPs. Note: This register field is sticky.
23:0	0x0	**INSERT_LANE_SKEW**: INSERT_LANE_SKEW is an internally reserved field. Do not use. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_TIMER_CTRL_MAX_FUNC_NUM_OFF_0

where <*j*> = 4, 8.

Description:This register holds the ack frequency, latency, replay, fast link scaling timers, and max function number values.

PCIE_X4_EP_PF0_PORT_LOGIC_TIMER_CTRL_MAX_FUNC_NUM_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_TIMER_CTRL_MAX_FUNC_NUM_OFF_0

Offset: 0x718
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00010000 (0b0000,0000,0000,0001,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RO	0x0	**RSVDP_31**: Reserved for future use.

Bit	R/W	Reset	Description
30:29	RW	0x0	**FAST_LINK_SCALING_FACTOR**: Fast Link Timer Scaling Factor. Sets the scaling factor of LTSSM timer when FAST_LINK_MODE field in PORT_LINK_CTRL_OFF is set to '1'. Default is set by the hidden configuration parameter DEFAULT_FAST_LINK_SCALING_FACTOR which defaults to '0'. Note: This register field is sticky.
28:24	RW	0x0	**UPDATE_FREQ_TIMER**: UPDATE_FREQ_TIMER is an internally reserved field. Do not use. Note: This register field is sticky.
23:19	RW	0x0	**TIMER_MOD_ACK_NAK**: Ack Latency Timer Modifier. Increases the timer value for the Ack latency timer in increments of 64 clock cycles. A value of '0' represents no modification to the timer value. For more details, see the ROUND_TRIP_LATENCY_TIME_LIMIT field of the ACK_LATENCY_TIMER_OFF register. Note: This register field is sticky.
18:14	RW	0x4	**TIMER_MOD_REPLAY_TIMER**: Replay Timer Limit Modifier. Increases the time-out value for the replay timer in increments of 64 clock cycles at Gen1 or Gen2 speed, and in increments of 256 clock cycles at Gen3 speed. A value of '0' represents no modification to the timer limit. For more details, see the REPLAY_TIME_LIMIT field of the ACK_LATENCY_TIMER_OFF register. At Gen3 speed, the controller automatically changes the value of this field to DEFAULT_GEN3_REPLAY_ADJ. Note: This register field is sticky.
13:8	RO	0x0	**RSVDP_8**: Reserved for future use.
7:0	RW	0x0	**MAX_FUNC_NUM**: Maximum function number that can be used in a request. Configuration requests targeted at function numbers above this value are returned with UR (unsupported request). Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_SYMBOL_TIMER_FILTER_1_OFF_0

where <*j*> = 4, 8.

Description:The Filter Mask 1 Register modifies the RADM filtering and error handling rules. For more details, see the "Receive Filtering" section. In each case, '0' applies the associated filtering rule and '1' masks the associated filtering rule.

PCIE_X4_EP_PF0_PORT_LOGIC_SYMBOL_TIMER_FILTER_1_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_SYMBOL_TIMER_FILTER_1_OFF_0

Offset: 0x71c
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000140 (0b0000,0000,0000,0000,0000,0001,0100,0000)

Bit	Reset	Description
31:16	0x0	**MASK_RADM_1**: Filter Mask 1. The Filter Mask 1 Register modifies the RADM filtering and error handling rules. For more details, see the "Receive Filtering" section. In each case, '0' applies the associated filtering rule and '1' masks the associated filtering rule. [31]: CX_FLT_MASK_RC_CFG_DISCARD - 0: For RADM RC filter to not allow CFG transaction being received - 1: For RADM RC filter to allow CFG transaction being received [30]: CX_FLT_MASK_RC_IO_DISCARD - 0: For RADM RC filter to not allow IO transaction being received - 1: For RADM RC filter to allow IO transaction being received [29]: CX_FLT_MASK_MSG_DROP - 0: Drop MSG TLP (except for Vendor MSG). Send decoded message on the SII. - 1: Do not Drop MSG (except for Vendor MSG). Send message TLPs to your application on TRGT1 and send decoded message on the SII. - The default for this bit is the inverse of FLT_DROP_MSG. That is, if FLT_DROP_MSG =1, then the default of this bit is '0' (drop message TLPs). This bit only controls message TLPs other than Vendor MSGs. Vendor MSGs are controlled by Filter Mask Register 2, bits [1:0]. The controller never passes ATS Invalidate messages to the SII interface regardless of this filter rule setting. The controller passes all ATS Invalidate messages to TRGT1 (or AXI bridge master), as they are too big for the SII. [28]: CX_FLT_MASK_CPL_ECRC_DISCARD - Only used when completion queue is advertised with infinite credits and is in store-and-forward mode. - 0: Discard completions with ECRC errors - 1: Allow completions with ECRC errors to be passed up - Reserved field for SW. [27]: CX_FLT_MASK_ECRC_DISCARD - 0: Discard TLPs with ECRC errors - 1: Allow TLPs with ECRC errors to be passed up [26]: CX_FLT_MASK_CPL_LEN_MATCH - 0: Enforce length match for completions; a violation results in cpl_abort, and possibly AER of unexp_cpl_err - 1: Mask length match for completions [25]: CX_FLT_MASK_CPL_ATTR_MATCH - 0: Enforce attribute match for completions; a violation results in a malformed TLP error, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca - 1: Mask attribute match for completions [24]: CX_FLT_MASK_CPL_TC_MATCH - 0: Enforce Traffic Class match for completions; a violation results in a malformed TLP error, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca - 1: Mask Traffic Class match for completions [23]: CX_FLT_MASK_CPL_FUNC_MATCH - 0: Enforce function match for completions; a violation results in cpl_abort, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca - 1: Mask function match for completions [22]: CX_FLT_MASK_CPL_REQID_MATCH - 0: Enforce Req. Id match for completions; a violation result in cpl_abort, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca - 1: Mask Req. Id match for completions [21]: CX_FLT_MASK_CPL_TAGERR_MATCH - 0: Enforce Tag Error Rules for completions; a violation result in cpl_abort, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca - 1: Mask Tag Error Rules for completions [20]: CX_FLT_MASK_LOCKED_RD_AS_UR - 0: Treat locked Read TLPs as UR for EP; Supported for RC - 1: Treat locked Read TLPs as Supported for EP; UR for RC [19]: CX_FLT_MASK_CFG_TYPE1_REQ_AS_UR - 0: Treat CFG type1 TLPs as UR for EP; Supported for RC - 1: Treat CFG type1 TLPs as Supported for EP; UR for RC - When CX_SRIOV_ENABLE is set then this bit is set to allow the filter to process Type 1 Config requests if the EP consumes more than one bus number. [18]: CX_FLT_MASK_UR_OUTSIDE_BAR - 0: Treat out-of-bar TLPs as UR - 1: Do not treat out-of-bar TLPs as UR [17]: CX_FLT_MASK_UR_POIS - 0: Treat poisoned request TLPs as UR - 1: Do not treat poisoned request TLPs as UR - The native controller always passes poisoned completions to your application except when you are using the DMA read channel. [16]: CX_FLT_MASK_UR_FUNC_MISMATCH - 0: Treat Function MisMatched TLPs as UR - 1: Do not treat Function MisMatched TLPs as UR Note: This register field is sticky.
15	0x0	**DISABLE_FC_WD_TIMER**: Disable FC Watchdog Timer. Note: This register field is sticky.
14:11	0x0	**EIDLE_TIMER**: EIDLE_TIMER is an internally reserved field. Do not use. Note: This register field is sticky.

Bit	Reset	Description
10:0	0x140	**SKP_INT_VAL**: SKP Interval Value. The number of symbol times to wait between transmitting SKP ordered sets. The controller waits the number of symbol times in this register plus 1, between transmitting SKP ordered sets. Your application must program this register accordingly. For example, if 1536 were programmed into this register (in a 250 MHz controller), then the controller actually transmits SKP ordered sets once every 1537 symbol times. The value programmed to this register is actually clock ticks and not symbol times. In a 125 MHz controller, programming the value programmed to this register should be scaled down by a factor of 2 (because one clock tick =two symbol times in this case). Note: This value is not used at Gen3 speed; the skip interval is hardcoded to 370 blocks. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_FILTER_MASK_2_OFF_0

where <*j*> = 4, 8.

Description:This register modifies the RADM filtering and error handling rules. For more details, see the "Receive Filtering" section. In each case, '0' applies the associated filtering rule and '1' masks the associated filtering rule.

PCIE_X4_EP_PF0_PORT_LOGIC_FILTER_MASK_2_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_FILTER_MASK_2_OFF_0

Offset: 0x720
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MASK_RADM_2**: Filter Mask 2. This field modifies the RADM filtering and error handling rules. For more details, see the "Receive Filtering" in the Databook. In each case, '0' applies the associated filtering rule and '1' masks the associated filtering rule. [31:10]: Reserved [9]: CX_FLT_MASK_CPL_IN_LUT_CHECK - 0: Disable masking of checking if the tag of CPL is registered in LUT - 1: Enable masking of checking if the tag of CPL is registered in LUT [8]: CX_FLT_MASK_POIS_ERROR_REPORTING - 0: Disable masking of error reporting for Poisoned TLPs - 1: Enable masking of error reporting for Poisoned TLPs [7]: CX_FLT_MASK_PRS_DROP - 0: Allow PRS message to pass through - 1: Drop PRS Messages silently - This bit is ignored when the CX_FLT_MASK_MSG_DROP bit in the MASK_RADM_1 field of the SYMBOL_TIMER_FILTER_1_OFF register is set to '1'. [6]: CX_FLT_UNMASK_TD - 0: Disable unmask TD bit if CX_STRIP_ECRC_ENABLE - 1: Enable unmask TD bit if CX_STRIP_ECRC_ENABLE [5]: CX_FLT_UNMASK_UR_POIS_TRGT0 - 0: Disable unmask CX_FLT_MASK_UR_POIS with TRGT0 destination - 1: Enable unmask CX_FLT_MASK_UR_POIS with TRGT0 destination [4]: CX_FLT_MASK_LN_VENMSG1_DROP - 0: Allow LN message to pass through - 1: Drop LN Messages silently [3]: CX_FLT_MASK_HANDLE_FLUSH - 0: Disable controller Filter to handle flush request - 1: Enable controller Filter to handle flush request [2]: CX_FLT_MASK_DABORT_4UCPL - 0: Enable DLLP abort for unexpected completion - 1: Do not enable DLLP abort for unexpected completion [1]: CX_FLT_MASK_VENMSG1_DROP - 0: Vendor MSG Type 1 dropped silently - 1: Vendor MSG Type 1 not dropped [0]: CX_FLT_MASK_VENMSG0_DROP - 0: Vendor MSG Type 0 dropped with UR error reporting - 1: Vendor MSG Type 0 not dropped Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_AMBA_MUL_OB_DECOMP_NP_SUB_REQ_CTRL_OFF_0

where <*j*> = 4, 8.

Description:This register controls multiple outbound decomposed NP subRequests operation.

PCIE_X4_EP_PF0_PORT_LOGIC_AMBA_MUL_OB_DECOMP_NP_SUB_REQ_CTRL_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_AMBA_MUL_OB_DECOMP_NP_SUB_REQ_CTRL_OFF_0

Offset: 0x724
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000001 (0b0000,0000,0000,0000,0000,0000,0000,0001)

Bit	R/W	Reset	Description
31:1	RO	0x0	**RSVDP_1**: Reserved for future use.

Bit	R/W	Reset	Description
0	RW	0x1	**OB_RD_SPLIT_BURST_EN**: Enable AMBA Multiple Outbound Decomposed NP SubRequests. You should not clear this register unless your application master is requesting an amount of read data greater than Max_Read_Request_Size, and the remote device (or switch) is reordering completions that have different tags. For more details, see "AXI Bridge Ordering" in the AXI chapter of the Databook. Note: The access attributes of this field are as follows: - Wire: R/W (sticky) - Dbi: R/W (sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_PL_DEBUG0_OFF_0

where <*j*> = 4, 8.

Description:This register holds cxpl_debug_info[31:0].

PCIE_X4_EP_PF0_PORT_LOGIC_PL_DEBUG0_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_PL_DEBUG0_OFF_0

Offset: 0x728
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DEB_REG_0**: The value on cxpl_debug_info[31:0].

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_PL_DEBUG1_OFF_0

where <*j*> = 4, 8.

Description:This register holds cxpl_debug_info[63:32].

PCIE_X4_EP_PF0_PORT_LOGIC_PL_DEBUG1_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_PL_DEBUG1_OFF_0

Offset: 0x72c
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DEB_REG_1**: The value on cxpl_debug_info[63:32].

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_TX_P_FC_CREDIT_STATUS_OFF_0

where <*j*> = 4, 8.

Description:This register provides transmit posted FC credit status.

PCIE_X4_EP_PF0_PORT_LOGIC_TX_P_FC_CREDIT_STATUS_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_TX_P_FC_CREDIT_STATUS_OFF_0

Offset: 0x730
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**RSVDP_TX_P_FC_CREDIT_STATUS**: Reserved for future use.
27:16	0x0	**TX_P_HEADER_FC_CREDIT**: Transmit Posted Header FC Credits. - The posted Header credits advertised by the receiver at the other end of the link, updated with each UpdateFC DLLP. - Default value depends on the number of advertised credits for header and data - Scaled Flow Control: [4'b0, xtlh_xadm_ph_cdts, xtlh_xadm_pd_cdts]; When the number of advertised posted credits (both header and data) are infinite, then the default would be [4'b0, 12'hFFF, 16'hFFFF]. - No Scaling: [12'b0, xtlh_xadm_ph_cdts, xtlh_xadm_pd_cdts]; When the number of advertised posted credits (both header and data) are infinite, then the default would be [12'b0, 8'hFF, 12'hFFF].
15:0	0x0	**TX_P_DATA_FC_CREDIT**: Transmit Posted Data FC Credits. - The posted Data credits advertised by the receiver at the other end of the link, updated with each UpdateFC DLLP. - Default value depends on the number of advertised credits for header and data - Scaled Flow Control: [4'b0, xtlh_xadm_ph_cdts, xtlh_xadm_pd_cdts]; When the number of advertised posted credits (both header and data) are infinite, then the default would be [4'b0, 12'hFFF, 16'hFFFF]. - No Scaling: [12'b0, xtlh_xadm_ph_cdts, xtlh_xadm_pd_cdts]; When the number of advertised posted credits (both header and data) are infinite, then the default would be [12'b0, 8'hFF, 12'hFFF].

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_TX_NP_FC_CREDIT_STATUS_OFF_0

where <*j*> = 4, 8.

Description:This register provides the transmit Non-Posted FC credit status.

PCIE_X4_EP_PF0_PORT_LOGIC_TX_NP_FC_CREDIT_STATUS_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_TX_NP_FC_CREDIT_STATUS_OFF_0

Offset: 0x734
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**RSVDP_TX_NP_FC_CREDIT_STATUS**: Reserved for future use.
27:16	0x0	**TX_NP_HEADER_FC_CREDIT**: Transmit Non-Posted Header FC Credits. - The non-posted Header credits advertised by the receiver at the other end of the link, updated with each UpdateFC DLLP. - Default value depends on the number of advertised credits for header and data - Scaled Flow Control: [4'b0, xtlh_xadm_nph_cdts, xtlh_xadm_npd_cdts]; When the number of advertised non-posted credits (both header and data) are infinite, then the default would be [4'b0, 12'hFFF, 16'hFFFF]. - No Scaling: [12'b0, xtlh_xadm_nph_cdts, xtlh_xadm_npd_cdts]; When the number of advertised non-posted credits (both header and data) are infinite, then the default would be [12'b0, 8'hFF, 12'hFFF].
15:0	0x0	**TX_NP_DATA_FC_CREDIT**: Transmit Non-Posted Data FC Credits. - The non-posted Data credits advertised by the receiver at the other end of the link, updated with each UpdateFC DLLP. - Default value depends on the number of advertised credits for header and data - Scaled Flow Control: [4'b0, xtlh_xadm_nph_cdts, xtlh_xadm_npd_cdts]; When the number of advertised non-posted credits (both header and data) are infinite, then the default would be [4'b0, 12'hFFF, 16'hFFFF]. - No Scaling: [12'b0, xtlh_xadm_nph_cdts, xtlh_xadm_npd_cdts]; When the number of advertised non-posted credits (both header and data) are infinite, then the default would be [12'b0, 8'hFF, 12'hFFF].

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_TX_CPL_FC_CREDIT_STATUS_OFF_0

where <*j*> = 4, 8.

Description:This register provides transmit completion FC credit status.

PCIE_X4_EP_PF0_PORT_LOGIC_TX_CPL_FC_CREDIT_STATUS_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_TX_CPL_FC_CREDIT_STATUS_OFF_0

Offset: 0x738
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**RSVDP_TX_CPL_FC_CREDIT_STATUS**: Reserved for future use.
27:16	0x0	**TX_CPL_HEADER_FC_CREDIT**: Transmit Completion Header FC Credits. - The Completion Header credits advertised by the receiver at the other end of the link, updated with each UpdateFC DLLP. - Default value depends on the number of advertised credits for header and data - Scaled Flow Control: [4'b0, xtlh_xadm_cplh_cdts, xtlh_xadm_cpld_cdts]; When the number of advertised completion credits (both header and data) are infinite, then the default would be [4'b0, 12'hFFF, 16'hFFFF]. - No Scaling: [12'b0, xtlh_xadm_cplh_cdts, xtlh_xadm_cpld_cdts]; When the number of advertised completion credits (both header and data) are infinite, then the default would be [12'b0, 8'hFF, 12'hFFF].
15:0	0x0	**TX_CPL_DATA_FC_CREDIT**: Transmit Completion Data FC Credits. - The Completion Data credits advertised by the receiver at the other end of the link, updated with each UpdateFC DLLP. - Default value depends on the number of advertised credits for header and data - Scaled Flow Control: [4'b0, xtlh_xadm_cplh_cdts, xtlh_xadm_cpld_cdts]; When the number of advertised completion credits (both header and data) are infinite, then the default would be [4'b0, 12'hFFF, 16'hFFFF]. - No Scaling: [12'b0, xtlh_xadm_cplh_cdts, xtlh_xadm_cpld_cdts]; When the number of advertised completion credits (both header and data) are infinite, then the default would be [12'b0, 8'hFF, 12'hFFF].

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_QUEUE_STATUS_OFF_0

where <*j*> = 4, 8.

Description:This register provides the queue status.

PCIE_X4_EP_PF0_PORT_LOGIC_QUEUE_STATUS_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_QUEUE_STATUS_OFF_0

Offset: 0x73c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,xx00,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**TIMER_MOD_FLOW_CONTROL_EN**: FC Latency Timer Override Enable. Note: This register field is sticky.
30:29	RO	0x0	**RSVDP_29**: Reserved for future use.
28:16	RW	0x0	**TIMER_MOD_FLOW_CONTROL**: FC Latency Timer Override Value. When you set the "FC Latency Timer Override Enable" in this register, the value in this field will override the FC latency timer value that the controller calculates according to the PCIe specification. For more details, see "Flow Control" in the Databook. Note: This register field is sticky.
13	RO	0x0	**RX_SERIALIZATION_Q_NON_EMPTY**: Receive Serialization Queue Not Empty.
12:4	RO	0x0	**RSVDP_4**: Reserved for future use.
3	RW	0x0	**RX_QUEUE_OVERFLOW**: Receive Credit Queue Overflow.
2	RO	0x0	**RX_QUEUE_NON_EMPTY**: Receive Credit Queue Not Empty.
1	RO	0x0	**TX_RETRY_BUFFER_NE**: Transmit Retry Buffer Not Empty.
0	RO	0x0	**RX_TLP_FC_CREDIT_NON_RETURN**: Received TLP FC Credits Not Returned.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_VC_TX_ARBI_1_OFF_0

where <*j*> = 4, 8.

Description:This register is used for setting the WRR weights for VC0 - VC3.

PCIE_X4_EP_PF0_PORT_LOGIC_VC_TX_ARBI_1_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_VC_TX_ARBI_1_OFF_0

Offset: 0x740
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000f (0b0000,0000,0000,0000,0000,0000,0000,1111)

Bit	Reset	Description
31:24	0x0	**WRR_WEIGHT_VC_3**: WRR Weight for VC3. Note: The access attributes of this field are as follows: - Wire: R - Dbi: R
23:16	0x0	**WRR_WEIGHT_VC_2**: WRR Weight for VC2. Note: The access attributes of this field are as follows: - Wire: R - Dbi: R
15:8	0x0	**WRR_WEIGHT_VC_1**: WRR Weight for VC1. Note: The access attributes of this field are as follows: - Wire: R - Dbi: R
7:0	0xf	**WRR_WEIGHT_VC_0**: WRR Weight for VC0. Note: The access attributes of this field are as follows: - Wire: R - Dbi: R

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_VC_TX_ARBI_2_OFF_0

where <*j*> = 4, 8.

Description:This register is used for setting the WRR weights for VC4 - VC7.

PCIE_X4_EP_PF0_PORT_LOGIC_VC_TX_ARBI_2_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_VC_TX_ARBI_2_OFF_0

Offset: 0x744
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**WRR_WEIGHT_VC_7**: WRR Weight for VC7. Note: The access attributes of this field are as follows: - Wire: R - Dbi: R
23:16	0x0	**WRR_WEIGHT_VC_6**: WRR Weight for VC6. Note: The access attributes of this field are as follows: - Wire: R - Dbi: R
15:8	0x0	**WRR_WEIGHT_VC_5**: WRR Weight for VC5. Note: The access attributes of this field are as follows: - Wire: R - Dbi: R
7:0	0x0	**WRR_WEIGHT_VC_4**: WRR Weight for VC4. Note: The access attributes of this field are as follows: - Wire: R - Dbi: R

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_VC0_P_RX_Q_CTRL_OFF_0

where <*j*> = 4, 8.

Description:This register controls segmented-buffer VC0 posted receive queue operation.

PCIE_X4_EP_PF0_PORT_LOGIC_VC0_P_RX_Q_CTRL_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_VC0_P_RX_Q_CTRL_OFF_0

Offset: 0x748
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x4622c140 (0b0100,0110,0010,0010,1100,0001,0100,0000)

Bit	Reset	Description
31	0x0	**VC_ORDERING_RX_Q**: VC Ordering for Receive Queues. Determines the VC ordering rule for the receive queues, used only in the segmented-buffer configuration: Note: This register field is sticky.
30	0x1	**TLP_TYPE_ORDERING_VC0**: TLP Type Ordering for VC0. Determines the TLP type ordering rule for VC0 receive queues, used only in the segmented-buffer configuration: Note: This register field is sticky.
29:28	0x0	**RESERVED5**: Reserved. Note: This register field is sticky.
27:26	0x1	**VC0_P_DATA_SCALE**: VC0 Scale Posted Data Credits. Note: This register field is sticky.
25:24	0x2	**VC0_P_HDR_SCALE**: VC0 Scale Posted Header Credits. Note: This register field is sticky.
23:21	0x1	**VC0_P_TLP_Q_MODE**: Reserved. Note: This register field is sticky.
20	0x0	**RESERVED4**: Reserved. Note: This register field is sticky.
19:12	0x2c	**VC0_P_HEADER_CREDIT**: VC0 Posted Header Credits. The number of initial posted header credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: R (sticky) Note: This register field is sticky.
11:0	0x140	**VC0_P_DATA_CREDIT**: VC0 Posted Data Credits. The number of initial posted data credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: R (sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_VC0_NP_RX_Q_CTRL_OFF_0

where <*j*> = 4, 8.

Description:This register controls the segmented-buffer VC0 non-posted receive queue operation.

PCIE_X4_EP_PF0_PORT_LOGIC_VC0_NP_RX_Q_CTRL_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_VC0_NP_RX_Q_CTRL_OFF_0

Offset: 0x74c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0622c02c (0b0000,0110,0010,0010,1100,0000,0010,1100)

Bit	Reset	Description
31:28	0x0	**RESERVED7**: Reserved. Note: This register field is sticky.
27:26	0x1	**VC0_NP_DATA_SCALE**: VC0 Scale Non-Posted Data Credits. Note: This register field is sticky.
25:24	0x2	**VC0_NP_HDR_SCALE**: VC0 Scale Non-Posted Header Credits. Note: This register field is sticky.
23:21	0x1	**VC0_NP_TLP_Q_MODE**: Reserved. Note: This register field is sticky.
20	0x0	**RESERVED6**: Reserved. Note: This register field is sticky.
19:12	0x2c	**VC0_NP_HEADER_CREDIT**: VC0 Non-Posted Header Credits. The number of initial non-posted header credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: R (sticky) Note: This register field is sticky.
11:0	0x2c	**VC0_NP_DATA_CREDIT**: VC0 Non-Posted Data Credits. The number of initial non-posted data credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: R (sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_VC0_CPL_RX_Q_CTRL_OFF_0

where <*j*> = 4, 8.

Description:This register controls the segmented-buffer VC0 completion receive queue operation.

PCIE_X4_EP_PF0_PORT_LOGIC_VC0_CPL_RX_Q_CTRL_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_VC0_CPL_RX_Q_CTRL_OFF_0

Offset: 0x750
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x06200000 (0b0000,0110,0010,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**RESERVED9**: Reserved. Note: This register field is sticky.
27:26	0x1	**VC0_CPL_DATA_SCALE**: VC0 Scale CPL Data Credits. Note: This register field is sticky.
25:24	0x2	**VC0_CPL_HDR_SCALE**: VC0 Scale CPL Header Credits. Note: This register field is sticky.
23:21	0x1	**VC0_CPL_TLP_Q_MODE**: Reserved. Note: This register field is sticky.
20	0x0	**RESERVED8**: Reserved. Note: This register field is sticky.
19:12	0x0	**VC0_CPL_HEADER_CREDIT**: VC0 Completion Header Credits. The number of initial Completion header credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: R (sticky) Note: This register field is sticky.
11:0	0x0	**VC0_CPL_DATA_CREDIT**: VC0 Completion Data Credits. The number of initial Completion data credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: R (sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_GEN2_CTRL_OFF_0

where <*j*> = 4, 8.

Description:This register controls various functions of the controller related to link training, lane reversal, and equalization.

Offset: 0x80c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0

PCIE_X4_EP_PF0_PORT_LOGIC_GEN2_CTRL_OFF_0

Reset: 0x00030434 (0b00xx,0000,0000,0011,0000,0100,0011,0100)

Bit	R/W	Reset	Description
31	RO	0x0	**RSVDP_31**: Reserved for future use.
30	RW	0x0	**FORCE_LANE_FLIP**: Enable to force the LANE_UNDER_TEST physical lane flips to logical lane 0. All the other physical lanes are turned off. The LINK_CAPABLE register must be set to 1 and only x1 link can be formed if the FORCE_LANE_FLIP register is set to 1. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: R/W (sticky) Note: This register field is sticky.
27:24	RW	0x0	**LANE_UNDER_TEST**: The Lane Under Test is the lane for Forced Lane Flip or for Loopback Eq. Only one lane is configured each time. The default of this field is the CX_DEFAULT_LANE_UNDER_TEST configuration parameter. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: R/W (sticky) Note: This register field is sticky.
23	RW	0x0	**SELECTABLE_DEEMPH_BIT_MUX**: The selectable deemphasis bit (Symbol 4 bit 6) of the transmitted TS2 Ordered Sets for DSP in Recovery.RcvrCfg state is muxed between the Selectable De-emphasis field in the Link Control 2 register and the value requested by USP in Recovery.RcvrLock state through Tx TS1s from USP. Note: This register field is sticky.
22	RW	0x0	**SELECT_DEEMPH_VAR_MUX**: The select_deemphasis variable for DSP on entry to Recovery.RcvrCfg state is muxed between the Selectable De-emphasis field in the Link Control 2 register and the value requested by the Upstream Port in the eight consecutive TS1 Ordered Sets it received. Note: This register field is sticky.
21	RW	0x0	**GEN1_EI_INFERENCE**: Electrical Idle Inference Mode at Gen1 Rate. Programmable mode to determine inferred electrical idle (EI) in Recovery.Speed or Loopback.Active (as slave) state at Gen1 speed by looking for a '1' value on RxElecIdle instead of looking for a '0' on RxValid. If the PHY fails to deassert the RxValid signal in Recovery.Speed or Loopback.Active (because of corrupted EIOS for example), then EI cannot be inferred successfully in the controller by just detecting the condition RxValid=0. Note: This register field is sticky.
20	RW	0x0	**SEL_DEEMPHASIS**: Used to set the de-emphasis level for upstream ports. This bit selects the level of de-emphasis the link operates at. Note: The access attributes of this field are as follows: - Wire: R/W (sticky) - Dbi: R/W (sticky) Note: This register field is sticky.
19	RW	0x0	**CONFIG_TX_COMP_RX**: Config Tx Compliance Receive Bit. Note: The access attributes of this field are as follows: - Wire: R/W (sticky) - Dbi: R/W (sticky) Note: This register field is sticky.
18	RW	0x0	**CONFIG_PHY_TX_CHANGE**: Config PHY Tx Swing. Controls the PHY transmitter voltage swing level. The controller drives the mac_phy_txswing output from this register bit field. Note: The access attributes of this field are as follows: - Wire: R/W (sticky) - Dbi: R/W (sticky) Note: This register field is sticky.

Bit	R/W	Reset	Description
17	RW	0x1	**DIRECT_SPEED_CHANGE**: Directed Speed Change. When the speed change occurs, the controller will clear the contents of this field; and a read to this field by your software will return a '0'. To manually initiate the speed change: - Write to LINK_CONTROL2_LINK_STATUS2_REG.PCIE_CAP_TARGET_LINK_SPEED in the local device - Deassert this field - Assert this field If you set the default of this field using the DEFAULT_GEN2_SPEED_CHANGE configuration parameter to '1', then the speed change is initiated automatically after link up, and the controller clears the contents of this field. If you want to prevent this automatic speed change, then write a lower speed value to the Target Link Speed field of the Link Control 2 register (LINK_CONTROL2_LINK_STATUS2_OFF.PCIE_CAP_TARGET_LINK_SPEED) through the DBI before link up. Note: The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
16	RW	0x1	**AUTO_LANE_FLIP_CTRL_EN**: Enable Auto flipping of the lanes. You must set the CX_AUTO_LANE_FLIP_CTRL_EN configuration parameter to include the hardware for this feature in the controller. For more details, see the 'Lane Reversal' appendix in the Databook. Note: The access attributes of this field are as follows: - Wire: R/W (sticky) - Dbi: R/W (sticky) Note: This register field is sticky.
15:13	RW	0x0	**PRE_DET_LANE**: Predetermined Lane for Auto Flip. This field defines which physical lane is connected to logical Lane0 by the flip operation performed in Detect. This field is used to restrict the receiver detect procedure to a particular lane when the default detect and polling procedure performed on all lanes cannot be successful. A notable example of when it is useful to program this field to a value different from the default, is when a lane is asymmetrically broken, that is, it is detected in Detect LTSSM state but it cannot exit Electrical Idle in Polling LTSSM state. Note: The access attributes of this field are as follows: - Wire: R/W (sticky) - Dbi: R/W (sticky) Note: This register field is sticky.
12:8	RW	0x4	**NUM_OF_LANES**: Predetermined Number of Lanes. Defines the number of lanes which are connected and not bad. Used to limit the effective link width to ignore 'broken" or "unused" lanes that detect a receiver. Indicates the number of lanes to check for exit from Electrical Idle in Polling.Active and L2.Idle. It is possible that the LTSSM might detect a receiver on a bad or broken lane during the Detect Substate. However, it is also possible that such a lane might also fail to exit Electrical Idle and therefore prevent a valid link from being configured. This value is referred to as the "Predetermined Number of Lanes" in section 4.2.6.2.1 of the PCI Express Base Specification. When you have unused lanes in your system, then you must change the value in this register to reflect the number of lanes. You must also change the value in the "Link Mode Enable" field of PORT_LINK_CTRL_OFF. The value in this register is normally the same as the encoded value in PORT_LINK_CTRL_OFF. If you find that one of your used lanes is bad then you must reduce the value in this register. For more information, see "How to Tie Off Unused Lanes" in the Databook. For information on upsizing and downsizing the link width, see "Link Establishment" in the Databook. Note: The access attributes of this field are as follows: - Wire: R/W (sticky) - Dbi: R/W (sticky) Note: This register field is sticky.

Bit	R/W	Reset	Description
7:0	RW	0x34	**FAST_TRAINING_SEQ**: Sets the Number of Fast Training Sequences (N_FTS) that the controller advertises as its N_FTS during Gen2 or Gen3 link training. This value is used to inform the link partner about the PHY's ability to recover synchronization after a low power state. The number should be provided by the PHY vendor. Do not set N_FTS to zero; doing so can cause the LTSSM to go into the recovery state when exiting from L0s. Note: The access attributes of this field are as follows: - Wire: R/W (sticky) - Dbi: R/W (sticky) Note: This register field is sticky.

PCIE_X8_EP_PF0_PORT_LOGIC_GEN2_CTRL_OFF_0

Reset: 0x00030834 (0b00xx,0000,0000,0011,0000,1000,0011,0100)

Bit	R/W	Reset	Description
Bit	**R/W**	**Reset**	**Description**
31	RO	0x0	**RSVDP_31**: Reserved for future use.
30	RW	0x0	**FORCE_LANE_FLIP**: Enable to force the LANE_UNDER_TEST physical lane flips to logical lane 0. All the other physical lanes are turned off. The LINK_CAPABLE register must be set to 1 and only x1 link can be formed if the FORCE_LANE_FLIP register is set to 1. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: R/W (sticky) Note: This register field is sticky.
27:24	RW	0x0	**LANE_UNDER_TEST**: The Lane Under Test is the lane for Forced Lane Flip or for Loopback Eq. Only one lane is configured each time. The default of this field is the CX_DEFAULT_LANE_UNDER_TEST configuration parameter. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: R/W (sticky) Note: This register field is sticky.
23	RW	0x0	**SELECTABLE_DEEMPH_BIT_MUX**: The selectable deemphasis bit (Symbol 4 bit 6) of the transmitted TS2 Ordered Sets for DSP in Recovery.RcvrCfg state is muxed between the Selectable De-emphasis field in the Link Control 2 register and the value requested by USP in Recovery.RcvrLock state through Tx TS1s from USP. Note: This register field is sticky.
22	RW	0x0	**SELECT_DEEMPH_VAR_MUX**: The select_deemphasis variable for DSP on entry to Recovery.RcvrCfg state is muxed between the Selectable De-emphasis field in the Link Control 2 register and the value requested by the Upstream Port in the eight consecutive TS1 Ordered Sets it received. Note: This register field is sticky.
21	RW	0x0	**GEN1_EI_INFERENCE**: Electrical Idle Inference Mode at Gen1 Rate. Programmable mode to determine inferred electrical idle (EI) in Recovery.Speed or Loopback.Active (as slave) state at Gen1 speed by looking for a '1' value on RxElecIdle instead of looking for a '0' on RxValid. If the PHY fails to deassert the RxValid signal in Recovery.Speed or Loopback.Active (because of corrupted EIOS for example), then EI cannot be inferred successfully in the controller by just detecting the condition RxValid=0. Note: This register field is sticky.

Bit	R/W	Reset	Description
Bit	**R/W**	**Reset**	**Description**
20	RW	0x0	**SEL_DEEMPHASIS**: Used to set the de-emphasis level for upstream ports. This bit selects the level of de-emphasis the link operates at. Note: The access attributes of this field are as follows: - Wire: R/W (sticky) - Dbi: R/W (sticky) Note: This register field is sticky.
19	RW	0x0	**CONFIG_TX_COMP_RX**: Config Tx Compliance Receive Bit. Note: The access attributes of this field are as follows: - Wire: R/W (sticky) - Dbi: R/W (sticky) Note: This register field is sticky.
18	RW	0x0	**CONFIG_PHY_TX_CHANGE**: Config PHY Tx Swing. Controls the PHY transmitter voltage swing level. The controller drives the mac_phy_txswing output from this register bit field. Note: The access attributes of this field are as follows: - Wire: R/W (sticky) - Dbi: R/W (sticky) Note: This register field is sticky.
17	RW	0x1	**DIRECT_SPEED_CHANGE**: Directed Speed Change. When the speed change occurs, the controller will clear the contents of this field; and a read to this field by your software will return a '0'. To manually initiate the speed change: - Write to LINK_CONTROL2_LINK_STATUS2_REG.PCIE_CAP_TARGET_LINK_SPEED in the local device - Deassert this field - Assert this field If you set the default of this field using the DEFAULT_GEN2_SPEED_CHANGE configuration parameter to '1', then the speed change is initiated automatically after link up, and the controller clears the contents of this field. If you want to prevent this automatic speed change, then write a lower speed value to the Target Link Speed field of the Link Control 2 register (LINK_CONTROL2_LINK_STATUS2_OFF.PCIE_CAP_TARGET_LINK_SPEED) through the DBI before link up. Note: The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
16	RW	0x1	**AUTO_LANE_FLIP_CTRL_EN**: Enable Auto flipping of the lanes. You must set the CX_AUTO_LANE_FLIP_CTRL_EN configuration parameter to include the hardware for this feature in the controller. For more details, see the 'Lane Reversal' appendix in the Databook. Note: The access attributes of this field are as follows: - Wire: R/W (sticky) - Dbi: R/W (sticky) Note: This register field is sticky.
15:13	RW	0x0	**PRE_DET_LANE**: Predetermined Lane for Auto Flip. This field defines which physical lane is connected to logical Lane0 by the flip operation performed in Detect. This field is used to restrict the receiver detect procedure to a particular lane when the default detect and polling procedure performed on all lanes cannot be successful. A notable example of when it is useful to program this field to a value different from the default, is when a lane is asymmetrically broken, that is, it is detected in Detect LTSSM state but it cannot exit Electrical Idle in Polling LTSSM state. Note: The access attributes of this field are as follows: - Wire: R/W (sticky) - Dbi: R/W (sticky) Note: This register field is sticky.

Bit	R/W	Reset	Description
Bit	**R/W**	**Reset**	**Description**
12:8	RW	0x8	**NUM_OF_LANES**: Predetermined Number of Lanes. Defines the number of lanes which are connected and not bad. Used to limit the effective link width to ignore 'broken" or "unused" lanes that detect a receiver. Indicates the number of lanes to check for exit from Electrical Idle in Polling.Active and L2.Idle. It is possible that the LTSSM might detect a receiver on a bad or broken lane during the Detect Substate. However, it is also possible that such a lane might also fail to exit Electrical Idle and therefore prevent a valid link from being configured. This value is referred to as the "Predetermined Number of Lanes" in section 4.2.6.2.1 of the PCI Express Base Specification. When you have unused lanes in your system, then you must change the value in this register to reflect the number of lanes. You must also change the value in the "Link Mode Enable" field of PORT_LINK_CTRL_OFF. The value in this register is normally the same as the encoded value in PORT_LINK_CTRL_OFF. If you find that one of your used lanes is bad then you must reduce the value in this register. For more information, see "How to Tie Off Unused Lanes" in the Databook. For information on upsizing and downsizing the link width, see "Link Establishment" in the Databook. Note: The access attributes of this field are as follows: - Wire: R/W (sticky) - Dbi: R/W (sticky) Note: This register field is sticky.
7:0	RW	0x34	**FAST_TRAINING_SEQ**: Sets the Number of Fast Training Sequences (N_FTS) that the controller advertises as its N_FTS during Gen2 or Gen3 link training. This value is used to inform the link partner about the PHY's ability to recover synchronization after a low power state. The number should be provided by the PHY vendor. Do not set N_FTS to zero; doing so can cause the LTSSM to go into the recovery state when exiting from L0s. Note: The access attributes of this field are as follows: - Wire: R/W (sticky) - Dbi: R/W (sticky) Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_PHY_STATUS_OFF_0

where <*j*> = 4, 8.

Description:Memory mapped register from phy_cfg_status GPIO input pins.

PCIE_X4_EP_PF0_PORT_LOGIC_PHY_STATUS_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_PHY_STATUS_OFF_0

Offset: 0x810
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PHY_STATUS**: PHY Status. Data received directly from the phy_cfg_status bus. These is a GPIO register reflecting the values on the static phy_cfg_status input signals. The usage is left completely to the user and does not in any way influence controller functionality. You can use it for any static sideband status signalling requirements that you have for your PHY. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_PHY_CONTROL_OFF_0

where <*j*> = 4, 8.

Description:Memory mapped register to cfg_phy_control GPIO output pins.

PCIE_X4_EP_PF0_PORT_LOGIC_PHY_CONTROL_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_PHY_CONTROL_OFF_0

Offset: 0x814
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**PHY_CONTROL**: PHY Control. Data sent directly to the cfg_phy_control bus. This is a GPIO register driving the values on the static cfg_phy_control output signals, and does not in any way influence controller functionality. It can be used for any static sideband control signaling requirements that you have for your PHY. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_TRGT_MAP_CTRL_OFF_0

where <*j*> = 4, 8.

Description:This register controls the target map.

PCIE_X4_EP_PF0_PORT_LOGIC_TRGT_MAP_CTRL_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_TRGT_MAP_CTRL_OFF_0

Offset: 0x81c
Read/Write: See table below

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000006b (0b0000,0000,0000,0000,000x,xxxx,x110,1011)

Bit	R/W	Reset	Description
31:21	RO	0x0	**TARGET_MAP_RESERVED_21_31**: Reserved. Note: The access attributes of this field are as follows: - Wire: RSVDP - Dbi: R (sticky)
20:16	RW	0x0	**TARGET_MAP_INDEX**: The number of the PF Function on which the Target Values are set. This register does not respect the Byte Enable setting, any write will affect all register bits.
15:13	RO	0x0	**TARGET_MAP_RESERVED_13_15**: Reserved. Note: The access attributes of this field are as follows: - Wire: RSVDP - Dbi: R (sticky)
6	RW	0x1	**TARGET_MAP_ROM**: Target Value for the ROM page of the PF Function selected by the index number. This register does not respect the Byte Enable setting, any write will affect all register bits.
5:0	RW	0x2b	**TARGET_MAP_PF**: Target Values for each BAR on the PF Function selected by the index number. This register does not respect the Byte Enable setting, any write will affect all register bits.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_CLOCK_GATING_CTRL_OFF_0

where <*j*> = 4, 8.

Description:This register enables you to disable dynamic clock gating. By default dynamic clock gating is on, allowing the controller to autonomously enable and disable its clocks. The clock gating is performed in the clock and reset module, DWC_pcie_clk_rst.v, and is initiated by the controllers clock enable signals. The following modules support dynamic clock gating: - AXI Bridge - RADM.

PCIE_X4_EP_PF0_PORT_LOGIC_CLOCK_GATING_CTRL_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_CLOCK_GATING_CTRL_OFF_0

Offset: 0x88c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000003 (0b0000,0000,0000,0000,0000,0000,0000,0011)

Bit	R/W	Reset	Description
31:2	RO	0x0	**RSVDP_2**: Reserved for future use.
1	RW	0x1	**AXI_CLK_GATING_EN**: AXI Clock Gating Enable. This register enables the AXI Bridge to autonomously enable and disable the AXI Master clock, the AXI Slave clock and the AXI DBI slave clock. The DWC_pcie_clk_rst.v module provides the gated clock, mstr_axi_aclk_gated, to the AXI Bridge and is enabled when the controllers clock enable signal, mstr_aclk_active, is asserted. For the AXI Slave this module provides the gated clock, slv_axi_aclk_gated, to the AXI Bridge and is enabled when the controllers clock enable signal, slv_aclk_active, is asserted. If the AXI DBI Slave is enabled (DBI_4SLAVE_POPULATED=1) the module provides the gated clock, dbi_axi_aclk_gated, to the AXI Bridge and is enabled when the controllers clock enable signal, dbi_aclk_active, is asserted. The controller de-asserts the clock enable signals when the respective AXI Master/Slave interfaces are idle. Note: This register field is sticky.
0	RW	0x1	**RADM_CLK_GATING_EN**: RADM Clock Gating Enable. This register, if set, enables the RADM to autonomously enable and disable its clock. The DWC_pcie_clk_rst.v module provides the gated clock, radm_clk_g, to the RADM and is enabled when the controllers clock enable signal, en_radm_clk_g, is asserted. The RADM clock is a gated version of the controller clock, core_clk. The controller de-asserts en_radm_clk_g when there is no Rx traffic, Rx queues and pre/post-queue pipelines are empty, RADM completion LUT is empty, and there are no FLR actions pending. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_GEN3_RELATED_OFF_0

where <*j*> = 4, 8.

Description:There is no Gen3-specific N_FTS field. The N_FTS field in the "Link Width and Speed Change Control Register" is used for both Gen2 and Gen3 speed modes. There is no Gen3-specific "Directed Speed Change" field. The "Directed Speed Change" field in the "Link Width and Speed Change Control Register" is used to change to Gen2 or Gen3 speed. A speed change to Gen3 occurs if (1) the "Directed Speed Change" field is set to '1' and (2) the "Target Link Speed" field in the Link Control 2 Register is set to Gen3. Gen3 support is advertised by both sides of the link during link training.

PCIE_X4_EP_PF0_PORT_LOGIC_GEN3_RELATED_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_GEN3_RELATED_OFF_0

Offset: 0x890
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00400002 (0b0000,0000,0100,0000,0000,0000,0000,0010)

Bit	R/W	Reset	Description
31:26	RO	0x0	**RSVDP_26**: Reserved for future use.
25:24	RW	0x0	**RATE_SHADOW_SEL**: Rate Shadow Select. This register value decide the Data Rate of shadow register. The following shadow registers are controlled by this register. - GEN3_RELATED_OFF[9] EQ_PHASE_2_3 - GEN3_RELATED_OFF[12] RXEQ_PH01_EN - GEN3_RELATED_OFF[19] RE_EQ_REQUEST_ENABLE - GEN3_RELATED_OFF[21] AUTO_EQ_DISABLE - GEN3_RELATED_OFF[22] USP_SEND_8GT_EQ_TS2_DISABLE - GEN3_EQ_LOCAL_FS_LF_OFF[5:0] GEN3_EQ_LOCAL_LF - GEN3_EQ_LOCAL_FS_LF_OFF[11:6] GEN3_EQ_LOCAL_FS - GEN3_EQ_PSET_COEFF_MAP_0[5:0] GEN3_EQ_PRE_CURSOR_PSET - GEN3_EQ_PSET_COEFF_MAP_0[11:6] GEN3_EQ_CURSOR_PSET - GEN3_EQ_PSET_COEFF_MAP_0[17:12] GEN3_EQ_POSET_CURSOR_PSET - GEN3_EQ_CONTROL_OFF[3:0] GEN3_EQ_FB_MODE - GEN3_EQ_CONTROL_OFF[4] GEN3_EQ_PHASE23_EXIT_MODE - GEN3_EQ_CONTROL_OFF[5] GEN3_EQ_EVAL_2MS_DISABLE - GEN3_EQ_CONTROL_OFF[23:8] GEN3_EQ_PSET_REQ_VEC - GEN3_EQ_CONTROL_OFF[24] GEN3_EQ_FOM_INC_INITIAL_EVAL - GEN3_EQ_CONTROL_OFF[25] GEN3_EQ_PSET_REQ_AS_COEF - GEN3_EQ_FB_MODE_DIR_CHANGE_OFF[4:0] GEN3_EQ_FMDC_T_MIN_PHASE23 - GEN3_EQ_FB_MODE_DIR_CHANGE_OFF[9:5] GEN3_EQ_FMDC_N_EVALS - GEN3_EQ_FB_MODE_DIR_CHANGE_OFF[13:10] GEN3_EQ_FMDC_MAX_PRE_CUSROR_DELTA - GEN3_EQ_FB_MODE_DIR_CHANGE_OFF[17:14] GEN3_EQ_FMDC_MAX_POST_CUSROR_DELTA Note: This register field is sticky.
23	RW	0x0	**GEN3_EQ_INVREQ_EVAL_DIFF_DISABLE**: Eq InvalidRequest and RxEqEval Different Time Assertion Disable. Disable the assertion of Eq InvalidRequest and RxEqEval at different time. Note: When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shared for Gen3 and Gen4/Gen5 data rate. Note: This register field is sticky.
22	RW	0x1	**USP_SEND_8GT_EQ_TS2_DISABLE**: Upstream Port Send 8GT/s or 16GT/s EQ TS2 Disable. The base spec defines that USP can optionally send 8GT or 16GT EQ TS2, which implies that USP can set DSP TxPreset value in Gen4 or Gen5 Data Rate. This applies to upstream ports only; It does not apply to downstream ports. Note: - When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shadow register for Gen3 and Gen4/Gen5 data rate. - If RATE_SHADOW_SEL==00b, this register is RSVD. - If RATE_SHADOW_SEL==01b, this register is for Gen4 data rate. - If RATE_SHADOW_SEL==10b, this register is for Gen5 data rate. Value after reset in Gen4/Gen5 is 0x1. Note: The access attributes of this field are as follows: - Wire: see description - Dbi: see description Note: This register field is sticky.
21	RW	0x0	**AUTO_EQ_DISABLE**: Autonomous Equalization Disable. Note: - When CX_GEN4_SPEED/ CX_GEN5_SPEED, this register is shadow register for Gen3 and Gen4/ Gen5 data rate. - If RATE_SHADOW_SEL==00b, this register is RSVD. - If RATE_SHADOW_SEL==01b, this register is for Gen4 data rate. - If RATE_SHADOW_SEL==10b, this register is for Gen5 data rate. Note: The access attributes of this field are as follows: - Wire: see description - Dbi: see description Note: This register field is sticky.
20:19	RO	0x0	**RSVDP_19**: Reserved for future use.

Bit	R/W	Reset	Description
18	RW	0x0	**GEN3_DC_BALANCE_DISABLE**: DC Balance Disable. Disable DC Balance feature. Note: When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shared for Gen3 and Gen4/Gen5 data rate. Note: This register field is sticky.
17	RW	0x0	**GEN3_DLLP_XMT_DELAY_DISABLE**: DLLP Transmission Delay Disable. Disable delay transmission of DLLPs before equalization. Note: When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shared for Gen3 and Gen4/Gen5 data rate. Note: This register field is sticky.
16	RW	0x0	**GEN3_EQUALIZATION_DISABLE**: Equalization Disable. Disable equalization feature. This bit cannot be changed once the LTSSM starts link training. Note: When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shared for Gen3 and Gen4/Gen5 data rate. Note: This register field is sticky.
15:14	RO	0x0	**RSVDP_14**: Reserved for future use.
13	RW	0x0	**RXEQ_RGRDLESS_RXTS**: When set to '1', the controller as Gen3 EQ master asserts RxEqEval to instruct the PHY to do Rx adaptation and evaluation after a 500ns timeout from a new preset request. Note: When CX_GEN4_SPEED/ CX_GEN5_SPEED, this register is shadow register for Gen3 and Gen4/ Gen5 data rate. If RATE_SHADOW_SEL==00b, this register is for Gen3 data rate. If RATE_SHADOW_SEL==01b, this register is for Gen4 data rate. If RATE_SHADOW_SEL==10b, this register is for Gen5 data rate. Note: The access attributes of this field are as follows: - Wire: see description - Dbi: see description Note: This register field is sticky.
12	RW	0x0	**RXEQ_PH01_EN**: Rx Equalization Phase 0/Phase 1 Hold Enable. When this bit is set the upstream port holds phase 0 (the downstream port holds phase 1) for 10ms. Holding phase 0 or phase 1 can be used to allow sufficient time for Rx Equalization to be performed by the PHY. This bit is used during Virtex-7 Gen3 equalization. The programmable bits [RXEQ_PH01_EN, EQ_PHASE_2_3] can be used to obtain the following variations of the equalization procedure: Note: When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shadow register for Gen3 and Gen4/Gen5 data rate. If RATE_SHADOW_SEL==00b, this register is for Gen3 data rate. If RATE_SHADOW_SEL==01b, this register is for Gen4 data rate. If RATE_SHADOW_SEL==10b, this register is for Gen5 data rate. Note: The access attributes of this field are as follows: - Wire: see description - Dbi: see description Note: This register field is sticky.
11	RW	0x0	**EQ_REDO**: Equalization Redo Disable. Disable autonomous mechanism for requesting to redo the equalization process. The received presets or coefficients mismatch in Recovery.RcvrLock after Recovery EQ phases causes the EQ redo requests. If the EQ redo is infinite or you do not want eq requests and redo, setting this bit to 1 will stop the EQ requests and EQ redo so that the link can go ahead to L0 state for packet trasmissions. Note: When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shared for Gen3 and Gen4/Gen5 data rate. Note: This register field is sticky.
10	RW	0x0	**EQ_EIEOS_CNT**: Equalization EIEOS Count Reset Disable. Disable requesting reset of EIEOS count during equalization. Note: When CX_GEN4_SPEED/ CX_GEN5_SPEED, this register is shared for Gen3 and Gen4/Gen5 data rate. Note: This register field is sticky.

Bit	R/W	Reset	Description
9	RW	0x0	**EQ_PHASE_2_3**: Equalization Phase 2 and Phase 3 Disable. This applies to downstream ports only. Note: When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shadow register for Gen3 and Gen4/Gen5 data rate. If RATE_SHADOW_SEL==00b, this register is for Gen3 data rate. If RATE_SHADOW_SEL==01b, this register is for Gen4 data rate. If RATE_SHADOW_SEL==10b, this register is for Gen5 data rate. Note: The access attributes of this field are as follows: - Wire: see description - Dbi: see description Note: This register field is sticky.
8	RW	0x0	**DISABLE_SCRAMBLER_GEN_3**: Disable Scrambler for Gen3 and Gen4 Data Rate. The Gen3 and Gen4 scrambler/descrambler within the controller needs to be disabled when the scrambling function is implemented outside of the controller (for example within the PHY). Note: When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shared for Gen3 and Gen4/Gen5 data rate. Note: This register field is sticky.
7:2	RO	0x0	**RSVDP_2**: Reserved for future use.
1	RW	0x1	**NO_SEED_VALUE_CHANGE**: If this bit is set to 1, the seed value of LFSR for scrambler at Gen3 rate does not change after LinkUp = 1. This bit takes effect only when CX_AUTO_LANE_FLIP_CTRL_EN is supported. This feature requires both sides of the link support it. Note: this register is shared for Gen3 and Gen4/Gen5 data rates. Note: This register field is sticky.
0	RW	0x0	**GEN3_ZRXDC_NONCOMPL**: Gen3 Receiver Impedance ZRX-DC Not Compliant. Receivers that operate at 8.0 GT/s with an impedance other than the range defined by the ZRX-DC parameter for 2.5 GT/s (40-60 Ohms) must meet additional behavior requirements in the following LTSSM states: Polling, Rx_L0s, L1, L2, and Disabled. Note: When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shared for Gen3 and Gen4/Gen5 data rates. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_GEN3_EQ_CONTROL_OFF_0

where <*j*> = 4, 8.

Description:This register controls equalization for Phase2 in an upstream port (USP), or Phase3 in a downstream port (DSP).

PCIE_X4_EP_PF0_PORT_LOGIC_GEN3_EQ_CONTROL_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_GEN3_EQ_CONTROL_OFF_0

Offset: 0x8a8
Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x05000f60 (0b0000,0101,0000,0000,0000,1111,0110,0000)

Bit	R/W	Reset	Description
31:27	RO	0x0	**RSVDP_27**: Reserved for future use.
26	RW	0x1	**GEN3_REQ_SEND_CONSEC_EIEOS_FOR_PSET_MAP**: Request controller to send back-to-back EIEOS in Recovery.RcvrLock state until presets to coefficients mapping is complete. Note: Gen3 and Gen4 share the same register bit and have the same feature. Note: This register field is sticky.
25	RW	0x0	**GEN3_EQ_PSET_REQ_AS_COEF**: GEN3_EQ_PSET_REQ_AS_COEF is an internally reserved field. Do not use. Note: This register field is sticky.
24	RW	0x1	**GEN3_EQ_FOM_INC_INITIAL_EVAL**: Include Initial FOM. Include or not the FOM feedback from the initial preset evaluation performed in the EQ Master, when finding the highest FOM among all preset evaluations. Note: - When CX_GEN4_SPEED/ CX_GEN5_SPEED, this register is shadow register for Gen3 and Gen4/Gen5 data rate. - If RATE_SHADOW_SEL==00b, this register is for Gen3 data rate. - If RATE_SHADOW_SEL==01b, this register is for Gen4 data rate. - If RATE_SHADOW_SEL==10b, this register is for Gen5 data rate. Note: This register field is sticky.
23:8	RW	0xf	**GEN3_EQ_PSET_REQ_VEC**: Preset Request Vector. Requesting of Presets during the initial part of the EQ Master Phase. Encoding scheme is as follows: Bit [15:0] =0x0: No preset is requested and evaluated in EQ Master Phase. Bit [i] =1: "Preset=i" is requested and evaluated in EQ Master Phase. - 0000000000000000: No preset be requested and evaluated in EQ Master Phase - 000000xxxxxxxxx1: Preset 0 is requested and evaluated in EQ Master Phase - 000000xxxxxxxx1x: Preset 1 is requested and evaluated in EQ Master Phase - 000000xxxxxxx1xx: Preset 2 is requested and evaluated in EQ Master Phase - 000000xxxxxx1xxx: Preset 3 is requested and evaluated in EQ Master Phase - 000000xxxxx1xxxx: Preset 4 is requested and evaluated in EQ Master Phase - 000000xxxx1xxxxx: Preset 5 is requested and evaluated in EQ Master Phase - 000000xxx1xxxxxx: Preset 6 is requested and evaluated in EQ Master Phase - 000000xx1xxxxxxx: Preset 7 is requested and evaluated in EQ Master Phase - 000000x1xxxxxxxx: Preset 8 is requested and evaluated in EQ Master Phase - 00000x1xxxxxxxxx: Preset 9 is requested and evaluated in EQ Master Phase - 000001xxxxxxxxxx: Preset 10 is requested and evaluated in EQ Master Phase - All other encodings: Reserved Note: You must contact your PHY vendor to ensure 24 ms timeout does not occur in presets requests in EQ master phase, that is, you must set a proper value to the GEN3_EQ_PSET_REQ_VEC register so that the EQ tunning for Figure of Merit in the EQ master phase completes before 24 ms timeout. Note: - When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shadow register for Gen3 and Gen4/Gen5 data rate. - If RATE_SHADOW_SEL==00b, this register is for Gen3 data rate. - If RATE_SHADOW_SEL==01b, this register is for Gen4 data rate. - If RATE_SHADOW_SEL==10b, this register is for Gen5 data rate. Note: This register field is sticky.
7	RO	0x0	**RSVDP_7**: Reserved for future use.
6	RW	0x1	**GEN3_LOWER_RATE_EQ_REDO_ENABLE**: Support EQ redo and lower rate change. Note: Gen3 and Gen4 share the same register bit and have the same feature. Note: This register field is sticky.

Bit	R/W	Reset	Description
5	RW	0x1	**GEN3_EQ_EVAL_2MS_DISABLE**: Phase2_3 2 ms Timeout Disable. Determine behavior in Phase2 for USP (Phase3 if DSP) when the PHY does not respond within 2ms to the assertion of RxEqEval. Note: - When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shadow register for Gen3 and Gen4/Gen5 data rate. - If RATE_SHADOW_SEL==00b, this register is for Gen3 data rate. - If RATE_SHADOW_SEL==01b, this register is for Gen4 data rate. - If RATE_SHADOW_SEL==10b, this register is for Gen5 data rate. Note: This register field is sticky.
4	RW	0x0	**GEN3_EQ_PHASE23_EXIT_MODE**: Behavior After 24 ms Timeout (when optimal settings are not found). For a USP: Determine next LTSSM state from Phase2 after 24ms Timeout - 0: Recovery.Speed - 1: Recovery.Equalization.Phase3 When optimal settings are not found then: - Equalization Phase 2 Successful status bit is not set in the "Link Status Register 2" when GEN3_EQ_PHASE23_EXIT_MODE = 0 - Equalization Phase 2 Successful status bit is set in the "Link Status Register 2" when GEN3_EQ_PHASE23_EXIT_MODE = 1 - Equalization Phase 2 Complete status bit is set in the "Link Status Register 2" For a DSP: Determine next LTSSM state from Phase3 after 24ms Timeout - 0: Recovery.Speed - 1: Recovery.Equalization.RcvrLock When optimal settings are not found then: - Equalization Phase 3 Successful status bit is not set in the "Link Status Register 2" when GEN3_EQ_PHASE23_EXIT_MODE = 0 - Equalization Phase 3 Successful status bit is set in the "Link Status Register 2" when GEN3_EQ_PHASE23_EXIT_MODE = 1 - Equalization Phase 3 Complete status bit is set in the "Link Status Register 2" Note: GEN3_EQ_PHASE23_EXIT_MODE = 1 affects Direction Change feed back mode. EQ requests for Figure Of Merit mode complete before 24 ms timeout. Please see GEN3_EQ_PSET_REQ_VEC Register for more. Note: - When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is shadow register for Gen3 and Gen4/Gen5 data rate. - If RATE_SHADOW_SEL==00b, this register is for Gen3 data rate. - If RATE_SHADOW_SEL==01b, this register is for Gen4 data rate. - If RATE_SHADOW_SEL==10b, this register is for Gen5 data rate. Note: This register field is sticky.
3:0	RW	0x0	**GEN3_EQ_FB_MODE**: Feedback Mode. Other values are reserved. Note: - When CX_GEN4_SPEED/CX_GEN5_SPEED, this register is a shadow register for Gen3 and Gen4/Gen5 data rate. - If RATE_SHADOW_SEL==00b, this register is for Gen3 data rate. - If RATE_SHADOW_SEL==01b, this register is for Gen4 data rate. - If RATE_SHADOW_SEL==10b, this register is for Gen5 data rate. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_ORDER_RULE_CTRL_OFF_0

where <*j*> = 4, 8.

Description:This register controls the order rule.

PCIE_X4_EP_PF0_PORT_LOGIC_ORDER_RULE_CTRL_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_ORDER_RULE_CTRL_OFF_0

Offset: 0x8b4
Read/Write: See table below
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:16	RO	0x0	**RSVDP_16**: Reserved for future use.
15:8	RW	0x0	**CPL_PASS_P**: Completion Passing Posted Ordering Rule Control. Determines if CPL can pass halted P queue.
7:0	RW	0x0	**NP_PASS_P**: Non-Posted Passing Posted Ordering Rule Control. Determines if NP can pass halted P queue.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_PIPE_LOOPBACK_CONTROL_OFF_0

where <*j*> = 4, 8.

Description:This register controls the PIPE Loopback.

PCIE_X4_EP_PF0_PORT_LOGIC_PIPE_LOOPBACK_CONTROL_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_PIPE_LOOPBACK_CONTROL_OFF_0

Offset: 0x8b8
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000f (0b0000,0000,0000,0000,0000,0000,0000,1111)

Bit	R/W	Reset	Description
31	RW	0x0	**PIPE_LOOPBACK**: PIPE Loopback Enable. Indicates RMMI Loopback if M-PCIe. Note: This register field is sticky.
30:27	RO	0x0	**RSVDP_27**: Reserved for future use.
26:24	WO	0x0	**RXSTATUS_VALUE**: RXSTATUS_VALUE is an internally reserved field. Do not use.
23:22	RO	0x0	**RSVDP_22**: Reserved for future use.
21:16	RW	0x0	**RXSTATUS_LANE**: RXSTATUS_LANE is an internally reserved field. Do not use. Note: This register field is sticky.

Bit	R/W	Reset	Description
15:0	RW	0xf	**LPBK_RXVALID**: LPBK_RXVALID is an internally reserved field. Do not use. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_MISC_CONTROL_1_OFF_0

where <*j*> = 4, 8.

Description:This is the DBI Read-Only write enable register.

PCIE_X4_EP_PF0_PORT_LOGIC_MISC_CONTROL_1_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_MISC_CONTROL_1_OFF_0

Offset: 0x8bc
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000bff49 (0b0000,0000,0000,1011,1111,1111,0100,1001)

Bit	R/W	Reset	Description
31:23	RO	0x0	**RSVDP_23**: Reserved for future use.
22	RW	0x0	**PORT_LOGIC_WR_DISABLE**: Disable port logic register write from wire side. Note: The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: R/W (sticky) Note: This register field is sticky.
21	RW	0x0	**P2P_ERR_RPT_CTRL**: Determines whether to enable Peer to Peer (P2P) error reporting. Note: This register field is sticky.
20	RW	0x0	**P2P_TRACK_CPL_TO_REG**: Determines whether to track completion of transmitted Non-Posted TLPs in P2P mode. Note: This register field is sticky.
19:18	RW	0x2	**TARGET_ABOVE_CONFIG_LIMIT_REG**: Configuration requests with an address greater than CONFIG_LIMIT_REG are directed to either ELBI or TRGT1 interface based on the setting of this field. This field can have the following values: Note: This register field is sticky.

Bit	R/W	Reset	Description
17:8	RW	0x3ff	**CONFIG_LIMIT_REG**: Configuration requests are directed either to CDM or ELBI/RTRGT1 based on the value of this field. - Configuration requests with an address less CONFIG_LIMIT_REG are directed to the CDM - Configuration requests with an address greater than CONFIG_LIMIT_REG are directed to either ELBI or TRGT1 interface based on the setting of TARGET_ABOVE_CONFIG_LIMIT_REG field. Your application must set a proper value for this field based on your extended configuration registers. For more details, see the "CDM/ELBI Register Space Access Through CFG Request" in "Register Module, LBC, and DBI" section in the "Controller Operations" chapter of the Databook. Note: This register field is sticky.
7	RW	0x0	**CFG_TLP_BYPASS_EN_REG**: Setting of this field defines how to decide the destination of Configuration requests. Note: When app_req_retry_en is asserted, the setting of this field is ignored. Note: This register field is sticky.
6	RW	0x1	**CPLQ_MNG_EN**: This field enables the Completion Queue Management feature. Note: This register field is sticky.
5	RW	0x0	**ARI_DEVICE_NUMBER**: When ARI is enabled, this field enables use of the device ID. Note: This register field is sticky.
4	RW	0x0	**DISABLE_AUTO_LTR_CLR_MSG**: Disable the autonomous generation of LTR clear message in upstream port. This field can have the following values: For more details, see "Latency Tolerance Reporting (LTR) Message Generation [EP Mode]" in "Message Generation" section of the "Controller Operations" chapter of the Databook. Note: This register field is sticky.
3	RW	0x1	**SIMPLIFIED_REPLAY_TIMER**: Enables Simplified Replay Timer (Gen4). For more details, see "Transmit Replay" in "Transmit TLP Processing" section in the "Controller Operations" chapter of the Databook. Simplified Replay Timer can have the following Values: - A value from 24,000 to 31,000 Symbol Times when Extended Synch is 0b. - A value from 80,000 to 100,000 Symbol Times when Extended Synch is 1b. The Simplified Replay Timer value must not be changed while the link is in use. Note: This register field is sticky.
2	RW	0x0	**UR_CA_MASK_4_TRGT1**: When this field is set to '1', the controller suppresses error logging, error message generation, and CPL generation for non-posted requests TLPs (with UR filtering status) forwarded to your application (that is, when DEFAULT_TARGET =1). For more details, see "Advanced Error Handling For Received TLPs" chapter of the Databook. Note: This register field is sticky.
1	RW	0x0	**DEFAULT_TARGET**: Default target for an IO or MEM request with UR/CA/CRS received. Based on the value of this field the controller either drops or forwards these requests to your application. For more details, see "ECRC Handling" and "Request TLP Routing Rules" in "Receive Routing" section of the "Controller Operations" chapter of the Databook. Note: This register field is sticky.

Bit	R/W	Reset	Description
0	RW	0x1	**DBI_RO_WR_EN**: Write to RO Registers Using DBI. For more details, see "Writing to Read-Only Registers" in "Register Module, LBC, and DBI" section in the "Controller Operations" chapter of the Databook. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_MULTI_LANE_CONTROL_OFF_0

where <*j*> = 4, 8.

Description:Used when upsizing or downsizing the link width through Configuration state without bringing the link down. For more details, see the "Link Establishment" section in the "Controller Operations" chapter of the Databook.

PCIE_X4_EP_PF0_PORT_LOGIC_MULTI_LANE_CONTROL_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_MULTI_LANE_CONTROL_OFF_0

Offset: 0x8c0
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:8	RO	0x0	**RSVDP_8**: Reserved for future use.
7	RW	0x0	**UPCONFIGURE_SUPPORT**: Upconfigure Support. The controller sends this value as the Link Upconfigure Capability in TS2 Ordered Sets in Configuration.Complete state. Note: This register field is sticky.
6	RW	0x0	**DIRECT_LINK_WIDTH_CHANGE**: Directed Link Width Change. - If the upconfigure_capable variable is '1' and the PCIE_CAP_HW_AUTO_WIDTH_DISABLE bit in LINK_CONTROL_LINK_STATUS_REG is '0', the controller starts upconfigure or autonomous width downsizing (to the TARGET_LINK_WIDTH value) in the Configuration state. - If TARGET_LINK_WIDTH value is 0x0, the controller does not start upconfigure or autonomous width downsizing in the Configuration state. The controller self-clears this field when the controller accepts this request.
5:0	RW	0x0	**TARGET_LINK_WIDTH**: Target Link Width.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_PHY_INTEROP_CTRL_OFF_0

where <*j*> = 4, 8.

Description:This register controls the PHY interoperability.

PCIE_X4_EP_PF0_PORT_LOGIC_PHY_INTEROP_CTRL_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_PHY_INTEROP_CTRL_OFF_0

Offset: 0x8c4
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0001e27f (0b0000,0000,0000,0001,1110,x010,0111,1111)

Bit	R/W	Reset	Description
31:18	RO	0x0	**RSVDP_18**: Reserved for future use.
17:12	RW	0x1e	**PHY_RST_TIMER**: Decide how many aux clock cycles the PHY reset lasts (0 to 63 aux clock cycles). Note: This register field is sticky.
10	RW	0x0	**L1_CLK_SEL**: L1 Clock control bit. This field is reserved for internal use. Note: This register field is sticky.
9	RO	0x1	**L1_NOWAIT_P1**: L1 entry control bit. This field is reserved for internal use. The access attributes of this field are as follows: - Wire: R/W (sticky) - Dbi: R/W (sticky) Note: This register field is sticky.
8	RW	0x0	**L1SUB_EXIT_MODE**: L1 Exit Control Using phy_mac_pclkack_n. This field is reserved for internal use. Note: This register field is sticky.
7	RO	0x0	**RSVDP_7**: Reserved for future use.

Bit	R/W	Reset	Description
6:0	RW	0x7f	**RXSTANDBY_CONTROL**: Rxstandby Control. Bits 0..5 determine if the controller asserts the RxStandby signal (mac_phy_rxstandby) in the indicated condition. Bit 6 enables the controller to perform the RxStandby/ RxStandbyStatus handshake. This field is reserved for internal use. [0]: Rx EIOS and subsequent T TX-IDLE-MIN [1]: Rate Change [2]: Inactive lane for upconfigure/downconfigure [3]: PowerDown=P1orP2 [4]: RxL0s.Idle [5]: EI Infer in L0 [6]: Execute RxStandby/RxStandbyStatus Handshake Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_TRGT_CPL_LUT_DELETE_ENTRY_OFF_0

where <*j*> = 4, 8.

Description:Using this register you can delete one entry in the target completion LUT. You should only use this register when you know that your application will never send the completion because of an FLR or any other reason. Note:: The target completion LUT (and associated target completion timeout event) is watching for completions (from your application on XALI0/1/2 or AXI master read channel) corresponding to previously received non-posted requests from the PCIe wire.

PCIE_X4_EP_PF0_PORT_LOGIC_TRGT_CPL_LUT_DELETE_ENTRY_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_TRGT_CPL_LUT_DELETE_ENTRY_OFF_0

Offset: 0x8c8
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	WO	0x0	**DELETE_EN**: This is a one-shot bit. This is a self-clearing register field. Reading from this register field always returns a '0'.
30:0	RW	0x0	**LOOK_UP_ID**: This number selects one entry to delete of the TRGT_CPL_LUT.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_LINK_FLUSH_CONTROL_OFF_0

where <*j*> = 4, 8.

Description:This register controls link reset request flush behaviour.

PCIE_X4_EP_PF0_PORT_LOGIC_LINK_FLUSH_CONTROL_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_LINK_FLUSH_CONTROL_OFF_0

Offset: 0x8cc
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xff000001 (0b1111,1111,0000,0000,0000,0000,0000,0001)

Bit	R/W	Reset	Description
31:24	RW	0xff	**RSVD_I_8**: This is an internally reserved field. Do not use. Note: This register field is sticky.
23:1	RO	0x0	**RSVDP_1**: Reserved for future use.
0	RW	0x1	**AUTO_FLUSH_EN**: Enables automatic flushing of pending requests before sending the reset request to the application logic to reset the PCIe controller and the AXI Bridge. The flushing process is initiated if any of the following events occur: - Hot reset request. A downstream port (DSP) can "hot reset" an upstream port (USP) by sending two consecutive TS1 ordered sets with the hot reset bit asserted. - Warm (Soft) reset request. Generated when exiting from D3 to D0 and cfg_pm_no_soft_rst=0. - Link down reset request. A high to low transition on smlh_req_rst_not indicates the link has gone down and the controller is requesting a reset. If you disable automatic flushing, your application is responsible for resetting the PCIe controller and the AXI Bridge. For more details see "Warm and Hot Resets" section in the Architecture chapter of the Databook. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_AMBA_ERROR_RESPONSE_DEFAULT_OFF_0

where <*j*> = 4, 8.

Description:This register holds the AXI bridge slave error responses.

PCIE_X4_EP_PF0_PORT_LOGIC_AMBA_ERROR_RESPONSE_DEFAULT_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_AMBA_ERROR_RESPONSE_DEFAULT_OFF_0

Offset: 0x8d0
Read/Write: See table below
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00009c00 (0b0000,0000,0000,0000,1001,1100,0000,0000)

Bit	R/W	Reset	Description
31:16	RO	0x0	**RSVDP_16**: Reserved for future use.
15:10	RW	0x27	**AMBA_ERROR_RESPONSE_MAP**: AXI Slave Response Error Map. Allows you to selectively map the errors received from the PCIe completion (for non-posted requests) to the AXI slave responses, slv_rresp or slv_bresp. The recommended setting is SLVERR. CRS is always mapped to OKAY. The controller sets the AXI slave read databus to 0xFFFF for all error responses. Note: This register field is sticky.
9:5	RO	0x0	**RSVDP_5**: Reserved for future use.
4:3	RW	0x0	**AMBA_ERROR_RESPONSE_CRS**: CRS Slave Error Response Mapping. Determines the AXI slave response for CRS completions. For more details see "Error Handling" in the AXI chapter of the Databook. Note: This register field is sticky.
2	RW	0x0	**AMBA_ERROR_RESPONSE_VENDORID**: Vendor ID Non-existent Slave Error Response Mapping. Determines the AXI slave response for errors on reads to non-existent Vendor ID register. For more details see "Error Handling" in the AXI chapter of the Databook. Note: This register field is sticky.
1	RO	0x0	**RSVDP_1**: Reserved for future use.
0	RW	0x0	**AMBA_ERROR_RESPONSE_GLOBAL**: Global Slave Error Response Mapping. Determines the AXI slave response for all error scenarios on non-posted requests. For more details see "Error Handling" in the AXI chapter of the Databook. The error response mapping is not applicable to Non-existent Vendor ID register reads. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_AMBA_LINK_TIMEOUT_OFF_0

where <*j*> = 4, 8.

Description:If your application AXI master issues outbound requests to the AXI bridge slave interface before the PCIe link is operational, the controller starts a "flush" timer. The timeout value of the timer is set by this register. If the timer times out before the PCIe link is operational, the bridge TX request queues are flushed. For more details, see the "AXI Bridge Initialization, Clocking and Reset" section in the AXI chapter of the Databook.

PCIE_X4_EP_PF0_PORT_LOGIC_AMBA_LINK_TIMEOUT_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_AMBA_LINK_TIMEOUT_OFF_0

Offset: 0x8d4
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000032 (0b0000,0000,0000,0000,0000,0000,0011,0010)

Bit	R/W	Reset	Description
31:9	RO	0x0	**RSVDP_9**: Reserved for future use.
8	RW	0x0	**LINK_TIMEOUT_ENABLE_DEFAULT**: Disable Flush. Note: This register field is sticky.
7:0	RW	0x32	**LINK_TIMEOUT_PERIOD_DEFAULT**: Timeout Value (ms). The timer will timeout and then flush the bridge TX request queues after this amount of time. The timer counts when there are pending outbound AXI slave interface requests and the PCIe TX link is not transmitting any of these requests. The timer is clocked by core_clk. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_AMBA_ORDERING_CTRL_OFF_0

where <*j*> = 4, 8.

Description:This register controls the AXI Bridge Ordering.

PCIE_X4_EP_PF0_PORT_LOGIC_AMBA_ORDERING_CTRL_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_AMBA_ORDERING_CTRL_OFF_0

Offset: 0x8d8
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:8	RO	0x0	**RSVDP_8**: Reserved for future use.
7	RW	0x0	**AX_MSTR_ZEROLREAD_FW**: AXI Master Zero Length Read Forward to the application. The DW PCIe controller AXI bridge is able to terminate in order with the Posted transactions the zero length read, implementing the PCIe express flush semantics of the Posted transactions. Note: This register field is sticky.
6:5	RO	0x0	**RSVDP_5**: Reserved for future use.

Bit	R/W	Reset	Description
4:3	RW	0x0	**AX_MSTR_ORDR_P_EVENT_SEL**: AXI Master Posted Ordering Event Selector. This field selects how the master interface determines when a P write is completed when enforcing the PCIe ordering rule, "NP must not pass P" at the AXI Master Interface. The AXI protocol does not support ordering between channels. Therefore, NP reads can pass P on your AXI bus fabric. This can result in an ordering violation when the read overtakes a P that is going to the same address. Therefore, the bridge master does not issue any NP requests until all outstanding P writes reach their destination. It does this by waiting for the all of the write responses on the B channel. This can affect the performance of the master read channel. For scenarios where the interconnect serializes the AXI master "AW", "W" and "AR" channels,you can increase the performance by reducing the need to wait until the complete Posted transaction has effectively reached the application slave. Note: This setting will not affect: - MSI interrupt catcher and P data ordering. This is always driven by the B'last event. - DMA read engine TLP ordering. This is always driven by the B'last event. - NP write transactions which are always serialized with P write transactions.Note: This register field is sticky.
2	RO	0x0	**RSVDP_2**: Reserved for future use.
1	RW	0x0	**AX_SNP_EN**: AXI Serialize Non-Posted Requests Enable. This field enables the AXI Bridge to serialize same ID Non-Posted Read/Write Requests on the wire. Serialization implies one outstanding same ID NP Read or Write on the wire and used to avoid AXI RAR and WAW hazards at the remote link partner. For more details, see the "Optional Serialization of AXI Slave Non-posted Requests" section in the AXI chapter of the Databook. Note: This register field is sticky.
0	RO	0x0	**RSVDP_0**: Reserved for future use.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_COHERENCY_CONTROL_1_OFF_0

where <*j*> = 4, 8.

Description:This register controls the ACE cache coherency operation.

PCIE_X4_EP_PF0_PORT_LOGIC_COHERENCY_CONTROL_1_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_COHERENCY_CONTROL_1_OFF_0

Offset: 0x8e0
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:2	RW	0x0	**CFG_MEMTYPE_BOUNDARY_LOW_ADDR**: Boundary Lower Address For Memory Type. Bits [31:0] of dword-aligned address of the boundary for Memory type. The two lower address LSBs are '00'. Addresses up to but not including this value are in the lower address space region; addresses equal or greater than this value are in the upper address space region. Note: This register field is sticky.
1	RO	0x0	**RSVDP_1**: Reserved for future use.
0	RW	0x0	**CFG_MEMTYPE_VALUE**: Sets the memory type for the lower and upper parts of the address space: Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_COHERENCY_CONTROL_2_OFF_0

where <*j*> = 4, 8.

Description:This register controls the ACE cache coherency operation.

PCIE_X4_EP_PF0_PORT_LOGIC_COHERENCY_CONTROL_2_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_COHERENCY_CONTROL_2_OFF_0

Offset: 0x8e4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CFG_MEMTYPE_BOUNDARY_HIGH_ADDR**: Boundary Upper Address For Memory Type. Bits [63:32] of the 64-bit dword-aligned address of the boundary for Memory type. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_COHERENCY_CONTROL_3_OFF_0

where <*j*> = 4, 8.

Description:This register controls the ACE cache coherency operation.

PCIE_X4_EP_PF0_PORT_LOGIC_COHERENCY_CONTROL_3_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_COHERENCY_CONTROL_3_OFF_0

Offset: 0x8e8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bx000,0xxx,x000,0xxx,x000,0xxx,x000,0xxx)

Bit	Reset	Description
30:27	0x0	**CFG_MSTR_AWCACHE_VALUE**: Master Write CACHE Signal Value. Value of the individual bits in mstr_awcache when CFG_MSTR_AWCACHE_MODE is '1'. Note: Not applicable to message requests; for message requests the value of mstr_awcache is always '0000'. Note: This register field is sticky.
22:19	0x0	**CFG_MSTR_ARCACHE_VALUE**: Master Read CACHE Signal Value. Value of the individual bits in mstr_arcache when CFG_MSTR_ARCACHE_MODE is '1'. Note: This register field is sticky.
14:11	0x0	**CFG_MSTR_AWCACHE_MODE**: Master Write CACHE Signal Behavior. Defines how the individual bits in mstr_awcache are controlled. Note: for message requests the value of mstr_awcache is always "0000" regardless of the value of this bit. Note: This register field is sticky.
6:3	0x0	**CFG_MSTR_ARCACHE_MODE**: Master Read CACHE Signal Behavior. Defines how the individual bits in mstr_arcache are controlled. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_AXI_MSTR_MSG_ADDR_LOW_OFF_0

where <*j*> = 4, 8.

Description:Lower 20 bits of the programmable AXI address to which Messages coming from wire are mapped. Bits [11:0] of the register are tied to zero for the address to be 4k-aligned. In previous releases, the third and fourth DWORDs of a message (Msg/MsgD) TLP header were delivered though the AXI master address bus (mstr_awaddr). These DWORDS are now supplied through the mstr_awmisc_info_hdr_34dw[63:0] output; and the value on mstr_awaddr is driven to the value you have programmed into the AXI_MSTR_MSG_ADDR_LOW_OFF and AXI_MSTR_MSG_ADDR_HIGH_OFF registers.

PCIE_X4_EP_PF0_PORT_LOGIC_AXI_MSTR_MSG_ADDR_LOW_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_AXI_MSTR_MSG_ADDR_LOW_OFF_0

Offset: 0x8f0
Read/Write: See table below
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:12	RW	0x0	**CFG_AXIMSTR_MSG_ADDR_LOW**: Lower 20-bits of the programmable AXI address for Messages. Note: This register field is sticky.
11:0	RO	0x0	**CFG_AXIMSTR_MSG_ADDR_LOW_RESERVED**: Reserved for future use. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_AXI_MSTR_MSG_ADDR_HIGH_OFF_0

where <*j*> = 4, 8.

Description:Upper 32 bits of the programmable AXI address to which Messages coming from wire are mapped.

PCIE_X4_EP_PF0_PORT_LOGIC_AXI_MSTR_MSG_ADDR_HIGH_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_AXI_MSTR_MSG_ADDR_HIGH_OFF_0

Offset: 0x8f4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CFG_AXIMSTR_MSG_ADDR_HIGH**: Upper 32 bits of the programmable AXI address for Messages. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_PCIE_VERSION_NUMBER_OFF_0

where <*j*> = 4, 8.

Description:The version number is given in hex format. You should convert each pair of hex characters to ASCII to interpret. Using 4.70a (GA) as an example: - VERSION_NUMBER = 0x3437302a which translates to 470* - VERSION_TYPE = 0x67612a2a which translates to ga** Using 4.70a-ea01 as an example: - VERSION_NUMBER = 0x3437302a which translates to 470* -

VERSION_TYPE = 0x65613031 which translates to ea01 GA is a general release available on www.designware.com EA is an early release available on a per-customer basis.

PCIE_X4_EP_PF0_PORT_LOGIC_PCIE_VERSION_NUMBER_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_PCIE_VERSION_NUMBER_OFF_0

Offset: 0x8f8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x3536322a (0b0011,0101,0011,0110,0011,0010,0010,1010)

Bit	Reset	Description
31:0	0x3536322a	**VERSION_NUMBER**: Version Number.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_PCIE_VERSION_TYPE_OFF_0

where <*j*> = 4, 8.

Description:The type is given in hex format. You should convert each pair of hex characters to ASCII to interpret. Using 4.70a (GA) as an example: - VERSION_NUMBER = 0x3437302a which translates to 470* - VERSION_TYPE = 0x67612a2a which translates to ga** Using 4.70a-ea01 as an example: - VERSION_NUMBER = 0x3437302a which translates to 470* - VERSION_TYPE = 0x65613031 which translates to ea01 GA is a general release available on www.designware.com EA is an early release available on a per-customer basis.

PCIE_X4_EP_PF0_PORT_LOGIC_PCIE_VERSION_TYPE_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_PCIE_VERSION_TYPE_OFF_0

Offset: 0x8fc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x65613035 (0b0110,0101,0110,0001,0011,0000,0011,0101)

Bit	Reset	Description
31:0	0x65613035	**VERSION_TYPE**: Version Type.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_INTERFACE_TIMER_CONTROL_OFF_0

where <*j*> = 4, 8.

Description:This register controls the interface timer.

PCIE_X4_EP_PF0_PORT_LOGIC_INTERFACE_TIMER_CONTROL_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_INTERFACE_TIMER_CONTROL_OFF_0

Offset: 0x930
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:5	RO	0x0	**RSVDP_5**: Reserved for future use.
4	RW	0x0	**FORCE_PENDING**: Writing to this bit forces the value of the pending flags. Note: This register field is sticky.
3:2	RW	0x0	**INTERFACE_TIMER_SCALING**: Interface timer scaling. This field can be used to reduce the timer duration for verification purpose. This field should only be programmed when the INTERFACE_TIMER_EN bit is set to 1'b0. Note: This register field is sticky.
1	RW	0x0	**INTERFACE_TIMER_AER_EN**: When set to 1 the Interface timer internal uncorrectable error generation is enabled. Note: This register field is sticky.
0	RW	0x0	**INTERFACE_TIMER_EN**: Interface timer enable. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_INTERFACE_TIMER_TARGET_OFF_0

where <*j*> = 4, 8.

Description:This is the interface timer target register.

PCIE_X4_EP_PF0_PORT_LOGIC_INTERFACE_TIMER_TARGET_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_INTERFACE_TIMER_TARGET_OFF_0

Offset: 0x934
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000032 (0b0000,0000,0000,0000,0000,0000,0011,0010)

Bit	R/W	Reset	Description
31:16	RO	0x0	**RSVDP_16**: Reserved for future use.
15:0	RW	0x32	**INTERFACE_TIMER_TARGET**: Interface timer target value. This field should only be programmed when the INTERFACE_TIMER_EN bit is set to 1'b0. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_INTERFACE_TIMER_STATUS_OFF_0

where <*j*> = 4, 8.

Description:This is the interface timer status register.

PCIE_X4_EP_PF0_PORT_LOGIC_INTERFACE_TIMER_STATUS_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_INTERFACE_TIMER_STATUS_OFF_0

Offset: 0x938
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,000x,x000,0x00)

Bit	Reset	Description
11	0x0	**SLAVE_RD_ADD_TIMEOUT**: Slave read address channel timeout.
10	0x0	**SLAVE_WR_DATA_TIMEOUT**: Slave write data channel timeout.
9	0x0	**SLAVE_WR_ADD_TIMEOUT**: Slave write address channel timeout.
6	0x0	**MASTER_RD_DATA_TIMEOUT**: Master read data channel timeout.
5	0x0	**MASTER_WR_RES_TIMEOUT**: Master write response channel timeout.

Bit	Reset	Description
4	0x0	**CLIENT2_INTERFACE_TIMEOUT**: Client2 interface timeout.
3	0x0	**CLIENT1_INTERFACE_TIMEOUT**: Client1 interface timeout.
1	0x0	**CPL_INTERFACE_TIMEOUT**: CPL interface timeout.
0	0x0	**MESSAGE_INTERFACE_TIMEOUT**: Message interface timeout.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_MSIX_ADDRESS_MATCH_LOW_OFF_0

where <*j*> = 4, 8.

Description:When you enable the MSI-X Table RAM feature (MSIX_TABLE_EN=1), the controller implements the logic and RAM required to generate MSI-X requests. For more details, see the Interrupts section in the "Controller Operations" chapter of the Databook. This register is only used in AXI configurations. When your local AXI application writes (MWr) to the address defined in this register (and MSIX_ADDRESS_MATCH_HIGH_OFF), the controller will load the MSIX_DOORBELL_OFF register with the contents of the MWr and subsequently create and send MSI-X TLPs.

PCIE_X4_EP_PF0_PORT_LOGIC_MSIX_ADDRESS_MATCH_LOW_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_MSIX_ADDRESS_MATCH_LOW_OFF_0

Offset: 0x940
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:2	RW	0x0	**MSIX_ADDRESS_MATCH_LOW**: MSI-X Address Match Low Address. Note: This register field is sticky.
1	RO	0x0	**MSIX_ADDRESS_MATCH_RESERVED_1**: Reserved. Note: This register field is sticky.
0	RW	0x0	**MSIX_ADDRESS_MATCH_EN**: MSI-X Match Enable. Enable the MSI-X Address Match feature when the AXI bridge is present. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_MSIX_ADDRESS_MATCH_HIGH_OFF_0

where <*j*> = 4, 8.

Description:MSI-X Address Match High Register. When you enable the MSI-X Table RAM feature (MSIX_TABLE_EN=1), the controller implements the logic and RAM required to generate MSI-X requests. For more details, see the Interrupts section in the "Controller Operations" chapter of the Databook. This register is only used in AXI configurations. When your local AXI application writes (MWr) to the address defined in this register (and MSIX_ADDRESS_MATCH_LOW_OFF), the controller will load the MSIX_DOORBELL_OFF register with the contents of the MWr and subsequently create and send MSI-X TLPs.

PCIE_X4_EP_PF0_PORT_LOGIC_MSIX_ADDRESS_MATCH_HIGH_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_MSIX_ADDRESS_MATCH_HIGH_OFF_0

Offset: 0x944
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MSIX_ADDRESS_MATCH_HIGH**: MSI-X Address Match High Address. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_MSIX_DOORBELL_OFF_0

where <*j*> = 4, 8.

Description:When you enable the MSI-X Table RAM feature (MSIX_TABLE_EN=1), the controller implements the logic and RAM required to generate MSI-X requests. For more details, see the Interrupts section in the "Controller Operations" chapter of the Databook. - For AXI configurations: when your local application writes (MWr) to the address defined in MSIX_ADDRESS_MATCH_LOW_OFF, the controller will load this register with the contents of the MWr and subsequently create and send MSI-X TLPs. - For non-AMBA configurations: when your local application writes to this register, the controller will create and send MSI-X TLPs.

PCIE_X4_EP_PF0_PORT_LOGIC_MSIX_DOORBELL_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_MSIX_DOORBELL_OFF_0

Offset: 0x948
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:29	0x0	**MSIX_DOORBELL_RESERVED_29_31**: Reserved.
28:24	0x0	**MSIX_DOORBELL_PF**: MSIX Doorbell Physical Function. This register determines the Physical Function for the MSI-X transaction.
23:16	0x0	**MSIX_DOORBELL_VF**: MSIX Doorbell Virtual Function. This register determines the Virtual Function for the MSI-X transaction.
15	0x0	**MSIX_DOORBELL_VF_ACTIVE**: MSIX Doorbell Virtual Function Active. This register determines whether a Virtual Function is used to generate the MSI-X transaction.
14:12	0x0	**MSIX_DOORBELL_TC**: MSIX Doorbell Traffic Class. This register determines which traffic class to generate the MSI-X transaction with.
11	0x0	**MSIX_DOORBELL_RESERVED_11**: Reserved.
10:0	0x0	**MSIX_DOORBELL_VECTOR**: MSI-X Doorbell Vector. This register determines which vector to generate the MSI-X transaction for.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_MSIX_RAM_CTRL_OFF_0

where <*j*> = 4, 8.

Description:When you enable the MSI-X Table RAM feature (MSIX_TABLE_EN=1), the controller implements the logic and RAM required to generate MSI-X requests. For more details, see the Interrupts section in the "Controller Operations" chapter of the Databook.

PCIE_X4_EP_PF0_PORT_LOGIC_MSIX_RAM_CTRL_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_MSIX_RAM_CTRL_OFF_0

Offset: 0x94c
Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:26	RO	0x0	**MSIX_RAM_CTRL_RESERVED_26_31**: Reserved. Note: This register field is sticky.
25	RW	0x0	**MSIX_RAM_CTRL_DBG_PBA**: MSIX PBA RAM Debug Mode. You can also use the dbg_pba input to activate debug mode. Debug mode turns off the PF/VF/Offset-based addressing into the RAM and maps the entire table linearly from the base address of the BAR (indicated by the BIR) in function 0. Note: This register field is sticky.
24	RW	0x0	**MSIX_RAM_CTRL_DBG_TABLE**: MSIX Table RAM Debug Mode. You can also use the dbg_table input to activate debug mode. Debug mode turns off the PF/VF/Offset-based addressing into the RAM and maps the entire table linearly from the base address of the BAR (indicated by the BIR) in function 0. Note: This register field is sticky.
23:17	RO	0x0	**MSIX_RAM_CTRL_RESERVED_17_23**: Reserved. Note: This register field is sticky.
16	RW	0x0	**MSIX_RAM_CTRL_BYPASS**: MSIX RAM Control Bypass. It is up to the application to ensure the RAMs are in the proper power state before trying to access them. Moreover, the application needs to observe all timing requirements of the RAM low power signals before trying to use the MSIX functionality. Note: This register field is sticky.
15:10	RO	0x0	**MSIX_RAM_CTRL_RESERVED_10_15**: Reserved. Note: This register field is sticky.
9	RW	0x0	**MSIX_RAM_CTRL_PBA_SD**: MSIX PBA RAM Shut Down. Note: This register field is sticky.
8	RW	0x0	**MSIX_RAM_CTRL_PBA_DS**: MSIX PBA RAM Deep Sleep. Note: This register field is sticky.
7:2	RO	0x0	**MSIX_RAM_CTRL_RESERVED_2_7**: Reserved. Note: This register field is sticky.
1	RW	0x0	**MSIX_RAM_CTRL_TABLE_SD**: MSIX Table RAM Shut Down. Note: This register field is sticky.
0	RW	0x0	**MSIX_RAM_CTRL_TABLE_DS**: MSIX Table RAM Deep Sleep. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_SAFETY_MASK_OFF_0

where <*j*> = 4, 8.

Description:This registers holds the masks for functional safety interrupt events.

PCIE_X4_EP_PF0_PORT_LOGIC_SAFETY_MASK_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_SAFETY_MASK_OFF_0

Offset: 0x960
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:6	RO	0x0	**RSVDP_6**: Reserved for future use.
5	RW	0x0	**SAFETY_INT_MASK_5**: Mask for functional safety interrupt event 5 (RASDP correctable). Note: This register field is sticky.
4	RW	0x0	**SAFETY_INT_MASK_4**: Mask for functional safety interrupt event 4 (PCIe correctable). Note: This register field is sticky.
3	RW	0x0	**SAFETY_INT_MASK_3**: Mask for functional safety interrupt event 3 (PCIe uncorrectable). Note: This register field is sticky.
2	RW	0x0	**SAFETY_INT_MASK_2**: Mask for functional safety interrupt event 2 (Interface timers). Note: This register field is sticky.
1	RW	0x0	**SAFETY_INT_MASK_1**: Mask for functional safety interrupt event 1 (CDM register checker). Note: This register field is sticky.
0	RW	0x0	**SAFETY_INT_MASK_0**: Mask for functional safety interrupt event 0 (RASDP). Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_SAFETY_STATUS_OFF_0

where <*j*> = 4, 8.

Description:This register provides the functional safety interrupt events status.

PCIE_X4_EP_PF0_PORT_LOGIC_SAFETY_STATUS_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_SAFETY_STATUS_OFF_0

Offset: 0x964
Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:6	RO	0x0	**RSVDP_6**: Reserved for future use.
5	RW	0x0	**SAFETY_INT_STATUS_5**: Status for functional safety interrupt event 5 (RASDP correctable). This register field is sticky.
4	RW	0x0	**SAFETY_INT_STATUS_4**: Status for functional safety interrupt event 4 (PCIe correctable). This register field is sticky.
3	RW	0x0	**SAFETY_INT_STATUS_3**: Status for functional safety interrupt event 3 (PCIe uncorrectable). This register field is sticky.
2	RW	0x0	**SAFETY_INT_STATUS_2**: Status for functional safety interrupt event 2 (Interface timers). This register field is sticky.
1	RW	0x0	**SAFETY_INT_STATUS_1**: Status for functional safety interrupt event 1 (CDM register checker). This register field is sticky.
0	RW	0x0	**SAFETY_INT_STATUS_0**: Status for functional safety interrupt event 0 (RASDP). This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_PL_APP_BUS_DEV_NUM_STATUS_OFF_0

where <*j*> = 4, 8.

Description:This register reflects the application driven bus and device number.

PCIE_X4_EP_PF0_PORT_LOGIC_PL_APP_BUS_DEV_NUM_STATUS_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_PL_APP_BUS_DEV_NUM_STATUS_OFF_0

Offset: 0xb10
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**RSVDP_16**: Reserved for future use.

Bit	Reset	Description
15:8	0x0	**RC_DSW_BUS_NUM**: This field reflects the value of bus number driven on app_bus_num input signal by your application. Note: This register field is sticky.
7:3	0x0	**RC_DSW_DEV_NUM**: This field reflects the value of device number driven on app_device_num input signal by your application. Note: This register field is sticky.
2:0	0x0	**RSVDP_0**: Reserved for future use.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_PCIPM_TRAFFIC_CTRL_OFF_0

where <*j*> = 4, 8.

Description:This register provides control over TLP Traffic during Non-D0 States.

PCIE_X4_EP_PF0_PORT_LOGIC_PCIPM_TRAFFIC_CTRL_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_PCIPM_TRAFFIC_CTRL_OFF_0

Offset: 0xb1c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:8	RO	0x0	**RSVDP_8**: Reserved for future use.
7:4	RO	0x0	**PCIPM_RESERVED_4_7**: Reserved. Note: This register field is sticky.
3	RW	0x0	**PCIPM_NEW_TLP_CLIENT2_BLOCKED**: This field indicates that all TLPs transmitted by Client 2 interface are blocked during non-D0 states. Note: This register field is sticky.
2	RW	0x0	**PCIPM_NEW_TLP_CLIENT1_BLOCKED**: This field indicates that all TLPs transmitted by Client 1 interface are blocked during non-D0 states. Note: This register field is sticky.
1	RW	0x0	**PCIPM_NEW_TLP_CLIENT0_BLOCKED**: This field indicates that all TLPs transmitted by Client 0 interface are blocked during non-D0 states. Note: This register field is sticky.

Bit	R/W	Reset	Description
0	RW	0x0	**PCIPM_VDM_TRAFFIC_BLOCKED**: This field indicates that VDM Message TLPs are blocked during non-D0 states. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_PL_CHK_REG_CONTROL_STATUS_OFF_0

where <*j*> = 4, 8.

Description:Controls register checking and displays status of register checking.

PCIE_X4_EP_PF0_PORT_LOGIC_PL_CHK_REG_CONTROL_STATUS_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_PL_CHK_REG_CONTROL_STATUS_OFF_0

Offset: 0xb20
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31:19	RO	0x0	**RSVDP_19**: Reserved for future use.
18	RW	0x0	**CHK_REG_COMPLETE**: The system has completed a checking cycle.
17	RW	0x0	**CHK_REG_LOGIC_ERROR**: The system has detected an error in its own checking logic.
16	RW	0x0	**CHK_REG_COMPARISON_ERROR**: The system has detected that there is a bit error in the CDM Register Data.
15:2	RO	0x0	**RSVDP_2**: Reserved for future use.
1	RW	0x0	**CHK_REG_CONTINUOUS**: Set Continuous Checking Sequence. Note: This register field is sticky.
0	RW	0x0	**CHK_REG_START**: Begins a checking sequence. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_PL_CHK_REG_START_END_OFF_0

where <*j*> = 4, 8.

Description:This register holds the first and last address to check.

PCIE_X4_EP_PF0_PORT_LOGIC_PL_CHK_REG_START_END_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_PL_CHK_REG_START_END_OFF_0

Offset: 0xb24
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0bff0000 (0b0000,1011,1111,1111,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0xbff	**CHK_REG_END_ADDR**: The last address that is checked by the system. Note: This register field is sticky.
15:0	0x0	**CHK_REG_START_ADDR**: The first address that is checked by the system. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_PL_CHK_REG_ERR_ADDR_OFF_0

where <*j*> = 4, 8.

Description:This register holds the CDM register checking error address.

PCIE_X4_EP_PF0_PORT_LOGIC_PL_CHK_REG_ERR_ADDR_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_PL_CHK_REG_ERR_ADDR_OFF_0

Offset: 0xb28
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**CHK_REG_ERR_ADDR**: The address at which an error has been detected. Valid only when the CDM Register Checker Comparison Error bit is set in the status register. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_PL_CHK_REG_ERR_PF_VF_OFF_0

where <*j*> = 4, 8.

Description:This register holds the CDM Register checking error PF and VF numbers.

PCIE_X4_EP_PF0_PORT_LOGIC_PL_CHK_REG_ERR_ADDR_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_PL_CHK_REG_ERR_PF_VF_OFF_0

Offset: 0xb2c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**RSVDP_28**: Reserved for future use.
27:16	0x0	**CHK_REG_VF_ERR_NUMBER**: The VF number at which the error was detected. Valid only when the CDM Register Checker Comparison Error bit is set in the status register. Note: This register field is sticky.
15:5	0x0	**RSVDP_5**: Reserved for future use.
4:0	0x0	**CHK_REG_PF_ERR_NUMBER**: The PF number at which the error was detected. Valid only when the CDM Register Checker Comparison Error bit is set in the status register. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_PL_LTR_LATENCY_OFF_0

where <*j*> = 4, 8.

Description:The function of this register field (and all other fields in this register) differs between an upstream port and a downstream port. For an upstream port, the register fields capture the corresponding fields in the LTR messages that are transmitted by the port. For a downstream port,

the register fields capture the corresponding fields in the LTR messages that are received by the port. The full content of the register is reflected on the app_ltr_latency[31:0] output.

PCIE_X4_EP_PF0_PORT_LOGIC_PL_LTR_LATENCY_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_PL_LTR_LATENCY_OFF_0

Offset: 0xb30
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**NO_SNOOP_LATENCY_REQUIRE**: No Snoop Latency Requirement. Note: The access attributes of this field are as follows: - Wire: R - Dbi: R/W
30:29	RO	0x0	**RSVDP_29**: Reserved for future use.
28:26	RW	0x0	**NO_SNOOP_LATENCY_SCALE**: No Snoop Latency Scale. Note: The access attributes of this field are as follows: - Wire: R - Dbi: R/W
25:16	RW	0x0	**NO_SNOOP_LATENCY_VALUE**: No Snoop Latency Value. Note: The access attributes of this field are as follows: - Wire: R - Dbi: R/W
15	RW	0x0	**SNOOP_LATENCY_REQUIRE**: Snoop Latency Requirement. Note: The access attributes of this field are as follows: - Wire: R - Dbi: R/W
14:13	RO	0x0	**RSVDP_13**: Reserved for future use.
12:10	RW	0x0	**SNOOP_LATENCY_SCALE**: Snoop Latency Scale. Note: The access attributes of this field are as follows: - Wire: R - Dbi: R/W
9:0	RW	0x0	**SNOOP_LATENCY_VALUE**: Snoop Latency Value. Note: The access attributes of this field are as follows: - Wire: R - Dbi: R/W

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_AUX_CLK_FREQ_OFF_0

where <*j*> = 4, 8.

Description:This register controls the auxiliary clock frequency.

PCIE_X4_EP_PF0_PORT_LOGIC_AUX_CLK_FREQ_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_AUX_CLK_FREQ_OFF_0

Offset: 0xb40
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000013 (0b0000,0000,0000,0000,0000,0000,0001,0011)

Bit	R/W	Reset	Description
31:10	RO	0x0	**RSVDP_10**: Reserved for future use.
9:0	RW	0x13	**AUX_CLK_FREQ**

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_L1_SUBSTATES_OFF_0

where <*j*> = 4, 8.

Description:This register holds L1 substates timing information.

PCIE_X4_EP_PF0_PORT_LOGIC_L1_SUBSTATES_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_L1_SUBSTATES_OFF_0

Offset: 0xb44
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000000d2 (0b0000,0000,0000,0000,0000,0000,1101,0010)

Bit	R/W	Reset	Description
31:9	RO	0x0	**RSVDP_9**: Reserved for future use.
8	RW	0x0	**L1SUB_LOW_POWER_CLOCK_SWITCH_MODE**: If the bit is set to 1'b1 the controller will delay the switching of aux_clk to the slow platform clock until it detects that the link partner has de-asserted CLKREQ#. Note: This register field is sticky.
7:6	RW	0x3	**L1SUB_T_PCLKACK**: Max delay (in 1us units) between a MAC request to remove the clock on mac_phy_pclkreq_n and a PHY response on phy_mac_pclkack_n. If the PHY does not respond within this time the request is aborted. Range is 1..4 Note: This register field is sticky.

Bit	R/W	Reset	Description
5:2	RW	0x4	**L1SUB_T_L1_2**: Duration (in 1us units) of L1.2. Range is 1..16. Note: This register field is sticky.
1:0	RW	0x2	**L1SUB_T_POWER_OFF**: Duration (in 1us units) of L1.2.Entry. Range is 1..4. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_POWERDOWN_CTRL_STATUS_OFF_0

where <*j*> = 4, 8.

Description:This is the Powerdown Control and Status register.

PCIE_X4_EP_PF0_PORT_LOGIC_POWERDOWN_CTRL_STATUS_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_POWERDOWN_CTRL_STATUS_OFF_0

Offset: 0xb48
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000220 (0b0000,0000,0000,0000,0000,0010,0010,0000)

Bit	R/W	Reset	Description
31:12	RO	0x0	**RSVDP_12**: Reserved for future use.
11:8	RO	0x2	**POWERDOWN_PHY_POWERDOWN**: This field represents the Powerdown value that has been acknowledged by the PHY. It is updated with the value of Powerdown driven by the controller, when the PHY has returned the Phystatus acknowledgment for the Powerdown transition.
7:4	RO	0x2	**POWERDOWN_MAC_POWERDOWN**: This field represents the Powerdown value driven by the controller to the PHY.
3:2	RO	0x0	**RSVDP_2**: Reserved for future use.
1	RW	0x0	**POWERDOWN_VMAIN_ACK**: Set this bit to 1 if you do not want to perform the handshake with the power-switch after PERST# assertion. By default the controller will perform the handshake with the power-switch if L1 power gating is enabled Note: This register field is sticky.

Bit	R/W	Reset	Description
0	WO	0x0	**POWERDOWN_FORCE**: This field is a one shot field. This field could be used for debug purposes in event that the P2 Powerdown transition does not complete. It will allow the controller to proceed with the transition to the P1 Powerdown state. This field always reads back as 1'b0.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_GEN4_LANE_MARGINING_1_OFF_0

where <*j*> = 4, 8.

Description:This is the Gen4 Lane Margining 1 Register.

PCIE_X4_EP_PF0_PORT_LOGIC_GEN4_LANE_MARGINING_1_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_GEN4_LANE_MARGINING_1_OFF_0

Offset: 0xb80
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x05201407 (0b0000,0101,0010,0000,0001,0100,0000,0111)

Bit	R/W	Reset	Description
31:30	RO	0x0	**RSVDP_30**: Reserved for future use.
29:24	RW	0x5	**MARGINING_MAX_VOLTAGE_OFFSET**: M(MaxVoltageOffset) for Lane Margining at the Receiver. This parameter is defined in "Electrical Sub-Block" section of the PCI Express Base Specification. Note: This register field is sticky.
23	RO	0x0	**RSVDP_23**: Reserved for future use.
22:16	RW	0x20	**MARGINING_NUM_VOLTAGE_STEPS**: M(NumVoltageSteps) for Lane Margining at the Receiver. This parameter is defined in "Electrical Sub-Block" section of the PCI Express Base Specification. Note: This register field is sticky.
15:14	RO	0x0	**RSVDP_14**: Reserved for future use.
13:8	RW	0x14	**MARGINING_MAX_TIMING_OFFSET**: M(MaxTimingOffset) for Lane Margining at the Receiver. This parameter is defined in "Electrical Sub-Block" section of the PCI Express Base Specification. Note: This register field is sticky.
7:6	RO	0x0	**RSVDP_6**: Reserved for future use.

Bit	R/W	Reset	Description
5:0	RW	0x7	**MARGINING_NUM_TIMING_STEPS**: M(NumTimingSteps) for Lane Margining at the Receiver. This parameter is defined in "Electrical Sub-Block" section of the PCI Express Base Specification. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_GEN4_LANE_MARGINING_2_OFF_0

where <*j*> = 4, 8.

Description:This is the Gen4 Lane Margining 2 Register.

PCIE_X4_EP_PF0_PORT_LOGIC_GEN4_LANE_MARGINING_2_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_GEN4_LANE_MARGINING_2_OFF_0

Offset: 0xb84
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x180f0f0f (0bx001,1000,0000,1111,0000,1111,0000,1111)

Bit	R/W	Reset	Description
30:29	RO	0x0	**RSVDP_29**: Reserved for future use.
28	RW	0x1	**MARGINING_IND_ERROR_SAMPLER**: M(IndErrorSampler) for Lane Margining at the Receiver. This parameter is defined in "Electrical Sub-Block" section of the PCI Express Base Specification. Note: This register field is sticky.
27	RW	0x1	**MARGINING_SAMPLE_REPORTING_METHOD**: M(SampleReportingMethod) for Lane Margining at the Receiver. This parameter is defined in "Electrical Sub-Block" section of the PCI Express Base Specification. Note: This register field is sticky.
26	RW	0x0	**MARGINING_IND_LEFT_RIGHT_TIMING**: M(IndLeftRightTiming) for Lane Margining at the Receiver. This parameter is defined in "Electrical Sub-Block" section of the PCI Express Base Specification. Note: This register field is sticky.
25	RW	0x0	**MARGINING_IND_UP_DOWN_VOLTAGE**: M(IndUpDownVoltage) for Lane Margining at the Receiver. This parameter is defined in "Electrical Sub-Block" section of the PCI Express Base Specification. Note: This register field is sticky.
24	RW	0x0	**MARGINING_VOLTAGE_SUPPORTED**: M(VoltageSupported) for Lane Margining at the Receiver. This parameter is defined in "Electrical Sub-Block" section of the PCI Express Base Specification. Note: This register field is sticky.

Bit	R/W	Reset	Description
23:21	RO	0x0	**RSVDP_21**: Reserved for future use.
20:16	RW	0xf	**MARGINING_MAXLANES**: M(MaxLanes) for Lane Margining at the Receiver. This parameter is defined in "Electrical Sub-Block" section of the PCI Express Base Specification. Note: This register field is sticky.
15:14	RO	0x0	**RSVDP_14**: Reserved for future use.
13:8	RW	0xf	**MARGINING_SAMPLE_RATE_TIMING**: M(SamplingRateTiming) for Lane Margining at the Receiver. This parameter is defined in "Electrical Sub-Block" section of the PCI Express Base Specification. Note: This value is not used when MARGINING_IND_ERROR_SAMPLER is 0b. The M(SamplingRateTiming) is fixed to 63 internally. Note: This register field is sticky.
7:6	RO	0x0	**RSVDP_6**: Reserved for future use.
5:0	RW	0xf	**MARGINING_SAMPLE_RATE_VOLTAGE**: M(SamplingRateVoltage) for Lane Margining at the Receiver. This parameter is defined in "Electrical Sub-Block" section of the PCI Express Base Specification. Note: This value is not used when MARGINING_IND_ERROR_SAMPLER is 0b. The M(SamplingRateVoltage) is fixed to 63 internally. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_PORT_LOGIC_PIPE_RELATED_OFF_0

where <*j*> = 4, 8.

Description:This register controls the PIPE's capabitity, control, and status parameters.

PCIE_X4_EP_PF0_PORT_LOGIC_PIPE_RELATED_OFF_0

PCIE_X8_EP_PF0_PORT_LOGIC_PIPE_RELATED_OFF_0

Offset: 0xb90
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,xxxx,xxxx)

Bit	R/W	Reset	Description
31:9	RO	0x0	**RSVDP_9**: Reserved for future use.

Bit	R/W	Reset	Description
8	RW	0x0	**PIPE_GARBAGE_DATA_MODE**: PIPE Garbage Data Mode. - RxValid is deasserted - a valid RxStartBlock is received at 128b/130b encoding - a valid COM symbol is received at 8b/10b encoding Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_TYPE0_HDR_DBI2_BAR0_MASK_REG_0

where <*j*> = 4, 8.

Description:This register is the mask for BAR0_REG. If implemented, it exists as a shadow register at the BAR0_REG address. Normally, the BAR masks are used for indicating the amount of address space that each BAR requests from host software. The BAR masks determine which bits in each BAR are non-writable by host software, which determines the size of the address space claimed by each BAR. The BAR mask values indicate the range of low-order bits, in each implemented BAR, not to use for address matching. The BAR mask value also indicates the range of low-order bits in the BAR that cannot be written from the host. The application can write to all BAR bits to allow setting of memory, I/O, and other standard BAR options. Your local CPU can change the mask at runtime using the DBI. The mask register is invisible to the PCIe wire but visible to your local CPU through the DBI. You cannot read the mask register but you can write to it. It is accessed by asserting dbi_cs2 and dbi_cs. If you only assert dbi_cs then you will access the BAR which is the primary register at that location. Use CS2 instead of dbi_cs2 when you are using the AHB/AXI bridge.

PCIE_X4_EP_PF0_TYPE0_HDR_DBI2_BAR0_MASK_REG_0

PCIE_X8_EP_PF0_TYPE0_HDR_DBI2_BAR0_MASK_REG_0

Offset: 0x1010
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:1	0x7fffffff	**PCI_TYPE0_BAR0_MASK**: BAR0 Mask. Note: The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: W (sticky) Note: This register field is sticky.
0	0x1	**PCI_TYPE0_BAR0_ENABLED**: BAR0 Mask Enabled. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_TYPE0_HDR_DBI2_BAR1_MASK_REG_0

where <*j*> = 4, 8.

Description:This register is the mask for BAR1_REG. If implemented, it exists as a shadow register at the BAR1_REG address. Normally, the BAR masks are used for indicating the amount of address space that each BAR requests from host software. The BAR masks determine which bits in each BAR are non-writable by host software, which determines the size of the address space claimed by each BAR. The BAR mask values indicate the range of low-order bits, in each implemented BAR, not to use for address matching. The BAR mask value also indicates the range of low-order bits in the BAR that cannot be written from the host. The application can write to all BAR bits to allow setting of memory, I/O, and other standard BAR options. Your local CPU can change the mask at runtime using the DBI. The mask register is invisible to the PCIe wire but visible to your local CPU through the DBI. You cannot read the mask register but you can write to it. It is accessed by asserting dbi_cs2 and dbi_cs. If you only assert dbi_cs then you will access the BAR which is the primary register at that location. Use CS2 instead of dbi_cs2 when you are using the AHB/AXI bridge.

PCIE_X4_EP_PF0_TYPE0_HDR_DBI2_BAR1_MASK_REG_0

PCIE_X8_EP_PF0_TYPE0_HDR_DBI2_BAR1_MASK_REG_0

Offset: 0x1014
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xfffffffe (0b1111,1111,1111,1111,1111,1111,1111,1110)

Bit	Reset	Description
31:1	0x7fffffff	**PCI_TYPE0_BAR1_MASK**: BAR1 Mask. Note: The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: W (sticky) Note: This register field is sticky.
0	0x0	**PCI_TYPE0_BAR1_ENABLED**: BAR1 Mask Enabled. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_TYPE0_HDR_DBI2_BAR2_MASK_REG_0

where <*j*> = 4, 8.

Description:This register is the mask for BAR2_REG. If implemented, it exists as a shadow register at the BAR2_REG address. Normally, the BAR masks are used for indicating the amount of address space that each BAR requests from host software. The BAR masks determine which bits in each BAR are non-writable by host software, which determines the size of the address space claimed by each BAR. The BAR mask values indicate the range of low-order bits, in each implemented BAR, not to use for address matching. The BAR mask value also indicates the range of low-order bits in the BAR that cannot be written from the host. The application can write to all BAR bits to allow setting

of memory, I/O, and other standard BAR options. Your local CPU can change the mask at runtime using the DBI. The mask register is invisible to the PCIe wire but visible to your local CPU through the DBI. You cannot read the mask register but you can write to it. It is accessed by asserting dbi_cs2 and dbi_cs. If you only assert dbi_cs then you will access the BAR which is the primary register at that location. Use CS2 instead of dbi_cs2 when you are using the AHB/AXI bridge.

PCIE_X4_EP_PF0_TYPE0_HDR_DBI2_BAR2_MASK_REG_0

PCIE_X8_EP_PF0_TYPE0_HDR_DBI2_BAR4_MASK_REG_0

Offset: 0x1018
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:1	0x7fffffff	**PCI_TYPE0_BAR2_MASK**: BAR2 Mask. Note: The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: W (sticky) Note: This register field is sticky.
0	0x1	**PCI_TYPE0_BAR2_ENABLED**: BAR2 Mask Enabled. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_TYPE0_HDR_DBI2_BAR3_MASK_REG_0

where <*j*> = 4, 8.

Description:This register is the mask for BAR3_REG. If implemented, it exists as a shadow register at the BAR3_REG address. Normally, the BAR masks are used for indicating the amount of address space that each BAR requests from host software. The BAR masks determine which bits in each BAR are non-writable by host software, which determines the size of the address space claimed by each BAR. The BAR mask values indicate the range of low-order bits, in each implemented BAR, not to use for address matching. The BAR mask value also indicates the range of low-order bits in the BAR that cannot be written from the host. The application can write to all BAR bits to allow setting of memory, I/O, and other standard BAR options. Your local CPU can change the mask at runtime using the DBI. The mask register is invisible to the PCIe wire but visible to your local CPU through the DBI. You cannot read the mask register but you can write to it. It is accessed by asserting dbi_cs2 and dbi_cs. If you only assert dbi_cs then you will access the BAR which is the primary register at that location. Use CS2 instead of dbi_cs2 when you are using the AHB/AXI bridge.

PCIE_X4_EP_PF0_TYPE0_HDR_DBI2_BAR3_MASK_REG_0

PCIE_X8_EP_PF0_TYPE0_HDR_DBI2_BAR3_MASK_REG_0

Offset: 0x101c
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xfffffffe (0b1111,1111,1111,1111,1111,1111,1111,1110)

Bit	Reset	Description
31:1	0x7fffffff	**PCI_TYPE0_BAR3_MASK**: BAR3 Mask. Note: The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: W (sticky) Note: This register field is sticky.
0	0x0	**PCI_TYPE0_BAR3_ENABLED**: BAR3 Mask Enabled. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_TYPE0_HDR_DBI2_BAR4_MASK_REG_0

where <*j*> = 4, 8.

Description:This register is the mask for BAR4_REG. If implemented, it exists as a shadow register at the BAR4_REG address. Normally, the BAR masks are used for indicating the amount of address space that each BAR requests from host software. The BAR masks determine which bits in each BAR are non-writable by host software, which determines the size of the address space claimed by each BAR. The BAR mask values indicate the range of low-order bits, in each implemented BAR, not to use for address matching. The BAR mask value also indicates the range of low-order bits in the BAR that cannot be written from the host. The application can write to all BAR bits to allow setting of memory, I/O, and other standard BAR options. Your local CPU can change the mask at runtime using the DBI. The mask register is invisible to the PCIe wire but visible to your local CPU through the DBI. You cannot read the mask register but you can write to it. It is accessed by asserting dbi_cs2 and dbi_cs. If you only assert dbi_cs then you will access the BAR which is the primary register at that location. Use CS2 instead of dbi_cs2 when you are using the AHB/AXI bridge.

PCIE_X4_EP_PF0_TYPE0_HDR_DBI2_BAR4_MASK_REG_0

PCIE_X8_EP_PF0_TYPE0_HDR_DBI2_BAR4_MASK_REG_0

Offset: 0x1020
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffffffff (0b1111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31:1	0x7fffffff	**PCI_TYPE0_BAR4_MASK**: BAR4 Mask. Note: The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: W (sticky) Note: This register field is sticky.
0	0x1	**PCI_TYPE0_BAR4_ENABLED**: BAR4 Mask Enabled. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_TYPE0_HDR_DBI2_BAR5_MASK_REG_0

where <*j*> = 4, 8.

Description:This register is the mask for BAR5_REG. If implemented, it exists as a shadow register at the BAR5_REG address. Normally, the BAR masks are used for indicating the amount of address space that each BAR requests from host software. The BAR masks determine which bits in each BAR are non-writable by host software, which determines the size of the address space claimed by each BAR. The BAR mask values indicate the range of low-order bits, in each implemented BAR, not to use for address matching. The BAR mask value also indicates the range of low-order bits in the BAR that cannot be written from the host. The application can write to all BAR bits to allow setting of memory, I/O, and other standard BAR options. Your local CPU can change the mask at runtime using the DBI. The mask register is invisible to the PCIe wire but visible to your local CPU through the DBI. You cannot read the mask register but you can write to it. It is accessed by asserting dbi_cs2 and dbi_cs. If you only assert dbi_cs then you will access the BAR which is the primary register at that location. Use CS2 instead of dbi_cs2 when you are using the AHB/AXI bridge.

PCIE_X4_EP_PF0_TYPE0_HDR_DBI2_BAR5_MASK_REG_0

PCIE_X8_EP_PF0_TYPE0_HDR_DBI2_BAR5_MASK_REG_0

Offset: 0x1024
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xfffffffe (0b1111,1111,1111,1111,1111,1111,1111,1110)

Bit	Reset	Description
31:1	0x7fffffff	**PCI_TYPE0_BAR5_MASK**: BAR5 Mask. Note: The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: W (sticky) Note: This register field is sticky.
0	0x0	**PCI_TYPE0_BAR5_ENABLED**: BAR5 Mask Enabled. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_TYPE0_HDR_DBI2_EXP_ROM_BAR_MASK_REG_0

where <*j*> = 4, 8.

Description:This register is the mask for EXP_ROM_BASE_ADDR_REG register. If implemented, it exists as a shadow register at EXP_ROM_BAR_MASK_REG address. The assertion of CS2 (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) is required to write to this register.

PCIE_X4_EP_PF0_TYPE0_HDR_DBI2_EXP_ROM_BAR_MASK_REG_0

PCIE_X8_EP_PF0_TYPE0_HDR_DBI2_EXP_ROM_BAR_MASK_REG_0

Offset: 0x1030
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0001fffe (0b0000,0000,0000,0001,1111,1111,1111,1110)

Bit	Reset	Description
31:1	0xffff	**ROM_MASK**: Expansion ROM Mask. Note: The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: if ROM_BAR_ENABLED && ROM_MASK_WRITABLE then W Note: This register field is sticky.
0	0x0	**ROM_BAR_ENABLED**: Expansion ROM Bar Mask Register Enabled. Note: The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: if ROM_MASK_WRITABLE then W Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_MSIX_CAP_DBI2_SHADOW_PCI_MSIX_CAP_ID_NEXT_CTRL_REG_0

where <*j*> = 4, 8.

Description:This is the shadow register of the MSI-X Capability ID, Next Pointer, and Control Register.

PCIE_X4_EP_PF0_MSIX_CAP_DBI2_SHADOW_PCI_MSIX_CAP_ID_NEXT_CTRL_REG_0

PCIE_X8_EP_PF0_MSIX_CAP_DBI2_SHADOW_PCI_MSIX_CAP_ID_NEXT_CTRL_REG_0

Offset: 0x10b0
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00070000 (0b0000,0000,0000,0111,0000,0000,0000,0000)

Bit	Reset	Description
31:27	0x0	**PCI_MSIX_RESERVED1**: reserved field 1 in the shadow register. Note: This register field is sticky.
26:16	0x7	**PCI_MSIX_TABLE_SIZE**: MSI-X Table Size in the shadow register. Note: The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
15:0	0x0	**PCI_MSIX_RESERVED0**: reserved field 0 in the shadow register. Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_MSIX_CAP_DBI2_SHADOW_MSIX_TABLE_OFFSET_REG_0

where <*j*> = 4, 8.

Description:This register provides Table BIR and MSI-x Table offset select.

PCIE_X4_EP_PF0_MSIX_CAP_DBI2_SHADOW_MSIX_TABLE_OFFSET_REG_0

PCIE_X8_EP_PF0_MSIX_CAP_DBI2_SHADOW_MSIX_TABLE_OFFSET_REG_0

Offset: 0x10b4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000002 (0b0000,0000,0000,0000,0000,0000,0000,0010)

Bit	Reset	Description
31:3	0x0	**PCI_MSIX_TABLE_OFFSET**: MSI-X Table Offset. Used as an offset from the address contained by one of the Function's Base Address Registers to point to the base of the MSI-X Table. The lower 3 Table BIR bits are masked off (set to zero) by software to form a 32-bit QWORD-aligned offset. Note: The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: if (DBI_RO_WR_EN == 1 && MSIX_CAP_ENABLE=1) then R/W else R Note: This register field is sticky.

Bit	Reset	Description
2:0	0x2	**PCI_MSIX_BIR**: MSI-X Table BAR Indicator Register Field. This field indicates which one of a Function's Base Address Registers , located beginning at 10h in Configuration Space, or entry in the Enhanced Allocation capability with a matching BEI , is used to map the Function's MSI-X Table into Memory Space. All encodings other than the defined encodings are reserved. For a 64-bit Base Address Register , the Table BIR indicates the lower DWORD. For Functions with Type 1 Configuration Space headers, BIR values 2 through 5 are also Reserved. Note: The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: if (DBI_RO_WR_EN == 1 && MSIX_CAP_ENABLE=1) then R/W else R Note: This register field is sticky.

PCIE_X<*j*>_EP_PF0_MSIX_CAP_DBI2_SHADOW_MSIX_PBA_OFFSET_REG_0

where <*j*> = 4, 8.

Description:This register provides PBA Offset and PBA BIR value.

PCIE_X4_EP_PF0_MSIX_CAP_DBI2_SHADOW_MSIX_PBA_OFFSET_REG_0

PCIE_X8_EP_PF0_MSIX_CAP_DBI2_SHADOW_MSIX_PBA_OFFSET_REG_0

Offset: 0x10b8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00010002 (0b0000,0000,0000,0001,0000,0000,0000,0010)

Bit	Reset	Description
31:3	0x2000	**PCI_MSIX_PBA_OFFSET**: MSI-X PBA Offset. Used as an offset from the address contained by one of the Function's Base Address Registers to point to the base of the MSI-X PBA. The lower 3 PBA BIR bits are masked off (set to zero) by software to form a 32-bit QWORD-aligned offset. Note: The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: if (DBI_RO_WR_EN == 1 && MSIX_CAP_ENABLE=1) then R/W else R Note: This register field is sticky.
2:0	0x2	**PCI_MSIX_PBA_BIR**: MSI-X PBA BIR. This field indicates which one of a Function's Base Address Registers , located beginning at 10h in Configuration Space, or entry in the Enhanced Allocation capability with a matching BEI, is used to map the Function's MSI-X PBA into Memory Space. The PBA BIR value definitions are identical to those for the Table BIR . Note: The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: if (DBI_RO_WR_EN == 1 && MSIX_CAP_ENABLE=1) then R/W else R Note: This register field is sticky.

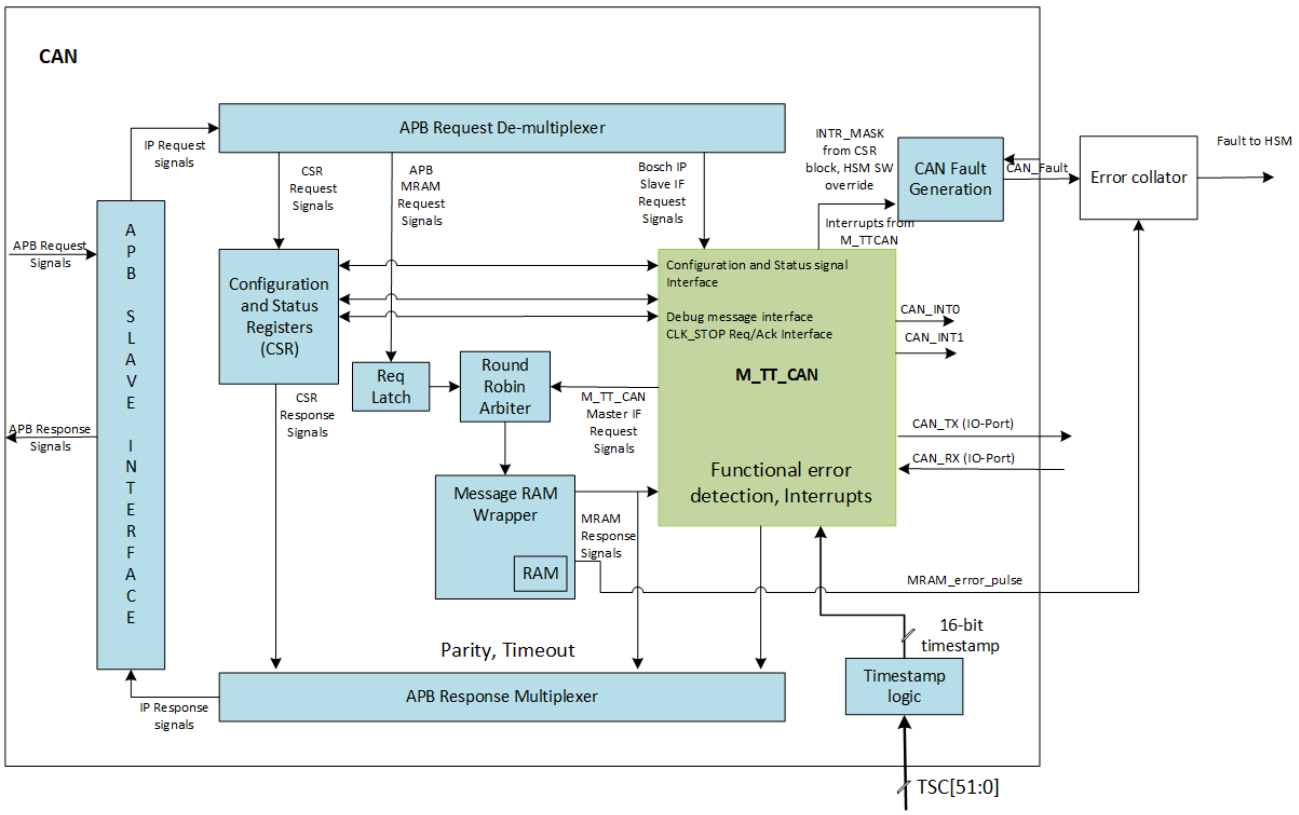
9.4 Controller Area Network (CAN)

9.4.1 Overview

The NVIDIA® Orin™ series System-on-Chip (SoC) integrates two time-triggered Controller Area Network (CAN) controllers to support connectivity to two CAN networks at the system level, with each CAN controller acting as an independent device.

The CAN transceivers themselves are external devices, and Orin supports a variety of CAN transceivers.

Figure 9.16 CAN Controller Block Diagram



In the figure above:

- M_TT_CAN above is the CAN controller .
- APB Slave interface communicates with the back-bone APB bus.
- APB Request De-Multiplexer re-directs the incoming APB request to Configuration and Status Register (CSR) block. The APB Response Multiplexer aggregates the responses from various sub-blocks into the APB Response.

- CAN controller has a separate Message RAM to store incoming/outgoing Messages, Tx Messages timestamp, filters to be applied on incoming Messages. Message RAM size is 4KB.
- The configuration and status registers block provide a register interface for software to control the auxiliary interfaces provided by CAN core. There are three auxiliary interfaces provided by CAN Core, these interfaces aggregate signals required for Clock calibration unit, Debug Message unit, coupling two Time Triggered networks and other status signals.
- A Round Robin arbiter arbitrates between Message RAM access requests from the M_TT_CAN Master interface and the APB interface.
- Req Latch is responsible for latching the APB Read/Write requests from the APB interface towards Message RAM.
- CAN Fault Generation generates the Fault detect signal from CAN. This is derived from the IR(interrupt register) and TTIR(time triggered interrupt register) buses provided at M_TT_CAN interface. These are a total of 51 status registers. In addition, there are 51 mask bits provided to make a status Fault or non-Fault. There is additional software configurable register which can set this fault even when there is no interrupt. For AON CAN instances, the fault is connected via AON error collator to SCE_HSM.
- Timestamp logic provides the external time stamp for CAN controller which is required in CAN FD mode. Internal time stamp is not used in CAN FD mode with Bit Rate Switch. 16-bit port is available from M_TT_CAN to hook-up the external time stamp counter. To support external timestamping, CAN gets an input TSC in the AON and FSI cluster.

9.4.1.1 References

- CAN 2.0 Specification
- CAN FD Specification

9.4.1.2 Hardware Features

- Compliant with ISO 11898 and tested according to ISO/DIS 16845 (CAN conformance test)
  - Includes ISO 11898-4, time triggered communication
  - Includes ISO/CD 11898-1 CAN FD Frame formats
- Two Independent CAN ports/channels
- Standard frame and extended frame transmission/reception enable
- CAN controller instances within AON cluster can support serial communication data transfer up-to 15 Mbps.
- 0 – 8 byte data length, with the ability to receive the first 8 bytes when Data length coding is > 8 Bytes
- 32 message buffers per channel
- Prioritization of transmit buffers
- Receive/transmit history list function

- Automatic block transmission function
- Multi-buffer receive block function
- Flexible maskable identifier filter supports of two 32-bit, or four 16-bit, or eight 8-bit filters for each channel
- Programmable data bit time, communication baud rate, and sample point
  - As an example, the following sample-point configurations can be configured: 66.7%, 70.0%, 75.0%, 80.0%, 81.3%, 85.0%, 87.5%
  - Baud rates in the range of 10 kbps up to 1000 kbps can be configured
- Enhanced features:
  - Each message buffer can be configured to operate as a transmit or a receive message buffer
  - Transmission priority is controlled by the identifier or by mailbox number (selectable)
  - A transmission request can be aborted by clearing the dedicated Transmit-Request flag of the concerned message buffer
  - Automatic block transmission (ABT) operation mode
  - Time stamp function for CAN channels 0 to "n" in collaboration with timers
- Release from bus-off state by software.
- Wake-Up with integrated low-pass filter (debounce) option to prevent short glitches on CAN bus, through CAN Rx signal toggling from CAN transceiver
  - For normal operation (after wake) there is a digital filter in the CAN controller
- Listen-Only mode to monitor CAN bus
- Loop Back for self-test
- Supports selectable 3.3V or 1.8V I/O statically at board design

9.4.1.3 Software Features

- CAN 2.0 A, B Software Compatible
- Sustain average interrupt of around 2000 messages/s on average of 500 µs/message, with a peak of 125 µs/message (8000 messages/s @ 1 Mbps)
- CAN Software Isolation from corruption from host side software, continue to execute CAN program when user, host side software is hung, crashed, or inoperable
- Supports CAN Firmware Update and upgrade securely
- Supports AUTOSAR, MCAL communication CAN Driver API
- Supports Android, Linux, QNX, Windows, etc.
- Supports virtualization, concurrent multi-OS

9.4.2 Functional Description

The CAN Controller implements CAN 2.0 and CAN FD specifications. CAN controller instances within AON cluster can support serial communication data transfer up-to 15 Mbps.

9.4.2.1 CAN Clock Domains

9.4.2.1.1 Clocking Policy for AON

CAN clock requirements are determined by CAN Interface timing requirements. To achieve the required bit timing on the CAN interface we can make use of several dividers present in the CAN IP.

These dividers are:

1. BTP – It is the programmable ratio between time_quantum (tq) and CAN Core Clock period. It can be programmed from 1 to 32
2. nTQ – Number of time Quanta per bit. It is programmable from 4 to 25.

So, core clock frequency is given by CCLK_Freq = BTP x nTQ x CAN Interface Data Rate. Further CCLK_Freq should be less than 80 MHz. For HCLK_Freq, we can program HCLK_Freq = {1, 2, 3, 4} x CCLK_Freq. Finally, AON PLL frequency should derive an HCLK_Freq that satisfies above conditions.

Additionally:

1. Keep HCLK_Freq as high as possible to reduce the latency on APB Bus
2. Keep CCLK_Freq as high as possible (still less than 80 MHz), to allow larger nTQ, which allows functionality on larger networks at higher frequencies.

The following table has the frequency ranges for CAN.

Clock	Minimum	Maximum
CCLK	8 MHz	80 MHz
HCLK (always N*CCLK, N=1,2,3,4)	8 MHz	320 MHz
PCLK (System Dependent)	N/A	N/A

For an AON clocking policy at CAN HCLK = 200 MHz and CCLK = 50 MHz, the valid BTP and nTQ values are as follows.

S.No.	Data rate	BTP x nTQ	Valid BTP	Valid nTQ
1	125Kbps	= 50/0.125 = 400	16,20,25	25,20,16
2	250Kbps	= 50/0.25 = 200	8,10,20,25	25,20,10,8

S.No.	Data rate	BTP x nTQ	Valid BTP	Valid nTQ
3	500Kbps	= 50/0.5 = 100	4,5,10,20,25	25,20,10,5,4
4	1Mbps	= 50/1 = 50	2,5,10	25,10,5
5	2Mbps	= 50/2 = 25	1,5	25,5
6	10Mbps	= 50/10 = 5	1	5

To support 15 Mbps throughput, the valid BTP and nTQ combinations at HCLK = 240 MHz and CCLK = 60 MHz are as follows.

S.No.	Data rate	BTP x nTQ	Valid BTP	Valid nTQ
1	125Kbps	= 60/0.125 = 480	20,24,30,32	24,20,16,15
2	250Kbps	= 60/0.25 = 240	10,12,15,16,20,24	24,20,16,15,12,10
3	500Kbps	= 60/0.5 = 120	5,6,8,10,12,15,20,24,30	24,20,15,12,10,8,6,5,4
4	1Mbps	= 60/1 = 60	3,4,5,6,10,12,15	20,15,12,10,6,5,4
5	2Mbps	= 60/2 = 30	2,3,5,6	15,10,6,5
6	10Mbps	= 60/10 = 6	1	6
7	15Mbps	=60/15 = 4	1	4

Refer to the data sheet for the Orin product being used to determine the maximum data rates supported. Not all rates shown here are applicable to all products.

9.4.2.2 CAN Resets

Each CAN controller gets a signal reset from SoC. This reset is sync'ed to CAN clocks before it drives logic in a particular CAN Clock domain. Reset for PCLK domain logic is sync'ed to PCLK.

9.4.2.3 External Time Stamp

The external time stamp is required in the CAN FD mode. The internal time stamp is not used in the CAN FD mode with the bit-rate switch. A 16-bit port is available from the M_TT_CAN to hook up the external time stamp counter. The TSC terminal node is available in the AON cluster PG partition. As per the driver requirements, a 1 µs to 10 µs granularity is required for time stamping. The terminal node operates under the TSC reference clock which is 31.25 MHz. With this frequency, the terminal node counter LSB gives a 32 ns granularity. By considering tsc [20:5], the result is a 1.024 µs granularity with a wraparound of 67 ms. Since the granularity requirement

varies, a way to select the required offset is needed. A 5-bit MUX select is provided which selects the different offsets of the TSC counter which gives the required granularity:

granularity = 1.024 µs * 2 ** N Where N = 0, 1, 2..15

The OFFSET_SEL field in the M_TTCAN_TIME_STAMP_0 register is used to select the required offset. Note that the value should not be changed dynamically when CAN is in operation.

For external time stamp mode:

1. TSCV register accurately reflects the external time input.
2. Rx time stamps (i.e., RXTS) reflects the external time input for normal and FD modes.
3. Tx event timestamps (i.e., TXTS) reflects the external time input for normal and FD modes.
4. Timeout counter (TOCC) works correctly in FD mode.
5. Time stamp wraparound is generated.

9.4.2.4 CAN Message RAM

Each CAN controller has a separate 4KB Message RAM to store incoming/outgoing Messages, TX Messages timestamp, and Filters to be applied on incoming Messages.

> The generic master interface and CAN APB slave interface are both synchronous to aon_apb_clk. The Message RAM Wrapper module can pipeline accesses to Message RAM. Message RAM is parity protected at byte boundaries to comply with safety requirements.

9.4.2.5 Round Robin Arbiter Between APB and CAN Core

Since both APB and CAN Core can access the Message RAM. An arbiter is required between the write/read requests coming from the two masters.

The arbiter used is a simple round-robin arbiter. A few points about the arbiter:

- It takes the arbiter three HCLKs to service a request.
- When both masters place the request at the same time, the grant is given to the master who didn't get grant the last time
- When a master requests access while a request is already in process, the request is stalled until the ongoing request is completed.

- The arbitration happens between requests of the same kind only (i.e., between reads or between writes and never between a read and a write). In case of read request from one master and write request from another master, write request is given preference.

9.4.2.6 Configuration and Status Registers

There are three auxiliary interfaces provided by TT_CAN Core, these interfaces aggregate signals required for Clock calibration unit, Debug Message unit, coupling two Time-Triggered networks, and some other status signals. These signals are provided as a register interface in case software wants to use them.

Table 9.4 Status Signals

Signal Name	Direction	Clock Domain	Comment
dis_mord	IN	HCLK	Required for Production testing. Software programmable
Swt	IN	ASYNC	Required for Synchronization of two TT CAN Networks. This is the software override for SWT signal, it can be otherwise driven from SOC, TMP, or RTP signals coming from another CAN controller
Evt	IN	ASYNC	Required for Synchronization of two TT CAN Networks. This is the software override for EVT signal, it can be otherwise driven from SOC, TMP, or RTP signals coming from another CAN controller
Soc	OUT	HCLK	Required for Synchronization of two TT CAN Networks. This signal is also provided as the output of CAN controller so that it can be used by another CAN controller to drive its SWT and EVT pins
Tmp	OUT	CCLK	Required for Synchronization of two TT CAN Networks. This signal is also provided as the output of CAN controller so that it can be used by another CAN controller to drive its SWT and EVT pins
Rtp	OUT	CCLK	Required for Synchronization of two TT CAN Networks. This signal is also provided as the output of CAN controller so that it can be used by another CAN controller to drive its SWT and EVT pins
Ascm	OUT	HCLK	Required for Asynchronous Serial Communication. ASC not implemented
Asct	OUT	HCLK	Required for Asynchronous Serial Communication. ASC not implemented
dma_ack	IN	HCLK	Required for DMA Message Unit
dma_req	OUT	HCLK	Required for DMA Message Unit

Signal Name	Direction	Clock Domain	Comment
Cok	IN	HCLK	Required for Clock Calibration Unit. CCU not implemented
ir[31:0]	OUT	HCLK	Though this bus is not used for driving interrupt controllers it is used for generating faults to HSM.
ttir[18:0]	OUT	HCLK	Though this bus is not used for driving interrupt controllers it is used for generating faults to HSM.
txbrp[31:0]	OUT	HCLK	Not required.
Rxfd	OUT	CCLK	Not required.
Txfd	OUT	CCLK	Not required.
fe[2:0]	OUT	HCLK	Required for Clock Calibration Unit. CCU not implemented
Cce	OUT	HCLK	Required for Clock Calibration Unit. CCU not implemented
Spt	OUT	CCLK	Required for Clock Calibration Unit. CCU not implemented
Mrx	OUT	CCLK	Required for Clock Calibration Unit. CCU not implemented
Calf	OUT	CCLK	Required for Clock Calibration Unit. CCU not implemented
Aff	OUT	HCLK	Required for Clock Calibration Unit. CCU not implemented

9.4.2.7 Time-Triggered (TT) CAN

9.4.2.7.1 TT Related Features Supported

- TTCAN Level 0, 1, and 2
- Time Mark Interrupts
- Stop Watch
- Watchdog Timer
- Synchronization to external events

9.4.2.7.2 TT Interface Between the Two CAN Controllers

In CAN based networks (with two CAN controllers and each CAN controller expected to sit on two different Networks), a node such as an NVIDIA SoC can act as a potential gateway between the two networks. In addition to forwarding messages between the two networks, in TT Networks a gateway can compare and align cycle time, global time, and even local time of the two networks

(e.g., a Time Master of both networks resided inside the NVIDIA SoC and needed to sync with the other network).

To enable this functionality m_ttcan_rtp (RTP – Register TM Interrupt Pulse), m_ttcan_tmp (TMP – Trigger TM Interrupt Pulse), and m_ttcan_soc (SoC – Start of Cycle) outputs of one CAN controller are connected to another CAN controller. These signals are connected to m_ttcan_evt and m_ttcan_swt inputs. For both m_ttcan_evt (EVT – Event Trigger) and m_ttcan_swt (SWT – Stop Watch Trigger) there are four connectivity options – m_ttcan_rtp, m_ttcan_tmp, and m_ttcan_soc outputs from the other CAN controller and a configuration and status register (CSR) field based signal from the same CAN controller.

9.4.2.8 System Interfaces

- CAN uses the standard APB interface to connect with the Control Fabric.
- CAN Rx and CAN Tx signals connect with the CAN PHY outside of the chip.
- Error pulse for message RAM parity error indication to AONError collator in the respective cluster.
- Fault signal for connectivity with HSM via AONError collator in the respective cluster.
- Input TSC for Timestamping of CAN messages during CAN data transfer.
- Two Interrupts from the CAN controller to connect with AON VIC and LIC in the AON cluster.
- AON-CAN Interrupts are also routed to the AON-GTE module. The GTE module is used to timestamp various CAN events that can result in an interrupt.

9.4.2.9 CAN SoC Interface

Each CAN Interface has six pins:

1. CAN_TX
2. CAN_RX
3. CAN_STB
4. CAN_WAK
5. CAN_EN
6. CAN_ERR

These pins are driven from pads that have a switchable voltage supply (between 1.8V and 3.3V). Only CAN_TX and CAN_RX interface with the CAN Controller. Others are GPIOs controlled directly by CAN software.

9.4.3 Programming Guidelines

Assuming use of supplied software driver, this section covers only extra programming options provided.

9.4.3.1 Selecting Trigger Source for SWT and EVT

For both SWT and EVT inputs, an option is provided to select from four sources.

The following register fields are involved for EVT:

- EXT_SYNC_SELECT.EVT_SEL – External Synchronization trigger source for EVT
- CONTROL_REGISTER0.EVT – Software Trigger for EVT

Table 9.5 EVT_SEL Field

EXT_SYNC_SELECT.EVT_SEL	Comment
0	EVT pin of M_TT_CAN is driven from SoC output of another CAN controller
1	EVT pin of M_TT_CAN is driven from TMP output of another CAN controller
2	EVT pin of M_TT_CAN is driven from RTP output of another CAN controller
4	EVT pin of M_TT_CAN is driven from CONTROL_REGISTER0.EVT register field

The following register fields are involved for SWT:

- EXT_SYNC_SELECT.SWT_SEL – External Synchronization trigger source for SWT
- CONTROL_REGISTER0.SWT – Software Trigger for SWT

Table 9.6 SWT_SEL Field

EXT_SYNC_SELECT.SWT_SEL	Comment
0	SWT pin of M_TT_CAN is driven from SoC output of another CAN controller
1	SWT pin of M_TT_CAN is driven from TMP output of another CAN controller
2	SWT pin of M_TT_CAN is driven from RTP output of another CAN controller
4	SWT pin of M_TT_CAN is driven from CONTROL_REGISTER0.SWT register field

9.4.3.2 Selecting Sources for Fault Detect Signal to HSM

A fault detect signal is generated from each CAN controller and routed to HSM. The Fault detect signal from CAN to HSM is derived from the IR (interrupt register) and TTIR (time triggered interrupt register) buses provided. These are a total of 51 status registers. In addition, there are 51 mask bits provided to make a status Fault or non-Fault. Mask registers are:

- HSM_MASK0.IR
- HSM_MASK1.TTIR

Programming 1'b1 at any location makes corresponding interrupt as Fault. For details about each bit in IR and TTIR registers see M_TTCAN_IR_0, M_TTCAN_TTIR_0.

9.4.3.3 Orin Specific Controller Initialization

The Orin series SoC implementation of CAN does not include a Clock Calibration Unit (CCU), the m_ttcan_cok signal generally coming from the CCU is driven from a register bit in CAN Wrapper. As part of controller initialization assert the m_ttcan_cok signal (this is done by programming the CONTROL_REGISTER0.COK bit as 1).

To assert the m_ttcan_cok signal, program the CONTROL_REGISTER0.COK bit as 1.

9.4.3.4 Arbitration Between STD Message and EXT Message

Per the CAN 2.0 specification, EXT ID has two parts: Base ID and Extended ID.

- If Base ID of Extended ID Message is lower in priority (has a higher numerical value) than STD ID Message then STD ID Message WINS arbitration.
- If Base ID of Extended ID Message has same priority (has same numerical value) as STD ID Message then STD ID Message WINS arbitration.
- If Base ID of Extended ID Message is higher in priority (has lower numerical value) than STD ID Message then EXT ID Message WINS arbitration.

Further arbitration between EXT ID Messages with same Base ID is done based on Extended ID part.

9.4.3.5 Prefetch Feature on Transmit Message Handler

The Transmit Message Handler has a prefetch buffer of two, so it can retrieve two Messages from the CAN Message RAM, for which the request is pending, even if the controller is in INIT state. This may lead to unexpected ordering in transmission.

9.4.4 CAN Registers

Refer to "Reading Register Tables" in the Introduction chapter for the register table protocol as well as recommendations for accessing registers.

The Base Addresses of the registers related to CAN are specified in the Address Map section of the Orin TRM.

M_TTCAN_IR_0

M_TTCAN Interrupt Register Flags

Offset: 0x1000
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**IR**

M_TTCAN_TTIR_0

M_TTCAN Interrupt Register Flags

Offset: 0x1004
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,x000,0000,0000,0000,0000)

Bit	Reset	Description
18:0	0x0	**TTIR**

M_TTCAN_TXBRP_0

M_TTCAN Transmit Buffer Request Pending

Offset: 0x1008
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TXBRP**

M_TTCAN_FD_DATA_0

M_TTCAN CAN FD Data Lines

Offset: 0x100c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**TX**
0	0x0	**RX**

M_TTCAN_STATUS_REGISTER0_0

M_TTCAN Status Register

Offset: 0x1010
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,0000,0000)

Bit	Reset	Description
12:10	0x0	**FE**
9	0x0	**SOC**
8	0x0	**TMP**
7	0x0	**RTP**
6	0x0	**ASCM**
5	0x0	**ASCT**
4	0x0	**CCE**

Bit	Reset	Description
3	0x0	**SPT**
2	0x0	**MRX**
1	0x0	**CALF**
0	0x0	**AFF**

M_TTCAN_CONTROL_REGISTER0_0

M_TTCAN Control Register

Offset: 0x1014
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,xxxx,xxxx,xxxx,0000)
PROD: 0x00000008 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1xxx)

Bit	Reset	PROD	Description
31:16	0x0	_NONE_	**EXT_TS**
3	0x0	0x1	**COK**
2	0x0	_NONE_	**DIS_MORD**
1	0x0	_NONE_	**EVT**
0	0x0	_NONE_	**SWT**

M_TTCAN_DMA_INTF0_0

M_TTCAN DMA Interface

Offset: 0x1018
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	R/W	Reset	Description
1	RO	0x0	**DMA_REQ**
0	RW	0x0	**DMA_ACK**

M_TTCAN_CLK_STOP_0

M_TTCAN CLKSTOP Interface

Offset: 0x101c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	R/W	Reset	Description
1	RO	0x0	**ACK**
0	RW	0x0	**REQ**

M_TTCAN_HSM_MASK0_0

M_TTCAN HSM Mask for IR

Offset: 0x1020
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**IR**

M_TTCAN_HSM_MASK1_0

M_TTCAN HSM Mask for TTIR

Offset: 0x1024
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,x000,0000,0000,0000,0000)

Bit	Reset	Description
18:0	0x0	**TTIR**

M_TTCAN_EXT_SYNC_SELECT_0

M_TTCAN EVT / SWT Select

Offset: 0x1028
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,xxx0,0000)

Bit	Reset	Description
12:8	0x0	**EVT_SEL**
4:0	0x0	**SWT_SEL**

M_TTCAN_HSM_SW_OVERRIDE_0

M_TTCAN HSM Software Override

Offset: 0x102c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**EC_PUL1_OR**
1	0x0	**EC_PUL0_OR**
0	0x0	**OVRD**

M_TTCAN_TIME_STAMP_0

M_TTCAN Time Stamp Select

Offset: 0x1030
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,xxx0,0000)

Bit	Reset	Description
8	0x0	**TIMER_SEL**: Register or TSC counter value **1** = REG **0** = TSC
4:0	0x0	**OFFSET_SEL**: TSC offset selection

9.4.4.1 CAN Core Registers

An issue exists with some of the register bits, where the expected function reset-on-read and set-on-read does not work. Workarounds are noted in the register tables below.

The following register status bits inside CAN do not get reset when read: ECR.CEL, PSR.PXE, PSR.RFDF, PSR.RBRS, PSR.RESI.
The following register status bits inside CAN do not get set when read: PSR.DLEC, PSR.LEC.

M_TTCAN_CORE_CREL_0
Offset: 0x0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x32380609 (0b0011,0010,0011,1000,0000,0110,0000,1001)

Bit	Reset	Description
31:28	0x3	**REL**

Bit	Reset	Description
27:24	0x2	**STEP**
23:20	0x3	**SUBSTEP**
19:16	0x8	**YEAR**
15:8	0x6	**MON**
7:0	0x9	**DAY**

M_TTCAN_CORE_ENDN_0
Offset: 0x4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x87654321 (0b1000,0111,0110,0101,0100,0011,0010,0001)

Bit	Reset	Description
31:0	0x87654321	**ETV**

M_TTCAN_CORE_CUST_0
Offset: 0x8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**RES1**

M_TTCAN_CORE_DBTP_0
Offset: 0xc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000a33 (0bxxxx,xxxx,0xx0,0000,xxx0,1010,0011,0011)

Bit	Reset	Description
23	0x0	**TDC**
20:16	0x0	**DBRP**
12:8	0xa	**DTSEG1**
7:4	0x3	**DTSEG2**
3:0	0x3	**DSJW**

M_TTCAN_CORE_TEST_0

Offset: 0x10
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	R/W	Reset	Description
7	RO	0x0	**RX**
6:5	RW	0x0	**TX**
4	RW	0x0	**LBCK**
3	RO	0x0	**CAT**
2	RO	0x0	**CAM**
1	RW	0x0	**TAT**
0	RW	0x0	**TAM**

M_TTCAN_CORE_RWD_0

Offset: 0x14
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	R/W	Reset	Description
15:8	RO	0x0	**WDV**
7:0	RW	0x0	**WDC**

M_TTCAN_CORE_CCCR_0

Offset: 0x18
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,0000,xx00,0000,0001)

Bit	R/W	Reset	Description
15	RW	0x0	**NISO**
14	RW	0x0	**TXP**
13	RO	0x0	**EFBI**
12	RO	0x0	**PXHD**
9	RW	0x0	**BRSE**
8	RW	0x0	**FDOE**
7	RW	0x0	**TEST**
6	RW	0x0	**DAR**
5	RW	0x0	**MON**
4	RW	0x0	**CSR**
3	RO	0x0	**CSA**
2	RW	0x0	**ASM**
1	RW	0x0	**CCE**
0	RW	0x1	**INIT**

M_TTCAN_CORE_NBTP_0

Offset: 0x1c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x06000a03 (0b0000,0110,0000,0000,0000,1010,x000,0011)

Bit	Reset	Description
31:25	0x3	**NSJW**

Bit	Reset	Description
24:16	0x0	**NBRP**
15:8	0xa	**NTSEG1**
6:0	0x3	**NTSEG2**

M_TTCAN_CORE_TSCC_0
Offset: 0x20
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,0000,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
19:16	0x0	**TCP**
1:0	0x0	**TSS**

M_TTCAN_CORE_TSCV_0
Offset: 0x24
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TSC**

M_TTCAN_CORE_TOCC_0
Offset: 0x28
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffff0000 (0b1111,1111,1111,1111,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
31:16	0xffff	**TOP**
2:1	0x0	**TOS**
0	0x0	**ETOC**

M_TTCAN_CORE_TOCV_0

Offset: 0x2c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000ffff (0bxxxx,xxxx,xxxx,xxxx,1111,1111,1111,1111)

Bit	Reset	Description
15:0	0xffff	**TOC**

M_TTCAN_CORE_ECR_0

Offset: 0x40
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:16	0x0	**CEL**
15	0x0	**RP**
14:8	0x0	**REC**
7:0	0x0	**TEC**

M_TTCAN_CORE_PSR_0

Offset: 0x44
Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000707 (0bxxxx,xxxx,x000,0000,x000,0111,0000,0111)

Bit	R/W	Reset	Description
22:16	RW	0x0	**TDCV**
14	RW	0x0	**PXE**
13	RW	0x0	**RFDF**
12	RW	0x0	**RBRS**
11	RW	0x0	**RESI**
10:8	RW	0x7	**DLEC**
7	RO	0x0	**BO**
6	RO	0x0	**EW**
5	RO	0x0	**EP**
4:3	RO	0x0	**ACT**
2:0	RW	0x7	**LEC**

M_TTCAN_CORE_TDCR_0
Offset: 0x48
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,x000,0000)

Bit	Reset	Description
14:8	0x0	**TDCO**
6:0	0x0	**TDCF**

M_TTCAN_CORE_IR_0
Offset: 0x50
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29	0x0	**ARA**
28	0x0	**PED**
27	0x0	**PEA**
26	0x0	**WDI**
25	0x0	**BO**
24	0x0	**EW**
23	0x0	**EP**
22	0x0	**ELO**
21	0x0	**BEU**
20	0x0	**BEC**
19	0x0	**DRX**
18	0x0	**TOO**
17	0x0	**MRAF**
16	0x0	**TSW**
15	0x0	**TEFL**
14	0x0	**TEFF**
13	0x0	**TEFW**
12	0x0	**TEFN**
11	0x0	**TFE**
10	0x0	**TCF**
9	0x0	**TC**
8	0x0	**HPM**
7	0x0	**RF1L**
6	0x0	**RF1F**
5	0x0	**RF1W**
4	0x0	**RF1N**
3	0x0	**RF0L**
2	0x0	**RF0F**
1	0x0	**RF0W**

Bit	Reset	Description
0	0x0	**RF0N**

M_TTCAN_CORE_IE_0

Offset: 0x54
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29	0x0	**ARAE**
28	0x0	**PEDE**
27	0x0	**PEAE**
26	0x0	**WDIE**
25	0x0	**BOE**
24	0x0	**EWE**
23	0x0	**EPE**
22	0x0	**ELOE**
21	0x0	**BEUE**
20	0x0	**BECE**
19	0x0	**DRXE**
18	0x0	**TOOE**
17	0x0	**MRAFE**
16	0x0	**TSWE**
15	0x0	**TEFLE**
14	0x0	**TEFFE**
13	0x0	**TEFWE**
12	0x0	**TEFNE**
11	0x0	**TFEE**
10	0x0	**TCFE**

Bit	Reset	Description
9	0x0	**TCE**
8	0x0	**HPME**
7	0x0	**RF1LE**
6	0x0	**RF1FE**
5	0x0	**RF1WE**
4	0x0	**RF1NE**
3	0x0	**RF0LE**
2	0x0	**RF0FE**
1	0x0	**RF0WE**
0	0x0	**RF0NE**

M_TTCAN_CORE_ILS_0

Offset: 0x58
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
29	0x0	**ARAL**
28	0x0	**PEDL**
27	0x0	**PEAL**
26	0x0	**WDIL**
25	0x0	**BOL**
24	0x0	**EWL**
23	0x0	**EPL**
22	0x0	**ELOL**
21	0x0	**BEUL**
20	0x0	**BECL**
19	0x0	**DRXL**

Bit	Reset	Description
18	0x0	**TOOL**
17	0x0	**MRAFL**
16	0x0	**TSWL**
15	0x0	**TEFLL**
14	0x0	**TEFFL**
13	0x0	**TEFWL**
12	0x0	**TEFNL**
11	0x0	**TFEL**
10	0x0	**TCFL**
9	0x0	**TCL**
8	0x0	**HPML**
7	0x0	**RF1LL**
6	0x0	**RF1FL**
5	0x0	**RF1WL**
4	0x0	**RF1NL**
3	0x0	**RF0LL**
2	0x0	**RF0FL**
1	0x0	**RF0WL**
0	0x0	**RF0NL**

M_TTCAN_CORE_ILE_0

Offset: 0x5c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**EINT1**
0	0x0	**ENIT0**

M_TTCAN_CORE_GFC_0

Offset: 0x80
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:4	0x0	**ANFS**
3:2	0x0	**ANFE**
1	0x0	**RRFS**
0	0x0	**RRFE**

M_TTCAN_CORE_SIDFC_0

Offset: 0x84
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
23:16	0x0	**LSS**
15:2	0x0	**FLSSA**

M_TTCAN_CORE_XIDFC_0

Offset: 0x88
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,x000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
22:16	0x0	**LSE**

Bit	Reset	Description
15:2	0x0	**FLESA**

M_TTCAN_CORE_XIDAM_0
Offset: 0x90
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x1fffffff (0bxxx1,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
28:0	0x1fffffff	**EIDM**

M_TTCAN_CORE_HPMS_0
Offset: 0x94
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15	0x0	**FLST**
14:8	0x0	**FIDX**
7:6	0x0	**MSI**
5:0	0x0	**BIDX**

M_TTCAN_CORE_NDAT1_0
Offset: 0x98
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**ND31**
30	0x0	**ND30**
29	0x0	**ND29**
28	0x0	**ND28**
27	0x0	**ND27**
26	0x0	**ND26**
25	0x0	**ND25**
24	0x0	**ND24**
23	0x0	**ND23**
22	0x0	**ND22**
21	0x0	**ND21**
20	0x0	**ND20**
19	0x0	**ND19**
18	0x0	**ND18**
17	0x0	**ND17**
16	0x0	**ND16**
15	0x0	**ND15**
14	0x0	**ND14**
13	0x0	**ND13**
12	0x0	**ND12**
11	0x0	**ND11**
10	0x0	**ND10**
9	0x0	**ND9**
8	0x0	**ND8**
7	0x0	**ND7**
6	0x0	**ND6**
5	0x0	**ND5**
4	0x0	**ND4**
3	0x0	**ND3**

Bit	Reset	Description
2	0x0	**ND2**
1	0x0	**ND1**
0	0x0	**ND0**

M_TTCAN_CORE_NDAT2_0

Offset: 0x9c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**ND63**
30	0x0	**ND62**
29	0x0	**ND61**
28	0x0	**ND60**
27	0x0	**ND59**
26	0x0	**ND58**
25	0x0	**ND57**
24	0x0	**ND56**
23	0x0	**ND55**
22	0x0	**ND54**
21	0x0	**ND53**
20	0x0	**ND52**
19	0x0	**ND51**
18	0x0	**ND50**
17	0x0	**ND49**
16	0x0	**ND48**
15	0x0	**ND47**
14	0x0	**ND46**

Bit	Reset	Description
13	0x0	**ND45**
12	0x0	**ND44**
11	0x0	**ND43**
10	0x0	**ND42**
9	0x0	**ND41**
8	0x0	**ND40**
7	0x0	**ND39**
6	0x0	**ND38**
5	0x0	**ND37**
4	0x0	**ND36**
3	0x0	**ND35**
2	0x0	**ND34**
1	0x0	**ND33**
0	0x0	**ND32**

M_TTCAN_CORE_RXF0C_0

Offset: 0xa0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,x000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31	0x0	**F0OM**
30:24	0x0	**F0WM**
22:16	0x0	**F0S**
15:2	0x0	**F0SA**

M_TTCAN_CORE_RXF0S_0

Offset: 0xa4
Read/Write: RO

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xx00,0000,xx00,0000,x000,0000)

Bit	Reset	Description
25	0x0	**RF0L**
24	0x0	**F0F**
21:16	0x0	**F0PI**
13:8	0x0	**F0GI**
6:0	0x0	**F0FL**

M_TTCAN_CORE_RXF0A_0
Offset: 0xa8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**F0AI**

M_TTCAN_CORE_RXBC_0
Offset: 0xac
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,00xx)

Bit	Reset	Description
15:2	0x0	**RBSA**

M_TTCAN_CORE_RXF1C_0
Offset: 0xb0
Read/Write: R/W

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,x000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
31	0x0	**F1OM**
30:24	0x0	**F1WM**
22:16	0x0	**F1S**
15:2	0x0	**F1SA**

M_TTCAN_CORE_RXF1S_0

Offset: 0xb4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b00xx,xx00,xx00,0000,xx00,0000,x000,0000)

Bit	Reset	Description
31:30	0x0	**DMS**
25	0x0	**RF1L**
24	0x0	**F1F**
21:16	0x0	**F1PI**
13:8	0x0	**F1GI**
6:0	0x0	**F1FL**

M_TTCAN_CORE_RXF1A_0

Offset: 0xb8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**F1AI**

M_TTCAN_CORE_RXESC_0

Offset: 0xbc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,x000,x000,x000)

Bit	Reset	Description
10:8	0x0	**RBDS**
6:4	0x0	**F1DS**
2:0	0x0	**F0DS**

M_TTCAN_CORE_TXBC_0

Offset: 0xc0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bx000,0000,xx00,0000,0000,0000,0000,00xx)

Bit	Reset	Description
30	0x0	**TFQM**
29:24	0x0	**TFQS**
21:16	0x0	**NDTB**
15:2	0x0	**TBSA**

M_TTCAN_CORE_TXFQS_0

Offset: 0xc4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xx00,0000,xxx0,0000,xx00,0000)

Bit	Reset	Description
21	0x0	**TFQF**
20:16	0x0	**TFQPI**
12:8	0x0	**TFGI**
5:0	0x0	**TFFL**

M_TTCAN_CORE_TXESC_0

Offset: 0xc8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2:0	0x0	**TBDS**

M_TTCAN_CORE_TXBRP_0

Offset: 0xcc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**TRP31**
30	0x0	**TRP30**
29	0x0	**TRP29**
28	0x0	**TRP28**
27	0x0	**TRP27**
26	0x0	**TRP26**
25	0x0	**TRP25**
24	0x0	**TRP24**
23	0x0	**TRP23**

Bit	Reset	Description
22	0x0	**TRP22**
21	0x0	**TRP21**
20	0x0	**TRP20**
19	0x0	**TRP19**
18	0x0	**TRP18**
17	0x0	**TRP17**
16	0x0	**TRP16**
15	0x0	**TRP15**
14	0x0	**TRP14**
13	0x0	**TRP13**
12	0x0	**TRP12**
11	0x0	**TRP11**
10	0x0	**TRP10**
9	0x0	**TRP9**
8	0x0	**TRP8**
7	0x0	**TRP7**
6	0x0	**TRP6**
5	0x0	**TRP5**
4	0x0	**TRP4**
3	0x0	**TRP3**
2	0x0	**TRP2**
1	0x0	**TRP1**
0	0x0	**TRP0**

M_TTCAN_CORE_TXBAR_0
Offset: 0xd0
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**AR31**
30	0x0	**AR30**
29	0x0	**AR29**
28	0x0	**AR28**
27	0x0	**AR27**
26	0x0	**AR26**
25	0x0	**AR25**
24	0x0	**AR24**
23	0x0	**AR23**
22	0x0	**AR22**
21	0x0	**AR21**
20	0x0	**AR20**
19	0x0	**AR19**
18	0x0	**AR18**
17	0x0	**AR17**
16	0x0	**AR16**
15	0x0	**AR15**
14	0x0	**AR14**
13	0x0	**AR13**
12	0x0	**AR12**
11	0x0	**AR11**
10	0x0	**AR10**
9	0x0	**AR9**
8	0x0	**AR8**
7	0x0	**AR7**
6	0x0	**AR6**
5	0x0	**AR5**
4	0x0	**AR4**

Bit	Reset	Description
3	0x0	**AR3**
2	0x0	**AR2**
1	0x0	**AR1**
0	0x0	**AR0**

M_TTCAN_CORE_TXBCR_0

Offset: 0xd4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**CR31**
30	0x0	**CR30**
29	0x0	**CR29**
28	0x0	**CR28**
27	0x0	**CR27**
26	0x0	**CR26**
25	0x0	**CR25**
24	0x0	**CR24**
23	0x0	**CR23**
22	0x0	**CR22**
21	0x0	**CR21**
20	0x0	**CR20**
19	0x0	**CR19**
18	0x0	**CR18**
17	0x0	**CR17**
16	0x0	**CR16**
15	0x0	**CR15**

Bit	Reset	Description
14	0x0	**CR14**
13	0x0	**CR13**
12	0x0	**CR12**
11	0x0	**CR11**
10	0x0	**CR10**
9	0x0	**CR9**
8	0x0	**CR8**
7	0x0	**CR7**
6	0x0	**CR6**
5	0x0	**CR5**
4	0x0	**CR4**
3	0x0	**CR3**
2	0x0	**CR2**
1	0x0	**CR1**
0	0x0	**CR0**

M_TTCAN_CORE_TXBTO_0

Offset: 0xd8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**TO31**
30	0x0	**TO30**
29	0x0	**TO29**
28	0x0	**TO28**
27	0x0	**TO27**
26	0x0	**TO26**

Bit	Reset	Description
25	0x0	**TO25**
24	0x0	**TO24**
23	0x0	**TO23**
22	0x0	**TO22**
21	0x0	**TO21**
20	0x0	**TO20**
19	0x0	**TO19**
18	0x0	**TO18**
17	0x0	**TO17**
16	0x0	**TO16**
15	0x0	**TO15**
14	0x0	**TO14**
13	0x0	**TO13**
12	0x0	**TO12**
11	0x0	**TO11**
10	0x0	**TO10**
9	0x0	**TO9**
8	0x0	**TO8**
7	0x0	**TO7**
6	0x0	**TO6**
5	0x0	**TO5**
4	0x0	**TO4**
3	0x0	**TO3**
2	0x0	**TO2**
1	0x0	**TO1**
0	0x0	**TO0**

M_TTCAN_CORE_TXBCF_0
Offset: 0xdc

Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**CF31**
30	0x0	**CF30**
29	0x0	**CF29**
28	0x0	**CF28**
27	0x0	**CF27**
26	0x0	**CF26**
25	0x0	**CF25**
24	0x0	**CF24**
23	0x0	**CF23**
22	0x0	**CF22**
21	0x0	**CF21**
20	0x0	**CF20**
19	0x0	**CF19**
18	0x0	**CF18**
17	0x0	**CF17**
16	0x0	**CF16**
15	0x0	**CF15**
14	0x0	**CF14**
13	0x0	**CF13**
12	0x0	**CF12**
11	0x0	**CF11**
10	0x0	**CF10**
9	0x0	**CF9**
8	0x0	**CF8**
7	0x0	**CF7**
6	0x0	**CF6**

Bit	Reset	Description
5	0x0	**CF5**
4	0x0	**CF4**
3	0x0	**CF3**
2	0x0	**CF2**
1	0x0	**CF1**
0	0x0	**CF0**

M_TTCAN_CORE_TXBTIE_0

Offset: 0xe0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**TIE31**
30	0x0	**TIE30**
29	0x0	**TIE29**
28	0x0	**TIE28**
27	0x0	**TIE27**
26	0x0	**TIE26**
25	0x0	**TIE25**
24	0x0	**TIE24**
23	0x0	**TIE23**
22	0x0	**TIE22**
21	0x0	**TIE21**
20	0x0	**TIE20**
19	0x0	**TIE19**
18	0x0	**TIE18**
17	0x0	**TIE17**

Bit	Reset	Description
16	0x0	**TIE16**
15	0x0	**TIE15**
14	0x0	**TIE14**
13	0x0	**TIE13**
12	0x0	**TIE12**
11	0x0	**TIE11**
10	0x0	**TIE10**
9	0x0	**TIE9**
8	0x0	**TIE8**
7	0x0	**TIE7**
6	0x0	**TIE6**
5	0x0	**TIE5**
4	0x0	**TIE4**
3	0x0	**TIE3**
2	0x0	**TIE2**
1	0x0	**TIE1**
0	0x0	**TIE0**

M_TTCAN_CORE_TXBCIE_0

Offset: 0xe4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**CFIE31**
30	0x0	**CFIE30**
29	0x0	**CFIE29**
28	0x0	**CFIE28**

Bit	Reset	Description
27	0x0	**CFIE27**
26	0x0	**CFIE26**
25	0x0	**CFIE25**
24	0x0	**CFIE24**
23	0x0	**CFIE23**
22	0x0	**CFIE22**
21	0x0	**CFIE21**
20	0x0	**CFIE20**
19	0x0	**CFIE19**
18	0x0	**CFIE18**
17	0x0	**CFIE17**
16	0x0	**CFIE16**
15	0x0	**CFIE15**
14	0x0	**CFIE14**
13	0x0	**CFIE13**
12	0x0	**CFIE12**
11	0x0	**CFIE11**
10	0x0	**CFIE10**
9	0x0	**CFIE9**
8	0x0	**CFIE8**
7	0x0	**CFIE7**
6	0x0	**CFIE6**
5	0x0	**CFIE5**
4	0x0	**CFIE4**
3	0x0	**CFIE3**
2	0x0	**CFIE2**
1	0x0	**CFIE1**
0	0x0	**CFIE0**

M_TTCAN_CORE_TXEFC_0

Offset: 0xf0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,xx00,0000,0000,0000,0000,00xx)

Bit	Reset	Description
29:24	0x0	**EFWM**
21:16	0x0	**EFS**
15:2	0x0	**EFSA**

M_TTCAN_CORE_TXEFS_0

Offset: 0xf4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,xxxx,0000,xxx0,0000,xx00,0000)

Bit	Reset	Description
25	0x0	**TEFL**
24	0x0	**EFF**
19:16	0x0	**EFPI**
12:8	0x0	**EFGI**
5:0	0x0	**EFFL**

M_TTCAN_CORE_TXEFA_0

Offset: 0xf8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000)

Bit	Reset	Description
4:0	0x0	**EFAI**

M_TTCAN_CORE_TTTMC_0
Offset: 0x100
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,x000,0000,0000,0000,0000,00xx)

Bit	Reset	Description
22:16	0x0	**TME**
15:2	0x0	**TMSA**

M_TTCAN_CORE_TTRMC_0
Offset: 0x104
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b00x0,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**RMPS**
30	0x0	**XTD**
28:0	0x0	**RID**

M_TTCAN_CORE_TTOCF_0
Offset: 0x108
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00010000 (0bxxxx,x000,0000,0001,0000,0000,0000,0x00)

Bit	Reset	Description
26	0x0	**EVTP**
25	0x0	**ECC**
24	0x0	**EGTF**
23:16	0x1	**AWL**
15	0x0	**EECS**
14:8	0x0	**IRTO**
7:5	0x0	**LDSDL**
4	0x0	**TM**
3	0x0	**GEN**
1:0	0x0	**OM**

M_TTCAN_CORE_TTMLM_0
Offset: 0x10c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,0000,0000,0000,xxxx,0000,0000,0000)

Bit	Reset	Description
27:16	0x0	**ENTT**
11:8	0x0	**TXEW**
7:6	0x0	**CSS**
5:0	0x0	**CCM**

M_TTCAN_CORE_TURCF_0
Offset: 0x110
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x10000000 (0b0x01,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**ELT**
29:16	0x1000	**DC**
15:0	0x0	**NCL**

M_TTCAN_CORE_TTOCN_0

Offset: 0x114
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0x00,0000,0000,0000)

Bit	R/W	Reset	Description
15	RO	0x0	**LCKC**
13	RW	0x0	**ESCN**
12	RW	0x0	**NIG**
11	RW	0x0	**TMG**
10	RW	0x0	**FGP**
9	RW	0x0	**GCS**
8	RW	0x0	**TTIE**
7:6	RW	0x0	**TMC**
5	RW	0x0	**RTIE**
4:3	RW	0x0	**SWS**
2	RW	0x0	**SWP**
1	RW	0x0	**ECS**
0	RW	0x0	**SGT**

M_TTCAN_CORE_TTGTP_0

Offset: 0x118
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CTP**
15:0	0x0	**TP**

M_TTCAN_CORE_TTTMK_0

Offset: 0x11c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,x000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
31	RO	0x0	**LCKM**
22:16	RW	0x0	**TICC**
15:0	RW	0x0	**TM**

M_TTCAN_CORE_TTIR_0

Offset: 0x120
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,x000,0000,0000,0000,0000)

Bit	Reset	Description
18	0x0	**CER**
17	0x0	**AW**
16	0x0	**WT**
15	0x0	**IWT**
14	0x0	**ELC**
13	0x0	**SE2**
12	0x0	**SE1**

Bit	Reset	Description
11	0x0	**TXO**
10	0x0	**TXU**
9	0x0	**GTE**
8	0x0	**GTD**
7	0x0	**GTW**
6	0x0	**SWE**
5	0x0	**TTMI**
4	0x0	**RTMI**
3	0x0	**SOG**
2	0x0	**CSM**
1	0x0	**SMC**
0	0x0	**SBC**

M_TTCAN_CORE_TTIE_0

Offset: 0x124
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,x000,0000,0000,0000,0000)

Bit	Reset	Description
18	0x0	**CERE**
17	0x0	**AWE**
16	0x0	**WTE**
15	0x0	**IWTE**
14	0x0	**ELCE**
13	0x0	**SE2E**
12	0x0	**SE1E**
11	0x0	**TXOE**
10	0x0	**TXUE**

Bit	Reset	Description
9	0x0	**GTEE**
8	0x0	**GTDE**
7	0x0	**GTWE**
6	0x0	**SWEE**
5	0x0	**TTMIE**
4	0x0	**RTMIE**
3	0x0	**SOGE**
2	0x0	**CSME**
1	0x0	**SMCE**
0	0x0	**SBCE**

M_TTCAN_CORE_TTILS_0

Offset: 0x128
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,x000,0000,0000,0000,0000)

Bit	Reset	Description
18	0x0	**CERL**
17	0x0	**AWL**
16	0x0	**WTL**
15	0x0	**IWTL**
14	0x0	**ELCL**
13	0x0	**SE2L**
12	0x0	**SE1L**
11	0x0	**TXOL**
10	0x0	**TXUL**
9	0x0	**GTEL**
8	0x0	**GTDL**

Bit	Reset	Description
7	0x0	**GTWL**
6	0x0	**SWEL**
5	0x0	**TTMIL**
4	0x0	**RTMIL**
3	0x0	**SOGL**
2	0x0	**CSML**
1	0x0	**SMCL**
0	0x0	**SBCL**

M_TTCAN_CORE_TTOST_0

Offset: 0x12c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000080 (0b0000,0000,00xx,xxxx,0000,0000,1000,0000)

Bit	Reset	Description
31	0x0	**SPL**
30	0x0	**WECS**
29	0x0	**AWE**
28	0x0	**WFE**
27	0x0	**GSI**
26:24	0x0	**TMP**
23	0x0	**GFI**
22	0x0	**WGTD**
15:8	0x0	**RTO**
7	0x1	**QCS**
6	0x0	**QGTP**
5:4	0x0	**SYS**
3:2	0x0	**MS**

Bit	Reset	Description
1:0	0x0	**EL**

M_TTCAN_CORE_TURNA_0
Offset: 0x130
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00010000 (0bxxxx,xxxx,xxxx,xx01,0000,0000,0000,0000)

Bit	Reset	Description
17:0	0x10000	**NAV**

M_TTCAN_CORE_TTLGT_0
Offset: 0x134
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**GT**
15:0	0x0	**LT**

M_TTCAN_CORE_TTCTC_0
Offset: 0x138
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x003f0000 (0bxxxx,xxxx,xx11,1111,0000,0000,0000,0000)

Bit	Reset	Description
21:16	0x3f	**CC**
15:0	0x0	**CT**

M_TTCAN_CORE_TTCPT_0
Offset: 0x13c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
31:16	0x0	**SWV**
5:0	0x0	**CCV**

M_TTCAN_CORE_TTCSM_0
Offset: 0x140
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**CSM**

9.5 SDMMC Controller

9.5.1 Overview

The SDMMC Controller is capable of interfacing to SD/eSD, SDIO, and eMMC devices. It has a direct interface to the memory controller, and is capable of initiating data transfers between system memory and an external card or device. It has an Control Plane interface to allow system access its configuration registers. To access the on-chip System RAM during boot, the SDMMC controller relies on the coherent steering path in the Memory Controller.

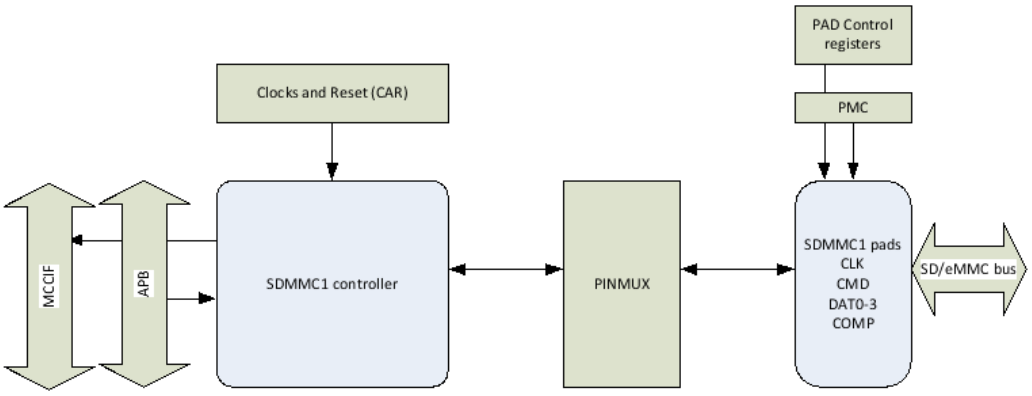
The SDMMC controller supports two different industry standard bus protocols:

1. SD/SDIO bus protocol for the SD/SDIO cards
2. eMMC bus protocol for eMMC devices

The System-on-Chip (SoC) supports two SDMMC controllers with pin-muxing support. Each SDMMC controller can interface with a SD/SDIO/eMMC device. Each SDMMC controller is capable of all listed protocols. It is the SDMMC controller, together with the type of pads it connects to, that defines the use case for that SDMMC. The following lists typical use cases for SDMMC controllers.

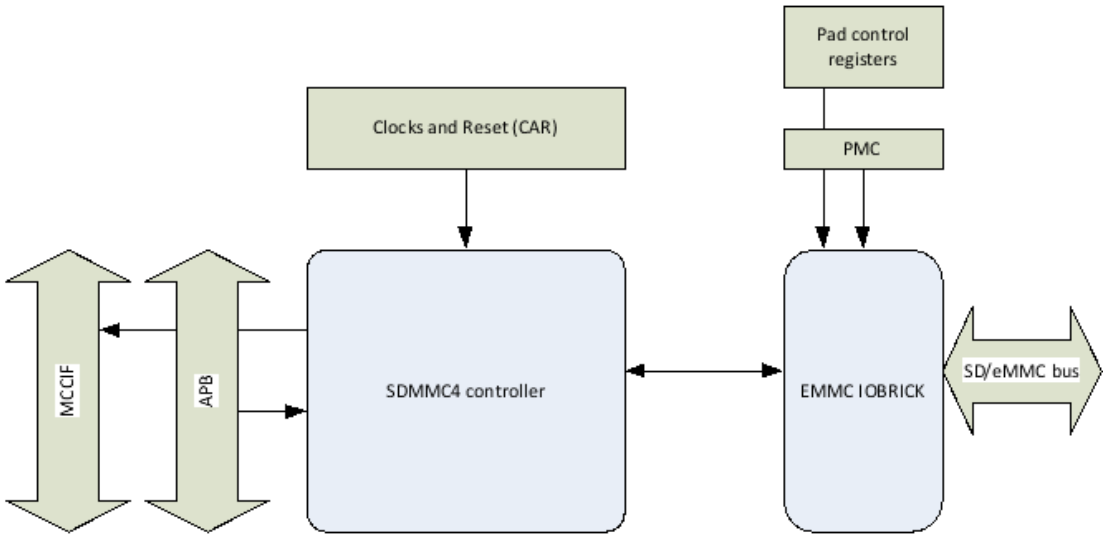
1. SDMMC1: Primary removable SD card with 3.3 V and 1.8 V I/O support or SDIO WIFI/MODEM
2. SDMMC4: eMMC5.1 device (embedded flash memory) for Booting and non-volatile storage with 1.8 V I/O support.

Figure 9.17 SDMMC1 Interface



*COMP is compensation pad used for pad drive strength calibration.

Figure 9.18 SDMMC4 Interface



9.5.2 SDMMC Controller Functional Description

9.5.2.1 Supported Features

- All SDMMC interfaces have pad calibration support
- Adheres to below SDA and JEDEC specifications
- SD Host Controller Specification Version 4.2 without UHS-II interface support
- SD Physical Layer Specification Version 4.2 without UHS-II interface support
- SDIO Physical Layer Specification Version 4.1 without UHS-II interface support
- eMMC Specification Version 5.1 with no support for 1.2V I/O signaling
- eMMC 5.1 CMD Queuing Host Controller Interface (CQHCI) specification
- Supports SD4.2 and SDIO4.1 devices (only legacy SD mode; no UHS-II mode)
- Supports eMMC 5.1 devices with CMD queuing feature only on SDMMC4
- Support of 1/4/8bit data interface for eMMC and eSDIO devices
- Support 1/4bit data interface for SD and SDIO cards
- Allow card to interrupt host in 1bit, 4bit, 8bit SDIO modes
- Supports up to 400MB/s data rate using 8 parallel data lines (eMMC HS400 mode) at 200 MHz. Both default and enhanced strobe modes are supported in this speed
  - The eMMC IOBRICK attached to the SDMMC4 controller must be used for this application, to meet timing requirements of HS400
- Up to 104 MB/s data rate using 4 parallel data lines (SD/SDIO 4-bit mode) at 208 MHz
- Up to 208 MB/s data rate using 8 parallel data lines (eSDIO 8-bit mode) at 208 MHz

- Supports SD/eMMC interface data flow control based on internal FIFO overrun and under-run status by stopping device clock
- Auto-tuning support in hardware
- DVFS independent trimmer for inbound clock path to avoid errors when DVFS changes Vcore
- SDMMC pads use internal clock loopback. Deep loopback option is supported in BDSDMEM* pads and IOBRICK to generate loopback clock with no reflections
- DLL calibration controller is added to determine delay code to use by Slave DLLs in iobrick to generate quarter cycle delay
- TMCLK (12 MHz) is used to implement data timeout counter in SD/eMMC legacy modes
- Support to access eMMC during IST

9.5.2.2 Supported Specifications and Standards

The SDMMC controller supports the specifications published by the SD Card Association and JEDEC (MMC) at the versions listed in the following table.

Table 9.7 Protocol Versions

Protocol	Version
SD	4.2[1]
SDIO	4.1[1]
eSD	2.1
eMMC	5.1
SDHOST	4.2[1]

1. SD 4x with legacy SD interface only supported – no UHS-II support.

These tables show the maximum data rates available over the physical interface. The speeds achievable in actual use is less than these rates, depending on the performance of the board and the device itself, protocol limitations, and the software driver.

For SD data transfer modes, the following table lists the maximum data transfer speeds:

Table 9.8 Maximum Data Transfer Speed (SD)

Speed Mode	Signal Voltage	I/O Frequency (MHz)	Bus Width	Max Throughput (MB/S)
Default Speed	3.3	25	1,4	12.5
High Speed	3.3	50	1,4	25

Speed Mode	Signal Voltage	I/O Frequency (MHz)	Bus Width	Max Throughput (MB/S)
SDR12	1.8	25	1,4	12.5
SDR25	1.8	50	1,4	25
SDR50	1.8	100	1,4	50
DDR50 [1]	1.8	50	1,4	50
SDR104	1.8	208	1,4	104

1. DDR50 is not supported, only SDR50 is supported. But Software can enable the DDR50 support by overriding the capability register bit, subject to the clock divider restriction.

Table 9.9 Maximum Data Transfer Speed (eSD 2.1)

Speed Mode	Signal Voltage	I/O Frequency (MHz)	Bus Width	Max Throughput (MB/S)	eSD protocol
Default Speed	3.3	25	1,4,8	12.5	eSD2.1
High Speed	3.3	50	1,4,8	25	eSD2.1

Table 9.10 Maximum Data Transfer Speed (SDIO)

Speed Mode	I/O Frequency (MHz)	Bus Width	Max Throughput (Mbytes/S)
Default Speed	25	1,4	12.5
High Speed	50	1,4	25
SDR12	25	1,4	12.5
SDR25	50	1,4	25
SDR50	100	1,4	50
DDR50	50	1,4	50
SDR104	208	1,4	104

Table 9.11 Maximum Data Transfer Speed (eMMC)

Speed Mode	I/O Frequency (MHz)	Bus Width	Max Throughput (Mbytes/S)	MMC rev
Legacy Speed	26	1,4,8	26 MB	MMC4.3
High Speed SDR	52	1,4,8	52 MB	MMC4.3
High Speed DDR	52	4,8	104	MMC4.4
HS200 (SDR)	200	4,8	200	MMC4.51

Speed Mode	I/O Frequency (MHz)	Bus Width	Max Throughput (Mbytes/S)	MMC rev
HS400 (DDR)	200	8	400	eMMC5.0

9.5.2.3 Hardware/Software Partitioning

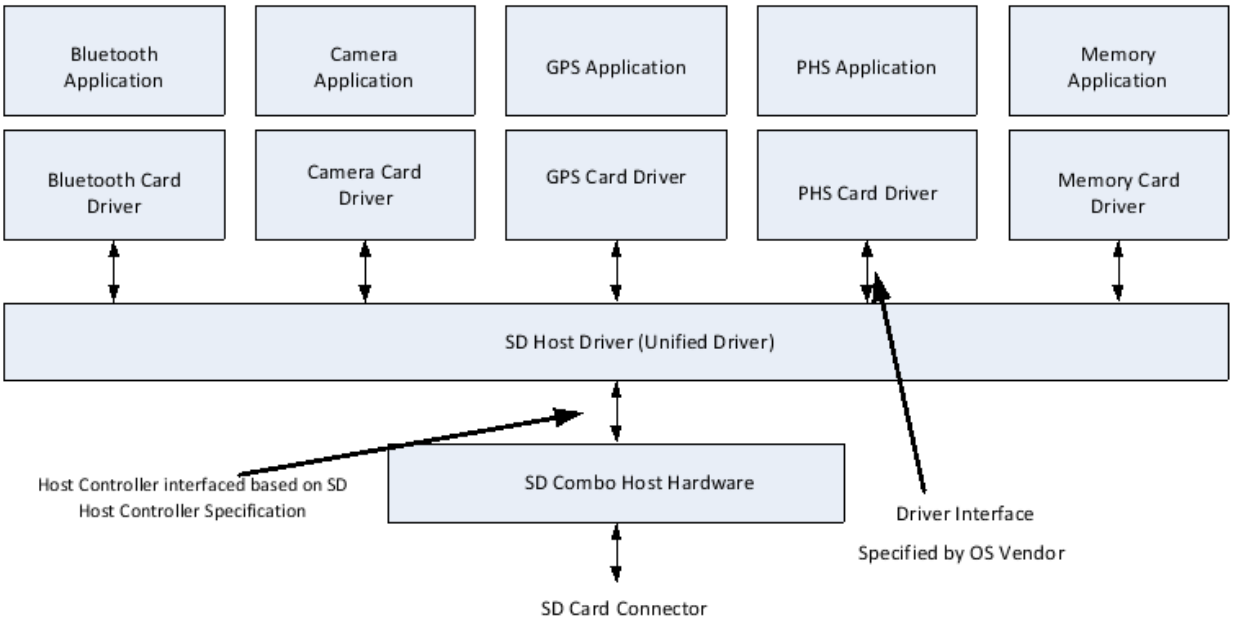
9.5.2.3.1 Hardware

The role of the hardware controller is to send out the programmed command, store the response, and make it visible to software. It updates the status registers and generates interrupts when attention is required. Software may also configure it for DMA operation.

9.5.2.3.2 Software

The software driver determines which type of card is inserted in the slot by sending the initialization commands and observing the responses received from the card. After identifying the card that is inserted, it should only program the corresponding set of commands that are applicable. It can enable interrupts for which notification is desired.

Figure 9.19 Host Hardware and Driver Architecture



9.5.2.3.3 Caveats and Assumptions

A single SDMMC controller can handle only one device at a time.

PMC can be configured to wake the device from the SC7 power state based on either:

- Card Detection pin
- Asynchronous interrupt on the DAT1 line

Write protect and card detect logic is implemented only for SDMMC1. Off-card ECC, described in the MMC specification, is not supported.

Table 9.12 SDMMC I/O Interfaces

Controller	No of Pinmuxings		Data Bus width	Pads Used	I/O Voltage (V)	Max. I/O Clock supported (MHz)	Speed Modes Supported			Typical Use Case	Maximum Bandwidth (MBps)
							SD card	SDIO	eMMC		
SDMMC1	1	SDMMC1	4	BDSDMEM	3.3/1.8	208	DS, HS, SDR12, SDR25, SDR50, SDR104	DS, HS, SDR12, SDR25, SDR50, and SDR104	Legacy MMC SDR, high-speed EMMC SDR, high-speed EMMC DDR and HS200 in 4-bit mode. No HS400 and CQ support.	Primary removable SD card or 4-bit SDIO or eMMC (4-bit support only and no CQE)	104
SDMMC4	1	SDMMC4	8	BDEMMC_IOBRICK	1.8	200	Removable SD card not supported	DS, HS, SDR12, SDR25, SDR50, and SDR104	Legacy MMC SDR, high-speed EMMC SDR, high-speed EMMC DDR, HS200, HS400 + CQE	Bootable eMMC or 4-bit SDIO or 8-bit eSDIO	400

Table 9.13 SD/eMMC I/O Interfaces

Controller	Host Specification Support	Devices Supported	Maximum Interface Speed
SDMMC1	SDHCI4.2 (no eMMC CQHCI support)	4-bit SDIO4.1 4-bit SD4.2 4-bit eMMC5.1 (up to HS200 + No CQE support)	SDR104 @208 MHz
SDMMC4	SDHCI4.2 + eMMC5.1 CQHCI	8-bit eMMC 5.1 8- or 4-bit SDIO 4.1	HS400 @200 MHz

Notes:

1. Removable MMC is not supported.
2. All SDMMC interfaces have pad calibration support.
3. Only UHS-I interface is supported for SD4.2/SDIO4.1 devices.
4. SDMMC1 do not support CQE since these interfaces supports up to eMMC4.51 speeds – HS200 only.
5. Dual eMMC is not supported.

Table 9.14 PINMUX Options

Controller	Pin Configuration	Signal	Ball Name	Pinmux Select
SDMMC1				
	sdmmc1_cmd_cfg	SDMMC1_CMD	SDMMC1_CMD	sdmmc1_cmd_pm
	sdmmc1_clk_cfg	SDMMC1_CLK	SDMMC1_CLK	sdmmc1_clk_pm
	sdmmc1_dat0_cfg	SDMMC1_DAT0	SDMMC1_DAT0	sdmmc1_dat0_pm
	sdmmc1_dat1_cfg	SDMMC1_DAT1	SDMMC1_DAT1	sdmmc1_dat1_pm
	sdmmc1_dat2_cfg	SDMMC1_DAT2	SDMMC1_DAT2	sdmmc1_dat2_pm
	sdmmc1_dat3_cfg	SDMMC1_DAT3	SDMMC1_DAT3	sdmmc1_dat3_pm
	sdmmc1_comp_cfg	SDMMC1_COMP	SDMMC1_COMP	sdmmc1_comp_pm
	soc_gpio_31_cfg	SDMMC1_WP	SOC_GPIO_31	soc_gpio_31_pm
	soc_gpio_11_cfg	SDMMC1_CD	SOC_GPIO_11	soc_gpio_11_pm
SDMMC4				No pinmuxing
	emmc4_cfg	SDMMC4_CMD	SDMMC4_CMD	IOBRICK
	emmc4_cfg	SDMMC4_CLK	SDMMC4_CLK	IOBRICK
	emmc4_cfg	SDMMC4_DAT0	SDMMC4_DAT0	IOBRICK
	emmc4_cfg	SDMMC4_DAT1	SDMMC4_DAT1	IOBRICK
	emmc4_cfg	SDMMC4_DAT2	SDMMC4_DAT2	IOBRICK
	emmc4_cfg	SDMMC4_DAT3	SDMMC4_DAT3	IOBRICK
	emmc4_cfg	SDMMC4_DAT4	SDMMC4_DAT4	IOBRICK
	emmc4_cfg	SDMMC4_DAT5	SDMMC4_DAT5	IOBRICK
	emmc4_cfg	SDMMC4_DAT6	SDMMC4_DAT6	IOBRICK
	emmc4_cfg	SDMMC4_DAT7	SDMMC4_DAT7	IOBRICK
	emmc4_cfg	SDMMC4_DQS	SDMMC4_DQS	IOBRICK

Controller	Pin Configuration	Signal	Ball Name	Pinmux Select
	emmc4_cfg	SDMMC4_COMP	SDMMC4_COMP	IOBRICK

9.5.2.4 SDMMC Clocking

SDMMC controller's clock is sourced from either PLLC4 (Primary source) or PLLP (secondary source) branches. PLLC4 supports spread spectrum (SS) and should be enabled to avoid EMI issues on SD/eMMC interface. PLLC4 VCO runs as per the eMMC part frequency in the platform. PLLP should not be used for eMMC HS400 clocking.

If PLLC4 is shared for SDMMC controllers OR QSPI interface post-boot, it's quite possible that other interfaces (SDMMC1, QSPI, or bridges, etc.) run at compromised (lower than respective engine speed). In such cases, the clocking policy is that eMMC gets the priority because of storage performance.

9.5.2.4.1 Clock Frequency Requirements for SDMMC Operating Modes

The SD and eMMC target IO frequency requirements are given in the tables below.

Table 9.15 SD Modes

Speed Mode	Signal Voltage	I/O Frequency (MHz)	Bus Width	Max Throughput (MB/S)
Default Speed	3.3	25	1,4	12.5
High Speed	3.3	50	1,4	25
SDR12	1.8	25	1,4	12.5
SDR25	1.8	50	1,4	25
SDR50	1.8	100	1,4	50
DDR50	1.8	50	1,4	50
SDR104	1.8	208	1,4	104

Table 9.16 eSD Modes

Speed Mode	Signal Voltage	I/O Frequency (MHz)	Bus Width	Max Throughput (MB/S)
Default Speed	3.3	25	1,4,8	12.5
High Speed	3.3	50	1,4,8	25

Table 9.17 SDIO Modes

Speed Mode	I/O Frequency (MHz)	Bus Width	Max Throughput (Mbytes/S)
Default Speed	25	1,4	12.5
High Speed	50	1,4	25
SDR12	25	1,4	12.5
SDR25	50	1,4	25
SDR50	100	1,4	50
DDR50	50	1,4	50
SDR104	208	1,4	104

Table 9.18 eMMC Modes

Speed Mode	I/O Frequency (MHz)	Bus Width	Max Throughput (Mbytes/S)	MMC rev
Legacy Speed	26	1,4,8	26 MB	MMC4.3
High Speed SDR	52	1,4,8	52 MB	
High Speed DDR	52	4,8	104	MMC4.4
HS200 (SDR)	200	4,8	200	MMC4.51
HS400(DDR)	200	8	400	eMMC5.0

9.5.2.5 Interface to Memory Controller

The SDMMC controller is a direct MC client. It uses 256-bit MCCIF to meet target bandwidth requirements.

9.5.2.6 Address Width

In order to access all of DRAM, SDMMC must generate addresses larger than 32-bit. Therefore a 64-bit system address needs to be advertised in the Host Control 2 Register.

The SYSTEM_ADDRESS_Reg used for SDMA is 32-bit wide. During 64-bit system addressing, the ADMA System Address Register would be used as the pointer to System Memory as mentioned in the SDHost4.2 specification.

Table 9.19 DMA Support

DMA	SoC
SDMA	Supported (32/64 bit)
ADMA1	Deprecated
ADMA2 32 bit	Supported
ADMA2 64 bit	Supported
ADMA3	Supported

SDMA: Single DMA. A large contiguous memory space may not be available in the virtual memory system. To perform long SDMA transfers, the SDMA System Address register requires updates at every system memory boundary. This requires interrupting the host CPU at the end of contiguous boundary so the next contiguous memory DMA can be programmed. SDHOST4.2 allows the SDMA to be 64 bit (previously 32 bit). When SDMA is used with 64-bit addressing, the ADMA System Address Register is used as the SDMA System Address Register.

ADMA: Advanced DMA – ADMA2 and ADMA3. The Host Driver builds a descriptor in the memory before commencing DMA. This is a scatter gather DMA operation and does not require interrupting the CPU after every transfer. Both ADMA2 and ADMA3 adopt the scatter gather DMA algorithm so that a higher data transfer speed is available. The Host Driver can program a list of data transfers between system memory and SD card to the Descriptor Table. ADMA2 performs a read / write SD command operation at the time. ADMA3 can program multiple read/write SD commands operation in a Descriptor Table. ADMA3 is suitable to perform very large data transfer.

ADMA1 is not supported from SDHOST3.0 onwards. It limited data transfers of only 4 KB aligned data in System Memory and ADMA2 removed this limitation. Furthermore, ADMA2 can be used to address the 32 bit or 64 bit (40 bit for this SoC) system memory. ADMA shall refer to ADMA2 in the document since ADMA1 is deprecated.

9.5.2.7 SMMU Translation

While SDMMC can support operation with SMMU translation off and using 64-bit addressing only, SMMU translation will normally be used. Also, SMMU must be used to access all of DRAM using a 32-bit DMA address. SMMU translates this address into DRAM Physical Address (PA).

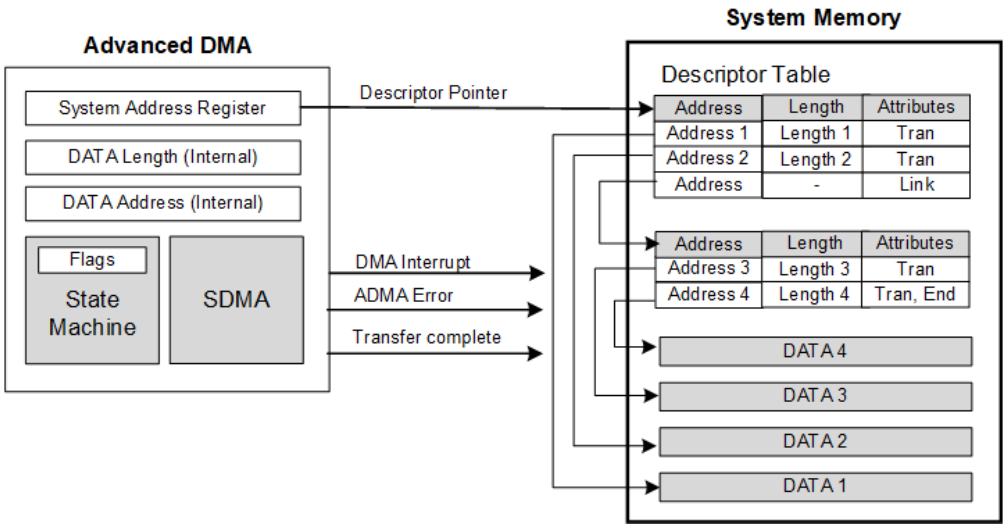
9.5.2.8 ADMA2 64 Bit

To support larger addressing requirements, 64 bit ADMA is added. This requires adding the ADMA_SYSTEM_ADDRESS register [63:32] at address offset 0x5Ch.

1. Updating the Host Control 2 register to indicate support for 64-bit addressing.
2. Updating the DMA select field with the new encoding in SDHOST4.2

The host driver shall program the ADMA_SYSTEM_ADDRESS register via two consecutive writes. The descriptors built by the Host Driver also need to be updated to support 64-bit address (3/4 words instead of 2).

Figure 9.20 Block Diagram for ADMA2



9.5.2.9 Descriptors for 64-bit Addressing

The 96-bit descriptor in SDHOST3.0 can be implemented in two ways:

1. Case 1: Multiple 96-bit descriptors are aligned contiguously without 32-bit padding - the descriptor start address should be 32 bit aligned.
2. Case 2: Multiple 96-bit descriptors are aligned with 32-bit padding - the descriptor start address should be 64 bit aligned.

In host4.1, Case 1 is implemented when the host driver chooses to run in sdhost3.0 compatibility mode in the 64-bit addressing system. Case 2 is implemented (128-bit descriptor) when the host driver chooses to run in sdhost4.0 mode in the 64-bit addressing system.

Possible cases are:
Addressing mode version select descriptor size addr_alignment

- 1> 32bit 3.0 64bit 32bit boundary
- 2> 32bit 4.0 64bit 32bit boundary
- 3> 64bit 3.0 96bit 32bit boundary
- 4> 64bit 4.0 128bit 64bit boundary

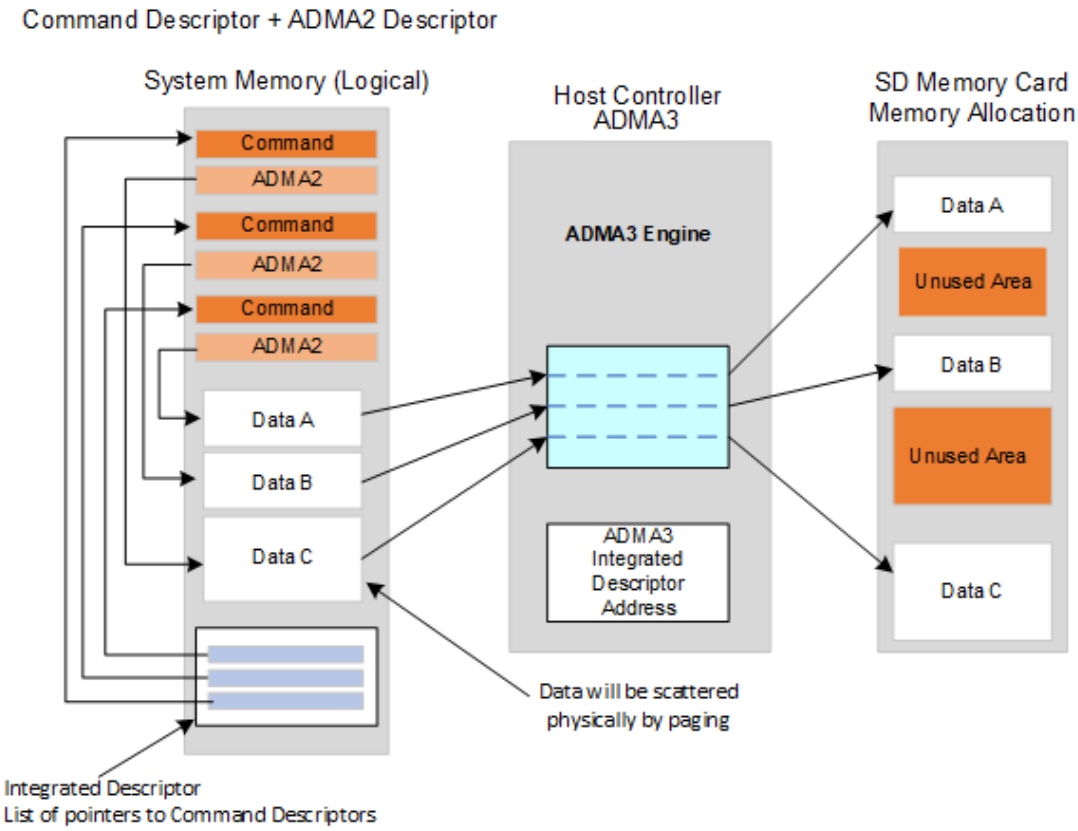
9.5.2.10 ADMA3

SDHOST4.2 mentions a new DMA called ADMA3. ACT0 bit (=1) in the Attribute field of the descriptor distinguishes ADMA2 descriptor from ADMA3 descriptor. ADMA2 can gather data from fragmented System Memory and store it into a contiguous location in the SD Flash. For a read, the SD Flash can be read contiguously and the data can be scattered into the System Memory.

ADMA3 supports scatter gather even in the Flash. This requires issuing different commands to the SD Device without CPU intervention.

1. The Host Driver builds an Integrated Descriptor in the System memory. {ACT2, ACT1, ACT0} = 3'b111 in the attribute field indicate an Integrated Descriptor. The address of the Integrated Descriptor is pointed to by ADMA3 Integrated System Address Register (090h).
2. The Integrated Descriptor entries point to Partial Descriptors location in the System Memory.
3. Each Partial Descriptor is a Command + ADMA Descriptor pair.
4. Command Descriptor has ACT2 = 0 and ACT0 = 1. The command descriptor is programmed with the Block Size and the block Count Value and the Command Mode Value.
5. The ADMA Descriptor is similar to the ADMA2 descriptor.

Figure 9.21 Block Diagram of ADMA3



9.5.2.11 Register Access

SDMMC uses a 32-bit APB Slave interface for register access from the Control Plane.

SDMMC has 64 KB register address space even though the Host spec defines only 512B register space. The spec defined registers are in first 512B space of AMAP allocated and remaining space is used for vendor specific registers which could be pad/IO brick control registers, CQE registers, etc.

Table 9.20 Address Space for the SDMMC Controllers

Controller	Offsets
SDMMC1	0340:0000 – 0340:FFFF
SDMMC4	0346:0000 – 0346:FFFF

9.5.2.12 Non-DMA Mode of Transfer (PIO mode)

For the Non-DMA Mode of Transfer to the Card, the Host Driver needs to write into the Buffer Data Port Register. The register access latency is in the range 400-500 ns. The theoretical maximum throughput with this latency is 8-10 MB per second. This does not meet the theoretical maximum for SD High-Speed bandwidth 25 MBps or MMC 52 MBps, but PIO/Non-DMA mode is expected to give such underperformance.

Boot uses SDMA/ADMA2 and is not affected by the register access latency during actual data transfer. This is also the latency for programming the registers to initiate transfers. This should not be a bottleneck as this latency affects all modules and not just SDMMC in particular.

9.5.2.13 Standard Driver Support

The SDMMC supports both Windows and Linux (Android) Drivers. Windows drivers are not open source and hence have stricter requirements to meet, i.e., they do not support chip-specific routines. For Android Drivers, the standard Linux MMC Driver (Linux Kernel driver which supports all SD, SDHC, SDXC, eMMC, MMC, SDIO cards) is modified to support NVIDIA SDMMC controller.

The Linux kernel MMC driver has three layers: Card, Core, and Host. The Host is open source GPL. For platform specific changes, they are in the Host Layer. Changes are done to Card and Core when needed. However, since card and core layers are common to all platforms, this makes the drivers non-standard.

Each platform (NVIDIA and non-NVIDIA) has its own platform specific implementation in the host layer. Card and core layers are common to all platforms. There is a defconfig file in the kernel which enables chip-specific CONFIG variables for MMC and disables other platforms so that during

compile time the NV specific driver is selected. Therefore, when the SD card is inserted on running Android phone/tablet/clamshell, the default driver is NV which identifies the card and gets mounted.

Note: Each SDMMC controller has a specific use case and requires a separate driver to control it. This means that each SDMMC controller has a unique Software ID in the MC controller.

9.5.2.13.1 SDMMC Drivers

The MMC driver from previous SoCs can be used.

Table 9.21 Drivers Supported on SDMMC Controllers

Controller	Drivers	Vendor	Devices Supported
SDMMC1	SDHCI4.2 Standard Drivers SDHCI4.2 NVIDIA Drivers	Windows on ARM Android	SD4.2/SDIO4.1
SDMMC4	NVIDIA Boot Drivers eMMC5.1 driver with CQHCI		eMMC5.1

9.5.3 SDMMC Programming Guidelines

This section assumes a SD Host driver complying with the SD Specifications' Part A2, SD Host Control Standard Specification 4.0, is already available. The following subsections detail the necessary changes required for fully-featured SD (and eMMC boot mode) operation. Refer to the table below for the suffix of the register associated with each controller.

Table 9.22 SDMMC Controller to Register Name Map

SDMMC Controller	Controller Register Prefix
SDMMC1	SDMMCA_
SDMMC4	SDMMCAB_

9.5.3.1 Clocks and Reset

CAR module registers need to be programmed to supply root clocks (first level clocks) and reset to SDMMC controllers.

- It makes sense to operate the SDMMC logic at I/O clock frequency. This requires generating the clock via the CAR itself and not relying on the SDMMC internal divisors. It is possible for SDMMC, however, to operate at 104 MHz, divide its clock internally by two and operate the I/O at 52 MHz. The initializing frequency (up to 400 kHz) is provided to the card using the internal divider of the SDMMC (current max divisor is 2046 as per SD host spec). This is software dependent.

- In DDR modes, Software does not use the fractional clock divider as it affects DCD which would in turn lead to timing issues.
- In HIGH SPEED EMMC DDR/SD card DDR50 modes, the SDMMC internal clock divider is set to ‘2’ [register value would be ‘1’].
- In HS400 mode, SDMMC internal clock divider is set to ‘1’ [register value would be ‘0’].
- Use PLLC4 with SSC enabled to clock the SD/eMMC interface in all modes to avoid EMI issues.

Table 9.23 Address Map for SDMMC Registers

Registers	Address map
SDMMC1 controller	0x0340:0000 – 0x0340:FFFF
SDMMC4 controller	0x0346:0000 – 0x0346:FFFF
SDMMC1 pinmux and pad control	0x0243:8000 – 0x0243:8FFFF
SDMMC4 pinmux and pad control	0x0243:6000 – 0x0243:6FFFF
PMC	0x0C36:0000 – 0x0C36:FFFF
Clocks and Reset	SDMMC1 : 0x2019:0000 – 0x2019:FFFF SDMMC1 : 0x201B:0000 – 0x201B:FFFF

9.5.3.2 Initialization

As a part of initialization, software enables the pads, the pinmux, the SDMMC controller clocks by programming the PLLs, CLK dividers, and programming the Host controller vendor registers, device initialization and tuning. Once initialization is done, the SDMMC controller is ready for executing the data transfers. The APB Slave interface provides access to the configuration registers. Each SDMMC controller is functionally the same but differs in supported data bus and speed modes. Each controller has a separate register set for programming the CLK sources, pad controls, and PINMUX settings. Full details of each step involved in this initialization process per controller is given in next sections.

9.5.3.3 SDMMC1 Initialization Sequence

This section provides the SDMMC1 controller initialization sequence which should be followed by the software driver before doing any CMD/DAT transfers to external SD/SDIO device.

9.5.3.3.1 Power ON

1. Make sure SDMMC1 controller is powered up by programming the external PMU/PMIC (platform specific) to set VDD_CORE voltage.
2. Refer to the PMU/PMIC datasheet for programming details.

9.5.3.3.2 Program Clocks and Reset

Refer to the Clock and Reset Controller (CAR) chapter in the TRM for full details of PLLs programming which are used for SDMMC1 clocking.

1. Set SDMMC1 Reset:
   a. Keep SDMMC1 in reset by writing CLK_RST_CONTROLLER_RST_DEV_SDMMC1_SET_0_ SET_SWR_SDMMC1_RST with ENABLE (0x1).
2. Program the CLK divider and source
   a. Program SDMMC1 core clock divider (CLK_RST_CONTROLLER_CLK_SOURCE_SDMMC1_0_ SDMMC1_CLK_DIVISOR) and PLL source (CLK_RST_CONTROLLER_CLK_SOURCE_SDMMC1_0_ SDMMC1_CLK_SRC) as per below guidelines.
   b. PLLP and PLLC4 are the main CLK sources for SDMMC1 controller.
   c. Software driver programs the SD CLK Frequency Select to obtain the desired SD/SDIO device clock which is derived from SDMMC host clock. Note: this should be done later, after de-asserting reset to the host controller. Follow SD clock supply sequence mentioned in SD Host specification 4.0.
   d. The maximum operating core clock frequency for the SDMMC1 controller is 208 MHz. This value is advertised in SDMMCA_CAPABILITIES_0_ BASE_CLOCK_FREQUENCY. Software may run the SDMMC1 controller host clock at a lower frequency based on the requirement. In such a case, software should set SDMMCA_VENDOR_CLOCK_CNTRL_0_BASE_CLK_FREQ with the host clock frequency value in MHz to advertise the base frequency in SDMMCA_CAPABILITIES_0_BASE_CLOCK_FREQUENCY for the standard SD driver usage. The standard SD Host driver uses this register value to determine the internal SDCLK Frequency Select value to get the desired I/O clock frequency.
   e. Software may run SDMMC1 controller core clock at a lower speed. In such a case, SDMMCA_VENDOR_CLOCK_CNTRL_0_ BASE_CLK_FREQ should be written with core clock frequency value in MHz to advertise base frequency in SDMMCA_CAPABILITIES_0_ BASE_CLOCK_FREQUENCY for standard SD driver usage.'
   f. The initializing frequency up to 400 kHz is provided to the SD/SDIO card using SD CLK Frequency Select (current max divisor 2046 as per SD host spec). This is completely software dependent. Note: There is no SDCLK Frequency Select restriction in SDR modes (DS, HS, SDR12, SDR25, SDR50, SDR104 and HS200).
   g. In HIGH SPEED EMMC DDR/DDR50 modes, the SDMMC1 host clock should be running at 2X frequency than device clock. This requires the SDMMC internal clock divider to be set to ‘2’ [Standard Host Register SDMMCA_SW_RESET_TIMEOUT_CTRL_CLOCK_CONTROL_0_ SDCLK_FREQUENCYSELECT value would be ‘1’]. This is mandatory to make DDR mode work.
3. Program the SDMMC1 timeout clock source and divider
   a. SDMMC1 uses 12 MHz TMCLK which is advertised in Host capabilities register SDMMCA_CAPABILITIES_0_ TIMEOUT_CLOCK_FREQUENCY.
   b. Software should program below registers to provide TMCLK to SDMMC1:

i. CLK_RST_CONTROLLER_CLK_SOURCE_SDMMC_LEGACY_TM_0_ SDMMC_LEGACY_TM_CLK_SRC
ii. CLK_RST_CONTROLLER_CLK_SOURCE_SDMMC_LEGACY_TM_0_ SDMMC_LEGACY_TM_CLK_DIVISOR
iii. PLLP_OUT0 is the recommended TMCLK source for SDMMC1.

4. Enable SDMMC1 core clock and TMCLK by setting below registers.
   a. CLK_RST_CONTROLLER_CLK_OUT_ENB_SDMMC1_0_ CLK_ENB_SDMMC1_ENABLE
   b. CLK_RST_CONTROLLER_CLK_OUT_ENB_SDMMC_LEGACY_TM_0_CLK_ENB_SDMMC_LEGACY_TM_ENABLE
5. Wait for at least 100 SDMMC1 clock (host clock not device clock) cycles time to get reset propagated down the pipe and stabilize internal clocks.
6. Clear SDMMC1 reset by writing CLK_RST_CONTROLLER_RST_DEV_SDMMC1_CLR_0_CLR_SWR_SDMMC1_RST with ENABLE. Enable takes effect immediately.

9.5.3.3.3 Card Detection and Write Protect Status

1. GPIO-based card detection mechanism is supported. Software should program SDMMC1_CD and SDMMC1_WP pinmux registers as per customer pinmux data sheet.
2. SDMMC1 controller gets SDCD and SDWP status from pinmuxed GPIO pins SDMMC1_WP and SDMMC1_CD. Software must program pinmux and CD/WP source appropriately to get SDMMC1 correct status received. Contact NVIDIA if platform requires usage of other general-purpose pins for SDMMC1_CD and SDMMC1_WP.
   a. Program the pinmux for SDMMC1 CD (Card Detect):
   b. Program the pinmux for SDMMC1 WP (Write Protect): (if used in the platform).

9.5.3.3.4 Power ON SD Card or SDIO Device

1. If the removable card is detected using GPIO, power ON SD card (3.3V to start with). No need to wait for the card detection in case of SDIO. SDMMC1 pads can support both 3.3V and 1.8V I/O. Incorrect programming could lead to pad damage.
2. To avoid high leakage and pad damage, set SDMMCA_SDMEMCOMPPADCTRL_0_ PAD_E_INPUT_OR_E_PWRD to 0x1 before turning ON SD card power.
3. Program E_33V for the SDMMC1 pads before turning ON the SD/SDIO card power:

- SDMMC1 pads have both 1.8V and 3.3V power supplies brought up on power ON. No PMIC programming is required to switch between the voltages.
- SDMMC1 pads support both 3.3V and 1.8V IO drivers. Software can select desired IO driver by programming below register.
- Set PMC_IMPL_E_33V_PWR_0_ SDMMC1_HV = 1 to select 3.3 V IO driver in pad.
- Set PMC_IMPL_E_33V_PWR_0_ SDMMC1_HV = 0 to select 1.8 V IO driver in pad.

9.5.3.3.5 PINMUX and Pad Control Programming

SDMMC1 pads are pinmuxed. Software programs the PINMUX registers to select SDMMC1 as the function controlling the pad and to program the pad controls. The pinmux programming per SDMMC1 usage is below as a comprehensive reference, however these settings may be initialized by the platform driver using the PINMUX CONFIG sheet.

1. Program CLK pad pinmux and control settings:
   a. Select 50 Ohm pad driver - PADCTL_SDMMC1_HV_SDMMC1_CLK_0_DRV_TYPE_ DRIVE_2X. Ensure SDMMC1_COMP pad is connected to GND via 100 Ohm resistor on board.
   b. DRV_TYPE_DRIVE2X and 100 Ohm calibration resistor are used to get drive strength codes for 50 Ohm driver.
   c. Both the E_INPUT and E_LPBK of the SDMMC1 CLK pad are enabled to provide loopback CLK for SDMMC1 Receiver. If disabled, reads from SD/eMMC device does not work.
      - PADCTL_SDMMC1_HV_SDMMC1_CLK_0 _E_LPBK = ENABLE
      - PADCTL_SDMMC1_HV_SDMMC1_CLK_0 _E_INPUT = ENABLE
   d. Remove tri-state to enable pad output driver - PADCTL_SDMMC1_HV_SDMMC1_CLK_0_TRISTATE_ PASSTHROUGH
   e. Select NONE - PADCTL_SDMMC1_HV_SDMMC1_CLK_0_PUPD_ NONE
   f. Select SDMMC1 in pinmux - PADCTL_SDMMC1_HV_SDMMC1_CLK_0 _PM_SDMMC1
   g. Other pad control settings in PADCTL_SDMMC1_HV_SDMMC1_CLK_0 register should be left as default values. No need to modify.
2. Program CMD pad pinmux and control settings:
   a. Select 50 Ohm pad driver - PADCTL_SDMMC1_HV_SDMMC1_CMD_0_DRV_TYPE_ DRIVE_2X.
   b. E_INPUT of SDMMC1 CMD pad should be enabled to activate receiver in CMD pad. If disabled, response sent by SD/SDIO is lost.
      - PADCTL_SDMMC1_HV_SDMMC1_CMD_0_E_INPUT_ENABLE
   c. Remove tri-state to enable pad output driver - PADCTL_SDMMC1_HV_SDMMC1_CMD_0_TRISTATE_ PASSTHROUGH
   d. Select PULL_UP - PADCTL_SDMMC1_HV_SDMMC1_CMD_0_PUPD_ PULL_UP
   e. Select SDMMC1 in pinmux - PADCTL_SDMMC1_HV_SDMMC1_CMD_0_PM_SDMMC1
   f. Other pad control settings in PADCTL_SDMMC1_HV_SDMMC1_CMD_0 register should be left as default values. No need to modify.
3. Program DAT0/1/2/3 pad pinmux and control settings:
   a. Select the 50 Ohm driver - PADCTL_SDMMC1_HV_SDMMC1_DATx_0_DRV_TYPE_ DRIVE_2X.
   b. E_INPUT of SDMMC1 DATx pad should be enabled to activate the receiver in the DATx pad. If disabled, data sent by SD/SDIO is lost.
      - PADCTL_SDMMC1_HV_SDMMC1_DATx_0_E_INPUT_ENABLE

c. Remove tri-state to enable pad output driver - PADCTL_SDMMC1_HV_SDMMC1_DATx_0_TRISTATE_ PASSTHROUGH
d. Select PULL_UP - PADCTL_SDMMC1_HV_SDMMC1_DATx_0_PUPD_ PULL_UP
e. Select SDMMC1 in pinmux - PADCTL_SDMMC1_HV_SDMMC1_DATx_0_PM_SDMMC1
f. Other pad control settings in PADCTL_SDMMC1_HV_SDMMC1_DATx_0 register should be left as the default values. Do not modify.
g. Note: x = 0, 1, 2 or 3 – DAT lane number

9.5.3.3.6 Program Pad Control and Vendor Registers

1. Program the registers below before running auto-calibration to prevent CRC errors from occurring during data transfers.
   a. Write 0x1 into bit 19 (SPARE_OUT[3]) of register SDMMCA_IO_SPARE_0.
   b. Write 0x0 into bit 2 (SEL_VREG) of register SDMMCA_VENDOR_IO_TRIM_CNTRL_0.
   c. Refer to the DLL and Inbound Clock Trimmer Power Supply (VREG) Programming for the software sequence to program this register field.
2. Set outbound clock trimmer tap value (works in all speed modes - up to 208 MHz) before doing any cmd/data transfers to device. This is needed to meet interface timing spec. SDMMCA_VENDOR_CLOCK_CNTRL_0_ TRIM_VAL = 0x5
3. Set inbound clock trimmer tap value (works up to 50 MHz – in non-tunable modes such as DS, HS, SDR12, SDR25, and DDR50) before doing any cmd/data transfers to device. Tuning procedure needs to be run in other modes – SDR50 and SDR104. SDMMCA_VENDOR_CLOCK_CNTRL_0_ TAP_VAL = 0xB

9.5.3.3.7 Run Auto-Calibration

The run auto-calibration procedure to required to set the proper drive strength codes in the SDMMC1 pads. This is needed to use the 50 Ohm driver for driving the SD/SDIO interface. Run this procedure before performing the device initialization when the SD card I/O is using 3.3V and the SDIO is using 1.8V. Ensure the device (SD/SDIO/eMMC) clock is turned off during the pad driver calibration to avoid unnecessary voltage swing changes on the CLK pad output.

Software must re-run the auto-calibration immediately after switching to 1.8V I/O from 3.3V I/O and before doing any transfers to SD device.

1. Program drive code offsets when for 3.3V operation before running auto-calibration:
   a. SDMMCA_AUTO_CAL_CONFIG_0_AUTO_CAL_PD_OFFSET = 0x0
   b. SDMMCA_AUTO_CAL_CONFIG_0_AUTO_CAL_PU_OFFSET = 0x0
2. Program drive code offsets for 1.8V operation before running auto-calibration:
   a. SDMMCA_AUTO_CAL_CONFIG_0_AUTO_CAL_PD_OFFSET = 0x0
   b. SDMMCA_AUTO_CAL_CONFIG_0_AUTO_CAL_PU_OFFSET = 0x0
3. Run auto-calibration procedure.

a. Turn off SD/SDIO/eMMC device clock before running calibration to avoid unwanted voltage swing changes on CLK pad output by programming SDMMCA_SW_RESET_TIMEOUT_CTRL_CLOCK_CONTROL_0[SD_CLOCK_EN] = 0x0.
b. Set E_INPUT_OR_E_PWRD = 1 in SDMMCA_SDMEMCOMPPADCTRL_0 register, if not set.
c. Wait for 1 µs after E_INPUT_OR_E_PWRD is enabled.
d. Set AUTO_CAL_START and AUTO_CAL_ENABLE to 1 in SDMMCA_AUTO_CAL_CONFIG_0 register.
e. Wait for 1 µs.
f. Wait for up to 10 ms for AUTO_CAL_ACTIVE in SDMMCA_AUTO_CAL_STATUS_0 register to become 0. If it is not 0 after 10 ms, software can assume auto-calibration has timed out. If it becomes zero within 10 ms, it is assumed that calibration has completed.
g. Clear E_INPUT_OR_E_PWRD to save power.
h. Turn on device clock by programming SDMMCA_SW_RESET_TIMEOUT_CTRL_CLOCK_CONTROL_0 _SD_CLOCK_EN = 0x1. Software should not clear AUTO_CAL_ENABLE in SDMMCA_AUTO_CAL_CONFIG_0 register after calibration. It should be set to 1 always to use calibration codes generated by calibration controller.
i. If the auto-calibration process fails or times out, software should clear SDMMCA_AUTO_CAL_CONFIG_0 [AUTO_CAL_ENABLE] = 0 and program the force calibration codes in the registers below. It is not necessary to program DRVUP_OVR and DRVDN_OVR fields before auto calibration starts.
j. For 3.3V IO:
   - SDMMCA_SDMEMCOMPPADCTRL_0_ COMP_PAD_DRVUP_OVR = 0x7
   - SDMMCA_SDMEMCOMPPADCTRL_0_ COMP_PAD_DRVDN_OVR = 0x7
k. For 1.8V IO:
   - SDMMCA_SDMEMCOMPPADCTRL_0_ COMP_PAD_DRVUP_OVR = 0x6
   - SDMMCA_SDMEMCOMPPADCTRL_0_ COMP_PAD_DRVDN_OVR = 0x7

4. Enable SD device clock by writing into SD Host registers (follow the SD bus power and clock supply sequence described in Section 3.2 of SD Host 4.0 specification). The SD bus power field in the Power Control register should be set to 1 before enabling the SD CLK. Flow chart from spec is given below for quick reference.
5. Issue a software reset before communication with the device. SDMMCA_SW_RESET_TIMEOUT_CTRL_CLOCK_CONTROL_0[SW_RESET_FOR_CMD_LINE] and SDMMCA_SW_RESET_TIMEOUT_CTRL_CLOCK_CONTROL_0[SW_RESET_FOR_DAT_LINE]
6. Do SD/SDIO device initialization as per the sequence mentioned in Section 3.6 of SD Host 4.0 specification published by SD Card Association.
7. Points to remember while doing device initialization:
   a. During the device initialization, when I/O power is switched to 1.8V from 3.3V, software need to run auto-calibration again to set proper drive strengths before accessing the external device.
   b. During the voltage switching phase, follow the guidelines below to select correct I/O driver in the pad.

i. SDMMC1 pads have both 3.3V and 1.8V supplies brought up on power ON. Programming PMU/PMIC is not required to switch between voltage supplies.
ii. Software should program the register below to select the desired I/O driver.
- Set PMC_IMPL_E_33V_PWR_0_ SDMMC1_HV = 1 to select 3.3 V I/O driver.
- Set PMC_IMPL_E_33V_PWR_0_ SDMMC1_HV = 0 to select 1.8 V I/O driver.

c. Software driver needs to set inbound clock trimmer tap value as per below guidelines to fix sampling point for data coming from device.
d. In non-tunable modes (DS, HS, SDR12, SDR25 and DDR50 – up to 50 MHz), software must program inbound trimmer tap value as mentioned in the Program Pad Control and Vendor Registers.
e. To do data transfers in tunable modes (SDR50 and SDR104), the tuning procedure must be run to fix the sampling point first. During tuning, all interrupts except buffer_read_ready should be disabled. Software should program the register before running tuning procedure:
i. SDMMCA_INTERRUPT_STATUS_ ENABLE_0 = 0x20
ii. SDMMCA_INTERRUPT_SIGNAL_ENABLE_0 = 0x20

8. Host and device are ready for doing data transfers.
9. Follow software programming sequences mentioned in Chapter 3 of SD Host 4.2 specification published by SD Card Association for doing data/CMD transfers.

9.5.3.4 SDMMC4 Initialization Sequence

This section provides SDMMC4 controller initialization sequence which should be followed by the software driver before doing any CMD/DAT transfers to external eMMC device.

9.5.3.4.1 Power ON

1. Make sure the SDMMC4 controller is powered up by programming the external PMU/PMIC (platform specific) to set VDD_CORE voltage.
2. Refer to the PMU/PMIC data sheet and power up sequence section in the TRM for programming details.

9.5.3.4.2 Program Clocks and Reset

Refer to the Clock and Reset Controller (CAR) chapter in this TRM for full details of PLLs programming which are used for SDMMC4 clocking.

1. Set SDMMC4 Reset:
   a. Keep SDMMC4 in reset by writing CLK_RST_CONTROLLER_RST_DEV_SDMMC4_SET_0_ SET_SWR_SDMMC4_RST with ENABLE (0x1).
2. Program CLK divider and source.
   a. Program the SDMMC4 core clock divider (CLK_RST_CONTROLLER_CLK_SOURCE_SDMMC4_0_ SDMMC4_CLK_DIVISOR) and

PLL source (CLK_RST_CONTROLLER_CLK_SOURCE_SDMMC4_0_ SDMMC4_CLK_SRC) as per below guidelines.

b. PLLP and PLLC4 are the main CLK sources for SDMMC4 controller. PLLC4 is the dedicated PLL for eMMC5.0 HS400 clocking. For eMMC HS400 modes, software should use PLLC4 only since PLLC4 clock path is optimized to meet the HS400 duty cycle distortion specification.

c. Software driver should program SDCLK Frequency select in SD host clock control register to get the desired eMMC device clock which is derived from the SDMMC4 host clock. This should be done later (after deasserting reset). Follow SD clock supply sequence mentioned in SD Host specification 4.0.

d. The maximum operating host clock frequency for SDMMC4 controller is 208 MHz. By default, 208 MHz is advertised in SDMMCAB_CAPABILITIES_0[BASE_CLOCK_FREQUENCY]. Software must program SDMMCAB_VENDOR_CLOCK_CNTRL_0[BASE_CLK_FREQ] to 200 to advertise 200 MHz support in CAPABILITIES field. Also software could choose to run SDMMC4 controller host clock at a different and lower than 200 MHz frequency based on the requirement. In such a case, software can set SDMMCAB_VENDOR_CLOCK_CNTRL_0[BASE_CLK_FREQ] with the host clock frequency value in MHz to the advertise base frequency in SDMMCAB_CAPABILITIES_0[BASE_CLOCK_FREQUENCY] for standard SD driver usage. Standard SD Host driver uses this register value to determine the SD CLK Frequency Select value to get the desired I/O clock frequency.

e. The initializing frequency up to 400 kHz is provided to eMMC device using the SDCLK Frequency select in SD host clock control register (current maximum divisor 2046 as per SD host specification). This is completely software dependent.

f. In HIGH SPEED EMMC DDR mode, SDMMC4 core should be run at 2X frequency than I/O. This requires SDMMC internal clock divider should be set to ‘2’ [Standard Host Register SDMMCAB_SW_RESET_TIMEOUT_CTRL_CLOCK_CONTROL_0_ SDCLK_FREQUENCYSELECT value would be ‘1’]. This is mandatory to make DDR50/52 mode work.

g. This is DDR mode only but has different timing and protocol compared to DDR50/52 mode. In this mode, host clock frequency should match device clock frequency. This is a design requirement. There is no specification compliance issue as this mode is specific to eMMC devices and not defined in the SD host specification.

h. Note: There is no SDCLK Frequency select restriction in SDR modes (DS, HS, SDR12, SDR25, SDR50, SDR104 and HS200).

3. Program the SDMMC4 timeout clock source and divider:

a. SDMMC4 uses 12 MHz TMCLK which is advertised in the Host capabilities register - SDMMCAB_CAPABILITIES_0_ TIMEOUT_CLOCK_FREQUENCY.

b. Software should program the registers below to provide TMCLK to SDMMC4:

- CLK_RST_CONTROLLER_CLK_SOURCE_SDMMC_LEGACY_TM_0_ SDMMC_LEGACY_TM_CLK_SRC
- CLK_RST_CONTROLLER_CLK_SOURCE_SDMMC_LEGACY_TM_0_ SDMMC_LEGACY_TM_CLK_DIVISOR

c. PLLP_OUT0 is the recommended TMCLK source for SDMMC4.

4. Enable the SDMMC4 core clock and TMCLK by setting the registers below.
   a. CLK_RST_CONTROLLER_CLK_OUT_ENB_SDMMC4_0_ CLK_ENB_SDMMC4_ENABLE
   b. CLK_RST_CONTROLLER_CLK_OUT_ENB_SDMMC_LEGACY_TM_0_CLK_ENB_SDMMC_LEGACY_TM_ENABLE
5. Wait for at least 100 SDMMC4 host clock cycles time to have reset propagated down the pipe and to get internal clocks stabilized.
6. Clear SDMMC4 reset by writing CLK_RST_CONTROLLER_RST_DEV_L_CLR_0_CLR_SDMMC4_RST with ENABLE. Enable takes effect immediately.

9.5.3.4.3 Power ON eMMC or SDIO Device

1. Power ON eMMC (1.8V I/O). Do not wait for card detection as it is an embedded device.

9.5.3.4.4 PINMUX and eMMC IOBRICK Control Programming

1. SDMMC4 uses IOBRICK not loose pads which is not pinmuxed. No PINMUX programming is needed.
2. E_INPUT and E_DEEP_LPBK for SDMMC4 CLK pin should be enabled to provide loopback CLK for SDMMC4 Rx. If disabled, reads from SD/eMMC device does not work.
   a. PADCTL_EMMC_EMMC_PAD_CLK_0_E_DEEP_LPBK = ENABLE
   b. PADCTL_EMMC_EMMC_PAD_CLK_0_E_INPUT = ENABLE
3. Program below register fields in the SDMMC_SDMEMCOMPPADCTRL_0 register.
   a. SDMMCAB_SDMEMCOMPPADCTRL_0_ COMP_PAD_DRVUP_OVR = 0xA
   b. SDMMCAB_SDMEMCOMPPADCTRL_0_ COMP_PAD_DRVDN_OVR = 0xA
4. Select 50 ohm driver:
   a. Ensure that the SDMMC4_COMP pad is connected to GND via the 100 ohm resistor on board.
   b. Make sure all DRV_TYPE fields in PADCTL_EMMC_EMMC_PAD_CLK/CMD/DATx_0 are not updated by software. Software should use default values (DRV_TYPE_2X) only.
5. Pull Up/Pull Down values in PADCTL_EMMC_EMMC_PAD_CLK/CMD/DATx_0 should match with default values. No updates are needed.
6. Pull Up/Pull Down (PUPD) field in PADCTL_EMMC_EMMC_PAD_DQS_0 register should be set to NONE for HS400 supported eMMC devices as device maintains PD on DQS line. No PU/PD required from Host side. If eMMC4.51 devices (which do not support HS400) are used, keep PUPD value at PD which is default value. Note: Boot ROM also keeps PUPD as PD as Boot ROM does not use HS400 mode.

9.5.3.4.5 Program Host Vendor Registers

1. Program below registers before running auto-calibration to prevent CRC errors during data transfers done later.
   a. Write 0x1 into bit 19 (SPARE_OUT[3]) of register SDMMCAB_IO_SPARE_0.

   b. Write 0x0 into bit 2 (SEL_VREG) of register SDMMCAB_VENDOR_IO_TRIM_CNTRL_0. Refer to the DLL and Inbound Clock Trimmer Power Supply (VREG) Programming for the software sequence to program this register field.
2. Set the outbound clock trimmer tap value (works in all speed modes - up to 200 MHz) before doing any CMD/data transfers to device. This is needed to meet interface timing specification.
   a. SDMMCAB_VENDOR_CLOCK_CNTRL_0_TRIM_VAL = 0x5.
3. Set the internal DQS trimmer tap value before any CMD/data transfers to device in HS400 mode. This is needed to meet interface timing specification.
   a. SDMMCAB_VENDOR_CAP_OVERRIDES_0_ DQS_TRIM_VAL = 40 (decimal)
4. Set inbound clock trimmer tap value (works up to 50 MHz – in non-tunable modes such as SDR@26, SDR@52 and HIGH SPEED EMMC DDR) before doing any CMD/data transfers to device. Tuning procedure needs to be run in other modes – HS200 and HS400.
   a. SDMMCAB_VENDOR_CLOCK_CNTRL_0_ TAP_VAL = 0x9

9.5.3.4.6 Run Auto-Calibration

Run auto-calibration procedure to set proper drive strength codes for SDMMC4 IOBRICK. This is needed to use 50 ohm driver for driving eMMC interface. This procedure should be run before doing device initialization. Ensure that device clock (not host clock) is turned off during pad driver calibration.

1. Program drive code offsets for 1.8V operation before running auto-calibration:
   a. SDMMCAB_AUTO_CAL_CONFIG_0_AUTO_CAL_PD_OFFSET = 0x0
   b. SDMMCAB_AUTO_CAL_CONFIG_0_AUTO_CAL_PU_OFFSET = 0x0
2. Run auto-calibration procedure.
   a. Turn off the SD/SDIO/eMMC device clock before running the calibration to avoid unwanted voltage swing changes on CLK pad output.
   b. Set E_INPUT_OR_E_PWRD = 1 in SDMMCAB_SDMEMCOMPPADCTRL_0 register, if not set.
   c. Wait for 1 µs after E_INPUT_OR_E_PWRD is enabled.
   d. Set AUTO_CAL_START and AUTO_CAL_ENABLE to 1 in SDMMCAB_AUTO_CAL_CONFIG_0 register.
   e. Wait for 1 µs.
   f. Wait for up to 10 ms for AUTO_CAL_ACTIVE in SDMMCAB_AUTO_CAL_STATUS_0 register to become 0. If it is not 0 after 10ms, software assumes auto-calibration has timed out. If it becomes zero within 10 ms, calibration has completed.
   g. Clear E_INPUT_OR_E_PWRD to save power. Turn on device clock.
   h. Software must not clear AUTO_CAL_ENABLE in SDMMCAB_AUTO_CAL_CONFIG_0 register after calibration. Set it to 1 always to use the calibration codes generated by the calibration controller.
   i. If auto-calibration process fails or timeout, software should clear SDMMCAB_AUTO_CAL_CONFIG_0 [AUTO_CAL_ENABLE] = 0 and program the force

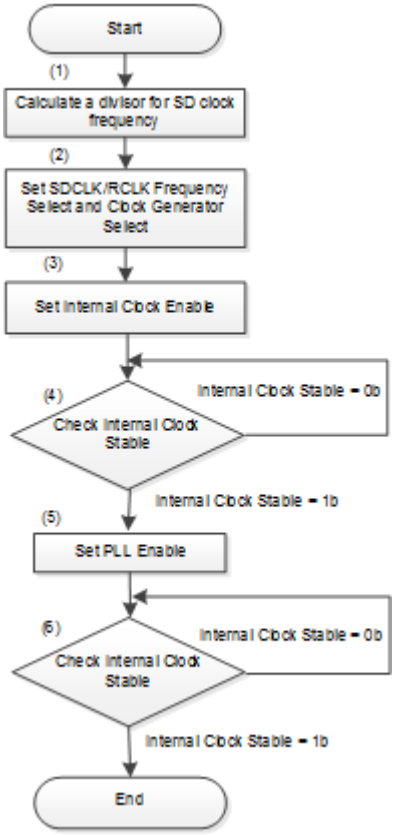
calibration codes in the registers below. It is not necessary to program DRVUP_OVR and DRVDN_OVR fields before auto calibration starts

i. SDMMCAB_SDMEMCOMPPADCTRL_0_ COMP_PAD_DRVUP_OVR = 0xA

ii. SDMMCAB_SDMEMCOMPPADCTRL_0_ COMP_PAD_DRVDN_OVR = 0xA

j. Enable eMMC device clock by writing into SD Host registers (should follow SD bus power and Clock supply sequence described in section 3.2 of SD Host 4.0 specification). Flow chart from spec is given below for quick reference. SD Bus Power register field in Power Control register should be set to 1 to enable SD CLK. See SD Clock Supply Sequence flowchart below.

Figure 9.22 SD Clock Supply Sequence



k. Do eMMC device initialization as per the sequence mentioned in eMMC5.0 specification published by JEDEC.

l. Run DLL calibration immediately after setting HS400 mode frequency, if software wants to use HS400 mode for doing data transfers to/from eMMC.

m. Software driver needs to set inbound clock trimmer tap value as per below guidelines to fix sampling point for data coming from device.

i. In non-tunable modes (LEGACY MMC SDR, HIGH SPEED EMMC SDR, and HIGH SPEED EMMC DDR – up to 52 MHz), software must program inbound trimmer tap value.

ii. To do data transfers in tunable modes (HS200 and HS400), tuning procedure must be run to fix sampling point first. disabled. Software should program below register before running tuning procedure.
   - SDMMCAB_INTERRUPT_STATUS_ENABLE_0 = 0x20
   - SDMMCAB_INTERRUPT_SIGNAL_ENABLE_0 = 0x20

n. Host and device are ready for doing a data transfers.
o. Follow software programming sequences mentioned in Chapter 3 of SD Host 4.0 specification published by SDA for doing data/CMD transfers.

9.5.3.5 General Guidelines

These settings are applicable for all SDMMC controllers in use.

1. Do not access SDMMCxx registers during reset. This causes system to hang.
2. Enable SDR50 mode tuning support in SD/MMC controller by setting SDMMCxx_VENDOR_CLOCK_CNTRL_0_SDR50_TUNING_OVERRIDE. This enables SDR50 mode tuning circuit in Host controller.
3. Enable HS200 mode by setting UHS_MODE_SEL to SDR104 in HOST_CONTROL_2 (offset 03Eh) register. This field is used for SDR104 by standard Host register specification. Reusing the same for HS200 mode as there is no standard specification for eMMC Host.
4. Enable HS400 (HS400 or HS667 or HS533 for eMMC5.0) support by setting UHS_MODE_SEL to HS400 in HOST_CONTROL_2 (offset 03Eh) register. This field is reserved as per standard Host register specification. But using the same as HS400 mode enable as there is no standard specification for eMMC Host.
5. Fx event functionality is defined only for SD cards. That is 'bit 14' of R1 response may be set by SD device to indicate Fx event to the Host. This functionality does not exist for eMMC. The software should disable FX_EVENT interrupt in SDMMC_INTERRUPT_STATUS_ENABLE_0 register when SD/MMC controller is used for eMMC. This is required to avoid receiving false interrupt if eMMC sends bit 14 set in R1 response. Note there is no device selection bit defined in host spec to differentiate between SD and eMMC.
6. Set SDMMCxx_VENDOR_CLOCK_CNTRL_0_PADPIPE_CLKEN_OVERRIDE with NORMAL to clamp clock.
7. Write '1' into bit 19 (SPARE_OUT[3]) of register SDMMCxx_IO_SPARE_0.
8. Write '0' into bit 2 (SEL_VREG) of register SDMMCxx_VENDOR_IO_TRIM_CNTRL_0.During tuning, all interrupts except buffer_read_ready should be disabled. The software should program the register below before running tuning procedure.
   - SDMMCxx_INTERRUPT_STATUS_ENABLE_0 = 0x20

   Enable second level clock gating by setting below values:
   - SDMMC_VENDOR_CLOCK_CNTRL_x_LEGACY_CLKEN_OVERRIDE_NORMAL
   - SDMMC_VENDOR_MISC_CNTRL2_0_ ADMA3_CLKEN_OVERRIDE_NORMAL
   - SDMMC_VENDOR_MISC_CNTRL2_0_ CQE_CLKEN_OVERRIDE_NORMAL
9. Periodic calibration is not required by any of the SDMMC pads/IOBRICK

10. SDMMCxx_AUTO_CMD12_ERR_STATUS_0_ VOLT_18_EN is set by the host driver when switching to 1.8 V modes and is cleared when switching back to 3.3 V mode (which includes the 1.8 V signaling fail case).

9.5.3.5.1 Errata

1. In the case of auto CMD12 after CMD25, the command inhibit DAT goes LOW while the device busy is active. This is a hardware limitation where the logic is looking at only the data present select bit and overlooking the R1b response from CMD12 while deasserting the command inhibit DAT line. The SDMMC driver should look at the transfer complete interrupt before issuing a new command.
2. As per the SD host spec, the command inhibit cmd line should be deasserted at the end bit of CMD response. In the case of card abort doing card write, it has been observed that the command inhibit cmd line stays high until the device deasserts the card busy.The driver looks at the command complete interrupt hence this issue with the command inhibit cmd line doesn't affect the functionality. Also, currently the driver doesn't support non-data cmd issue while the card abort is in progress. Hence this doesn't affect any use case currently.
3. A standard SD driver looks at the command inhibit cmd line to issue a non-data cmd after the cmd 12 response and does not issue it until it goes LOW. So we don't run into any card errors or functional issues. But this causes an unnecessary delay as the new non-data cmd can be issued only after the card busy deassertion in this case.

9.5.3.6 ADMA2

9.5.3.6.1 64-bit Addressing

The host driver enables the 64-bit addressing and programs the ADMA_SYSTEM_ADDRESS register via the two consecutive writes with bits [63:40]='h0. Bits [39:0] would appear on the address bus. MC, in turn, works with the 40 bit translated address with the SMMU disabled.

9.5.3.7 ADMA3

9.5.3.7.1 Issuing Subcommands During ADMA3 Operation

When ADMA3 is enabled, the CMD and DAT line control is owned by ADMA3 engine. Software cannot issue commands on its own. All data bearing commands (CMD_with_DAT) and non-data commands (CMD_wo_DAT) can be issued only via CMD descriptors. Sometimes, software may need to issue status read commands to device (e.g., CMD13) during data transfers. ADMA3 specification allows it by using subcommands (refer to Part A2 SD Host Controller Specification Ver4.20 Final). Subcommand (CMD_wo_DAT) can be issued only after pausing the ADMA3 operation at the nearest block gap by issuing stop-at-block-gap request. Once sub-command is executed by Host, software should issue continue request to resume ADMA3 operation.

The software driver must not disable SDMMCAB_CMD_XFER_MODE_0_ DMA_EN bit when issuing sub-commands during ADMA3 operation since DMA operation is paused but not completed and needs to be resumed after completing sub-command execution.

This is only valid for sub-commands issued during ADMA3 operation but not for non-data commands issued when ADMA3 is not active. The software driver should always disable DMA while issuing CMD_wo_DAT commands as they don't trigger any data transfer.

If software tries to write into the following registers while ADMA3 is in progress (except during stop-at-blockgap), it results in an APB slave error.

- SDMMCxx_SYSTEM_ADDRESS_0
- SDMMCxx_BLOCK_SIZE_BLOCK_COUNT_0
- SDMMCxx_ARGUMENT_0
- SDMMCxx_CMD_XFER_MODE_0
- SDMMCxx_ADMA_SYSTEM_ADDRESS_0
- SDMMCxx_UPPER_ADMA_SYSTEM_ADDRESS_0

9.5.3.7.2 Data Transfers Using ADMA3

Refer to Section 3.13.7 in the Part A2 SD Host Controller Specification Ver4.2.

9.5.3.7.3 CQE

The software should program the registers below to enable CQE to use CQ feature of eMMC. Note all these register bits should set at the same time in multiple register write operations in same sequence. There should not be any other register writes in between.

1. SDMMCAB_AUTO_CMD12_ERR_STATUS_0_HOST_VERSION_4_EN = ENABLE
2. SDMMCAB_POWER_CONTROL_HOST_0_DMA_SELECT = ADMA3_CQE
3. SDMMCAB_CMD_XFER_MODE_0_ DMA_EN = ENABLE
4. SDMMCAB_CQE_CQCFG_0_CQ_EN = 0x1

9.5.3.7.4 CQE Interrupt Coalescing

The software must choose either CQIC timer (INTR_COAL_TIMEOUT_VAL) based interrupt or CQIC counter (INTR_COAL_CNTR_THRESHOLD) based interrupt when Interrupt coalescing enabled. If both are enabled, timeout won't have any effect.

9.5.3.7.5 CQ RED Interrupt Generation

Response Error Detected interrupt is generated whenever any error bit is set in device status field in R1/R1B responses. The software must use CQRMEM (Response Mode Error Mask) register to configure which device status bits to trigger RED interrupt. Note that software must enable this

mask/feature only for the CMDS with R1/R1B responses. The software sees spurious RED interrupt, if RED interrupt and mask are enabled for CMDs with other response types.

Hardware controller cannot mask RED generation for non R1/R1B responses on its own as it doesn't have any knowledge about the response type of the CMD issued by software. The software writes CMD packet image into Host registers and sets response length only. It is not possible to identify R1/R1B responses just based on this information as responses vary from device to device.

9.5.3.7.6 CQ Software Sequences

Refer to section B.6 in the eMMC-5.1-JESD84-B51.pdf.

If software tries to write into the following registers while CQE is in progress (except during halt), it results in an APB slave error.

- SDMMCAB_SYSTEM_ADDRESS_0
- SDMMCAB_BLOCK_SIZE_BLOCK_COUNT_0
- SDMMCAB_ARGUMENT_0
- SDMMCAB_CMD_XFER_MODE_0
- SDMMCAB_ADMA_SYSTEM_ADDRESS_0
- SDMMCAB_UPPER_ADMA_SYSTEM_ADDRESS_0

9.5.3.8 Programming Guidelines for eMMC HS400 Mode

This section provides HS400 mode programming guidelines. This mode is eMMC specific mode and supported by SDMMC4 controller only.

9.5.3.8.1 HS400 Mode Selection

The software driver community is using SD host specification for developing eMMC driver. SDMMC supports HS400 speed mode for devices which support eMMC5.1 specification and beyond. But SD host specification doesn't have support for these modes. Our SD/eMMC host controller uses RSVD fields of UHS_MODE_SEL to enable HS400 mode which won't break any SD specification compliance.

The software driver should program UHS_MODE_SEL as per below to select appropriate speed modes.

9.5.3.8.2 Host Control2 Register/Auto CMD Error Status Register

UHS_MODE_SEL - UHS Mode Select

1. 000b SDR12/Legacy eMMC Speed
2. 001b SDR25/High Speed eMMC SDR

3. 010b SDR50
4. 011b SDR104 (SD/SDIO)/HS200(eMMC)
5. 100b DDR50 (SD/SDIO)/HIGH SPEED EMMC DDR (eMMC)
6. 101b HS400 (eMMC))
7. 111b UHS2
8. 111b Reserved

9.5.3.8.3 Clock Divider Programming in HS400 Modes

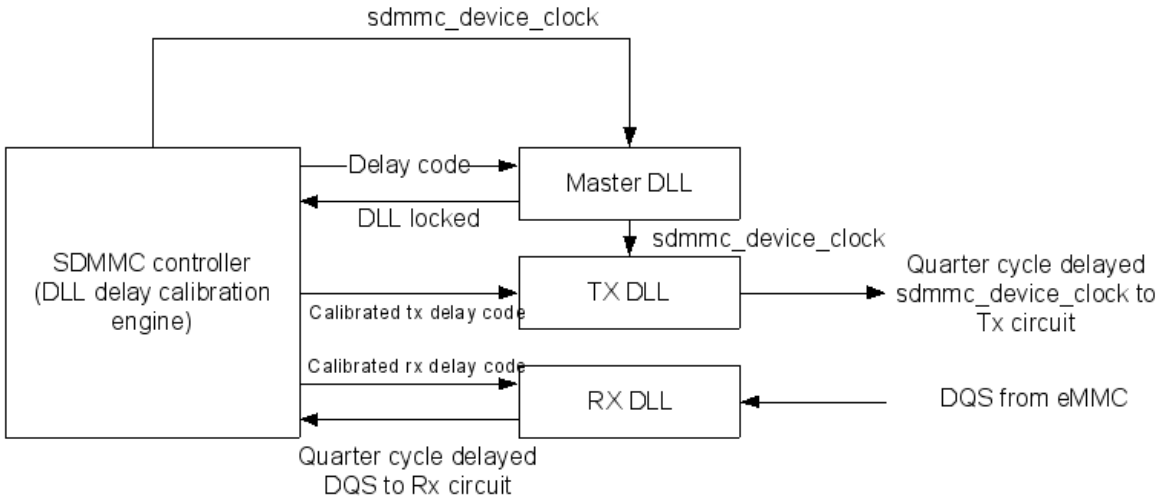
SD_CLK_DIVISOR should be set to '0' in HS400 modes. No other divisors are supported. To get 200 I/O clock frequency, sdmmc_clk should be running at 200 MHz only. That means both I/O and controller should be running at same clock frequency in HS400 mode.

9.5.3.8.4 DLL Calibration

HS400 transmitter and receiver circuit uses a pair of DLLs (Tx DLL and Rx DLL) to delay Tx/Rx clock by quarter cycle. Quarter cycle delay is needed as per our HS400 implementation. DLL is a mixed signal circuit whose delay code should be programmed to get quarter cycle delay. DLL delay code varies based on the process and frequency of clock used. So, we use a Master DLL for calibration to get delay code required for getting quarter cycle delay at given frequency of operation. SDMMC controller has DLL calibration controller which sends delay code in iterative fashion to Master DLL and gets feedback from it. As per the feedback received from Master DLL per applied delay code, the DLL calibration controller determines the delay code needed to get quarter cycle delay. Hardware applies the determined code to both Tx and Rx DLLs and turns off Master DLL which ends the DLL calibration process.

The diagram below shows DLLs present in the SD/MMC subsystem.

Figure 9.23 DLLs in the SDMMC Subsystem



The software needs to calibrate DLL before using eMMC5.0 HS400 mode (before tuning process). Re-calibration is required whenever eMMC I/O clock frequency is changed. This can be done as a part of standard clock frequency change sequence.

Points to note:

1. DLL calibration should work at when the I/O CLK is running at ~200 MHz and above.
2. For I/O CLK frequency is below 180, software should calculate delay code as per below equation and program delay code:
   a. Delay Code = (Delay Code @ 200 MHz) * 200 MHz / Target Frequency
   b. Delay code@ ~200 MHz is determined by running DLL calibration and the same is available for software in the following RO registers.
      i. Tx DLL: SDMMCxx_VENDOR_DLLCAL_CFG_STA_0_ TX_DLY_CODE_ADJ
      ii. Rx DLL: SDMMCxx_VENDOR_DLLCAL_CFG_STA_0_ RX_DLY_CODE_ADJ
      iii. Note there are two DLLs in Rx circuit. So, software needs to program both Tx and Rx DLL registers after calculating delay code as per above equation.
3. Under 85 MHz, DLL delay code may reach its maximum code (128 taps) for FF corner chips. In this case, software should use 128 taps.
4. The software can run DLL calibration at 200 MHz once at boot time, save those values, scale and use them whenever frequency is changed. This is needed if the software wants to do DFS.
5. Program calculated delay code by writing:
   a. Set SDMMCAB_VENDOR_DLL_CTRL0_0_ DLLCAL_BYPASS with 0x1 to bypass DLL calibration.
   b. Set SDMMCAB_VENDOR_DLL_CTRL0_0_ TX_SLV_DLL_DLY_CODE with delay code calculated in above step.

c. Set SDMMCAB_VENDOR_DLL_CTRL0_0_ RX_SLV_DLL_DLY_CODE with delay code calculated in above step.

9.5.3.8.5 Software Programming Sequence for Executing DLL Calibration

1. Set SDMMCAB_VENDOR_IO_TRIM_CNTRL_0_SEL_VREG to 0x0 to select Band Gap VREG to supply DLL. If it is 0x1, set it to 0x0 and follow guidelines before triggering DLL calibration process.
2. Write DLL CAL CFG registers:
   a. SDMMCAB_VENDOR_DLLCAL_CFG_0,
   b. SDMMCAB_VENDOR_DLL_CTRL0_0,
   c. SDMMCAB_VENDOR_DLL_CTRL1_0 and SDMMCAB_VENDOR_DLLCAL_CFG_STA_0 to update various control parameters, if software wants to override default values. **Note:** It is strongly recommended to use default values for DLL calibration since the default settings are verified and characterized.
3. Software can choose either using the software timer (5ms) or END_COUNT for stopping calibration.
4. Program END_COUNT field in register SDMMCAB_VENDOR_DLLCAL_CFG_STA0_0 or maintain the software timer for DLL lock time (5ms). Default value of END_COUNT is sufficient.
5. Set CALIBRATE bit in register SDMMCAB_VENDOR_DLLCAL_CFG_STA0_0 to trigger calibration process.
6. If END_COUNT is set to zero, software should stop calibration by clearing CALIBRATE bit once the software timer expires. Else poll for CALIBRATE = 0 (Hardware clears it once it sees end of calibration condition)
7. Poll for CAL_ACTIVE = 0 to see if calibrated delay code is applied to slave DLLs.
8. Software can exit DLL calibration loop after this and can start tuning process.

9.5.3.8.6 DLL and Inbound Clock Trimmer Power Supply (VREG) Programming

1. Inbound clock trimmer macro is used to delay loopback clock to sample incoming data/ response in SDR50, SDR104, HS200, and HS400 (only response).
2. DLL is used in HS400 receiver to latch incoming data only in HS400 (non-enhanced) and to latch both response and data incoming in HS400 enhanced mode.
3. Delay chain in trimmer and DLL have two power supplies.
   a. Regulated power supply (VREG) from Band Gap (BG) reference circuit which is a constant power supply has almost zero variation with VDD_CORE changes. This is the recommended power supply for both trimmer and DLL.
   b. VDD_CORE/VAUXC is another option. This one is used when BG circuit is turned off in idle state to save power. The software should not use this during transfers as it causes drift in trimmer and DLL delay with VDD_CORE/VAUXC change.
4. Software should use BG power supply to power both trimmer and DLL to make delay produced by internal delay chain invariant to Vcore changes.

a. By default, VDD_CORE/VAUXC (core power supply) is used as trimmer/DLL power supply which saves power by turning off BG regulator supply. The software should set SDMMCxx_VENDOR_IO_TRIM_CNTRL_0_SEL_VREG to 0x0 in all speed modes to select VREG as trimmer power supply which makes trimmer delay independent of VDD_CORE.

5. Software should follow the sequence below for switching power supply from VDD_CORE/ VAUXC to BG.
   a. Make sure SD/MMC is idle – no activity on SD/eMMC interface.
   b. Turn off SD interface clock.
   c. Program SDMMCxx_VENDOR_IO_TRIM_CNTRL_0_SEL_VREG with 0x0 to turn ON BG supply.
   d. BG circuit power up time is ~3 µs. Wait for 3 µs to make sure BG supply is stabilized.
   e. Both DLL and inbound clock trimmer are powered using BG supply and their output could glitch with this power supply switching to/from BG from/to VAUXC/VDD_CORE. This glitch can happen irrespective of input clock/DQS state which would cause disturbance in downstream Rx logic which in turn affects data transfers later on.
   f. To prevent errors on interface due to DLL/trimmer output glitch, the software driver should reset host controller by issuing both SW_RESET_DAT and SW_RESET_CMD (sequence should be followed as per SD host specification) before starting any new transfers.
6. Software can choose to switch back to VAUXC from BG to turn OFF Band gap regulator to save power when eMMC/SD interface is idle. But software should switch back to BG before starting new transfers and follow above sequence for VAUXC to BG switching.

Notes:

1. DLL output glitch (not trimmer glitch) can be avoided by following below steps:
   a. Program below register fields first before switching power supply from/to BG to/from VAUXC.
      i. SDMMCAB_VENDOR_DLLCAL_CFG_STA_0SLV_DLL_CLK_OUT_DIS_OVERRIDE_EN = 0x1
      ii. SDMMCAB_VENDOR_DLLCAL_CFG_STA_0SLV_DLL_CLK_OUT_DIS = 0x1
   b. Wait for 32 SDMMC clock cycles after power supply switching (above sequence).
   c. Clear override SLV_DLL_CLK_OUT_DIS_OVERRIDE_EN to give control to DLL calibration engine.
   d. This sequence can be used to clamp DLL output to zero whenever there is change in DLL controls.
2. This sequence can help avoiding DLL glitch but not trimmer glitch. Software issues a software reset to recover from a bad state due to trimmer glitch. For a simple fix and reduce software overhead, software must issue SW_RESET to recover from the DLL and trimmer glitch issues whenever there is a change in the DLL and trimmer power supply.

9.5.3.8.7 Enhanced Strobe Mode

Refer to the eMMC5.1 specification for HS400 timing mode selection with enhanced strobe mode. HS400 (Enhanced Strobe) Selection flow diagram of the eMMC 5.1 Specification to initialize the Host and Device to HS400 mode. Follow the steps below to enable the enhanced strobe mode at the host:

1. After switching to High Speed mode (SDCLK <= 52MHz) and before enabling HS400 mode, send a switch command to the device to enable 8-bit DDR mode with enhanced strobe.
2. Program the following EXT_CSD at the device:BUS_WIDTH[183] = 0x86.
3. Change the SDCLK to 200 MHz.
4. Run DLL calibration.
5. Enable HS400 mode at the device and Host controller.
6. After switching to HS400 mode at the device and host, program the following register to enable enhanced strobe mode at the host: SDMMCAB_VENDOR_SYS_SW_CNTRL_0 _ENHANCED_STROBE_MODE=1
7. Device is ready for CMD or data transfer.

9.5.3.8.8 Software Programming Sequence for Entering into HS400 Mode

Refer to the eMMC5.1 specification for HS400 mode selection.

9.5.3.8.9 Software Programming Sequence for Changing the Frequency in HS400 Mode

1. Follow the clock stop sequence defined in SD host software sequence chapter.
2. Change the SD/MMC Clk Frequency.
3. SD Clock turn ON sequence.
4. Perform DLL calibration.
5. Configure device in HS400 mode.
6. Set HS400 mode in the controller.

9.5.3.9 Clock Trimmer Settings

9.5.3.9.1 IB and OB Trimmer Tap Value Setting

Software should select proper tap values for outbound clock trimmer (SDMMC_VENDOR_CLOCK_CNTRL_0_TRIM_VAL), and inbound clock trimmer (SDMMC_VENDOR_CLOCK_CNTRL_0_TAP_VAL), to prevent data/cmd CRC errors. Note that the trimmer settings depend on speed mode used. Program as listed in the table below.

Inbound tap values (SDMMC_VENDOR_CLOCK_CNTRL_0_TAP_VAL):*

The tap values below can be used up to SDR25/DDR50 and HIGH SPEED EMMC DDR (eMMC) speeds (Frequency <= 52 MHz).

Controller	Tap Val for I/O trimmer (default)	Tap Val for core trimmer (if SDMMC_VENDOR_SYS_SW_CNTRL_0_IO_TRIM_BYPASS is set to 1)
SDMMC1	9	16
SDMMC4	8	15

The software should use tuning procedures to determine the optimal tap value in SDR50, SDR104, HS200, and HS400 modes.

Outbound tap values (SDMMC_VENDOR_CLOCK_CNTRL_0_TRIM_VAL)*

The settings in the table below are valid in all speed modes.

Controller	Tap Val
SDMMC1	5
SDMMC4	20

9.5.3.9.2 DQS Trimmer Setting

DQS Trimmer Tap Values (SDMMC_VENDOR_CAP_OVERRIDES_0_ DQS_TRIM_VAL)*

The software has to program DQS trimmer tap value as per below table in HS400 mode. The settings below are valid in HS400 speed mode. Effective only for SDMMC4.

Controller	Mode	Tap Val
SDMMC4	HS400	40

9.5.3.10 SDR50/SDR104/HS200 Tuning Procedure

9.5.3.10.1 Vendor Registers Programming Before Executing Tuning Procedure

1. SDMMCxx_VENDOR_IO_TRIM_CNTRL_0_SEL_VREG should be set to 0x0 to select VREG as trimmer power supply which makes trimmer delay independent of VDD_CORE. By default, VDD_CORE is used as trimmer power supply which saves power by turning off BG+REG.
2. SDMMCxx_VENDOR_SYS_SW_CNTRL_0_ IO_TRIM_BYPASS should be set to 0x0 (default value).Set SDMMC_VENDOR_TUNING_CNTRL0_0_ NUM_TUNING_ITERATIONS based on speed mode selected.
   a. TRIES_256 for SDR50 mode
   b. TRIES_128 for SDR104 mode

c. TRIES_128 for HS200 and HS400 modes

3. Set SDMMCxx_VENDOR_TUNING_CNTRL1_0_ STEP_SIZE_SDR104_HS200 with 0x0
4. Set SDMMCxx_VENDOR_TUNING_CNTRL1_0_ STEP_SIZE_SDR50 with 0x0
5. DQ offset function may be turned on during tuning by setting SDMMCxx_VENDOR_TUNING_CNTRL1_0_ DQ_OFFSET. This is to avoid the window merge issue. This field needs to be set to 0 during normal operation.

9.5.3.10.2 Pad Drive Strength Calibration Sequence

SD/MMC controller pads need calibration to set proper drive strength for 50 Ohm driver. Calibration process uses a dedicated calibration pad which is connected to a 100 Ohm resistor to ground. Pad control DRV_TYPE is should select DRIVE2X to get 50 Ohm driver codes using 100 Ohm calibration resistor. Programming details are given in initialization sequence.

SDMMC1/3/4: COMP pad external resistor should be 100 Ohm. DRV_TYPE is set to 2x which provides a 50 Ohm driver.

9.5.3.10.3 Drive Strength Calibration Process

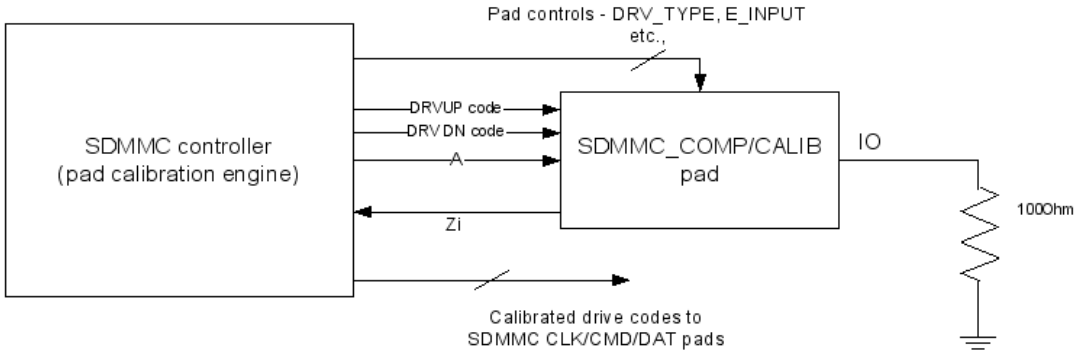
In order to maintain a constant drive strength of 50 Ohm for SD or eMMC interface, an additional calibration pad (SDMMCxx_COMP_PD) is used, which have the same driver as the other pads, to account for:

1. Process, Voltage, and Temperature (PVT) variations - resistor calibration is done once during boot.
2. Temperature Drifts, resistor calibration needs to be done when pads are idle and dynamically

For this, SD/MMC implements resistor calibration registers. The calibration can either be done manually by the Host Driver or SD/MMC can do auto-calibration. The voltage must continue to increase and identify the correct level for driving a 1 and a 0 since PVT/Temperature varies the drive impedance. This is done with the help of an external precise calibration resistor which is connected to the COMP_PD pad and ground.

Within the calibration pads, the drive codes are incremented in steps which, in turn, increase the voltage level on the COMP_PU and decreases the voltage levels on COMP_PD. An internal comparator compares the voltage with a reference voltage and drives a 1 on a match. At this point, the values on the DRVUP/DN values of the comp pads are sampled and programmed into the other pads.

Figure 9.24 Calibration Circuit Diagram



9.5.3.10.4 Calibration Software Sequence

This section provides drive strength calibration sequence details.

1. Turn off SD/SDIO/eMMC device clock before running calibration to avoid unwanted voltage swing changes on CLK pad output
2. Set E_INPUT_OR_E_PWRD = 1 in SDMMCxx_SDMEMCOMPPADCTRL_0 register.
3. Wait for 1 µs after E_INPUT_OR_E_PWRD is enabled.
4. Set AUTO_CAL_START and AUTO_CAL_ENABLE to 1 in SDMMCxx_AUTO_CAL_CONFIG_0 register.
5. Wait for 1 µs.
6. Wait for up to 10 ms for AUTO_CAL_ACTIVE in SDMMCxx_AUTO_CAL_STATUS_0 register to become 0. If it is not 0 after 10 ms, the software can assume auto-calibration has timed out. If it becomes zero within 10 ms, it is assumed that calibration has completed.
7. Clear E_INPUT_OR_E_PWRD to save power. Turn on SD/SDIO/eMMC device clock.
8. The software should not clear AUTO_CAL_ENABLE in SDMMCxx_AUTO_CAL_CONFIG_0 register after calibration. It should be set to 1 always to use calibration codes generated by calibration controller.
9. If timeout occurs during calibration process due to process issues, software should program drive strength up/down static values.

9.5.3.11 Boot ROM Support

The following table lists the cards and voltage rails that are supported for bootable SD/MMC interfaces.

Controller	I/O Segment	I/O Voltage	Data Bus Width
sdmmc4	VDDIO_SDMMC4	1.8 V	8b

Notes:

1. There is no voltage switching supported in Boot ROM. It uses fuses to select the voltage rail for the bootable SDMMC interfaces. If required, bootloader can do voltage switching later on.
2. Boot ROM uses 8-bit data bus width.
3. Reset output available: Yes in the column means that there is a dedicated reset output available for this interface that can be connected to eMMC card.
4. Higher speed modes of eMMC (HS200 or HS400) are not supported by the boot ROM.
5. Boot ROM uses external clock frequency of 375 kHz for initialization of card. For data transfer, it uses 51 MHz depending on speed supported.
6. Boot ROM uses PLLP_OUT0 as the clock source for SDMMC and sets the divisor as need to achieve the desired frequency:
7. Frequency selection for low-speed mode and high-speed mode:
   a. For <= EMMC v4.0, v4.1, v4.2:
      i. Low-speed mode < 20 MHz
      ii. High-speed mode >=20 MHz
   b. For EMMC v4.3 and above:
      i. Low-speed mode < 26 MHz
      ii. High-speed mode >=26 MHz
8. Boot ROM doesn't set SDMMCAB_POWER_CONTROL_HOST_0.HIGH_SPEED_EN bit for any speed mode.

9.5.3.12 SD3.0 Signal Voltage Switching

SD3.0 UHS-I Signal Voltage Switching is only applicable to the removable-card-capable SDMMC1 controller. The SDMMC4 controller does not require these settings because their embedded devices would operate at a fixed voltage.

9.5.4 SDMMC Registers

SDMMC_SYSTEM_ADDRESS_0

32-bit Block Count (SDMA System Address) Register

When Host Version 4 Enable is set to 0 in the Host Control 2 register, SDMA uses this register as system address in only 32-bit addressing mode. Auto CMD23 cannot be used with SDMA. When Host Version 4 Enable is set to 1, SDMA uses ADMA System Address register (05Fh-058h) instead of using this register to support both 32-bit and 64-bit addressing. This register is re-assigned to 32-bit Block Count and then SDMA may use Auto CMD23.

1. SDMA System Address (Host Version 4 Enable = 0)
   This register contains the system memory address for a SDMA transfer in 32-bit addressing mode. When the Host Controller stops a SDMA transfer, this register shall point to the

system address of the next contiguous data position. It can be accessed only if no transaction is executing (i.e., after a transaction has stopped). Reading this register during SDMA transfers may return an invalid value. The Host Driver shall initialize this register before starting a SDMA transaction.
After SDMA has stopped, the next system address of the next contiguous data position can be read from this register. The SDMA transfer waits at the every boundary specified by the SDMA Buffer Boundary in the Block Size register.
The Host Controller generates DMA Interrupt to request the Host Driver to update this register. The Host Driver sets the next system address of the next data position to this register. When the most upper byte of this register (003h) is written, the Host Controller restarts the SDMA transfer. When restarting SDMA by setting Continue Request in the Block Gap Control register, the Host Controller shall start at the next contiguous address stored here in the SDMA System Address register. ADMA does not use this register.

2. 32-bit Block Count (Host Version 4 Enable = 1) Host Controller Version 4.10 re-defines this register as 32-bit Block Count. In version 4.00, this register may be used as 32-bit block count only for Auto CMD23 to set the argument of the CMD23 while executing Auto CMD23.

   FFFF_FFFFh 4G - 1 block
   … …
   0000_0002h 2 blocks
   0000_0001h 1 block
   0000_0000h Stop Count

The Host Controller would decrement the block count of this register every block transfer and data transfer stops when the count reaches zero. This register should be accessed only when no transaction is executing. Reading this register during data transfers may return invalid value.

Offset: 0x0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ADDRESS**

SDMMC_BLOCK_SIZE_BLOCK_COUNT_0

Block Size Register

HOST_DMA_BUFFER_SIZE
The large contiguous memory space may not be available in the virtual memory system. To perform long SDMA transfer, SDMA System Address register shall be

updated at every system memory boundary during SDMA transfer.
These bits specify the size of contiguous buffer in the system memory. The SDMA transfer shall wait at the every boundary specified by these fields and the Host Controller generates the DMA Interrupt to request the Host Driver to update the SDMA System Address register. At the end of transfer, the Host Controller may issue or may not issue DMA Interrupt. In particular, DMA Interrupt shall not be issued after Transfer Complete Interrupt is issued.
In case of this register is set to 0 (buffer size = 4K bytes), lower 12-bit of byte address points data in the contiguous buffer and the upper 20-bit points the location of the buffer in the system memory. The SDMA transfer stops when the Host Controller detects carry out of the address from bit 11 to 12.
These bits shall be supported when the SDMA Support in the Capabilities register is set to 1 and this function is active when the DMA Enable in the Transfer Mode register is set to 1.
ADMA does not use this register.
000b 4K bytes (Detects A11 carry out)
001b 8K bytes (Detects A12 carry out)
010b 16K Bytes (Detects A13 carry out)
011b 32K Bytes (Detects A14 carry out)
100b 64K bytes (Detects A15 carry out)
101b 128K Bytes (Detects A16 carry out)
110b 256K Bytes (Detects A17 carry out)
111b 512K Bytes (Detects A18 carry out)
XFER_BLOCK_SIZE_11_0
This register specifies the block size of data transfers for CMD17, CMD18, CMD24, CMD25, and CMD53. Values ranging from 1 up to the maximum buffer size can be set. In case of memory, it shall be set up to 512 bytes (Refer to Implementation Note in Section 1.7.2). It can be accessed only if no transaction is executing (i.e., after a transaction has stopped). Read operations during transfers may return an invalid value, and write operations shall be ignored.
0800h 2048 Bytes
.....
0200h 512 Bytes
01FFh 511 Bytes
.....
0004h 4 Bytes
0003h 3 Bytes
0002h 2 Bytes
0001h 1 Byte
0000h No data transfer
16-bit BLOCKS_COUNT
Host Controller Version 4.10 extends block count to 32-bit (Refer to Section 1.15 in SD host spec4.1).
Selection of either 16-bit Block Count register or 32-bit Block Count register is defined as follows:
(1) If Host Version 4 Enable in the Host Control 2 register is set to 0 or 16-bit Block Count register is set to non-zero, 16-bit Block Count register is selected.

(2) If Host Version 4 Enable is set to 1 and 16-bit Block Count register is set to zero, 32-bit Block Count register is selected.
Use of 16-bit/32-bit Block Count register is enabled when Block Count Enable in the Transfer Mode register is set to 1 and is valid only for multiple block transfers. The Host Driver shall set this
register to a value between 1 and the maximum block count. The Host Controller decrements the block count after each block transfer and stops when the count reaches zero. Setting the block count to 0 results in no data blocks is transferred. This register should be accessed only when no transaction is executing (i.e., after transactions are stopped). During data transfer, read operations on this register may return an invalid value and write operations are ignored.
When a suspend command is completed, the number of blocks yet to be transferred can be determined by reading this register. Before issuing a resume command, the Host Driver shall restore the previously saved block count.
FFFFh 65535 blocks
.....
0002h 2 blocks
0001h 1 block
0000h Stop Count

Offset: 0x4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BLOCKS_COUNT**
14:12	0x0	**HOST_DMA_BUFFER_SIZE**: **0** = DMA4K **1** = DMA8K **2** = DMA16K **3** = DMA32K **4** = DMA64K **5** = DMA128K **6** = DMA256K **7** = DMA512K
11:0	0x0	**XFER_BLOCK_SIZE_11_0**

SDMMC_ARGUMENT_0

Argument 1 Register

COMMAND_ARGUMENT

The SD Command Argument is specified as bit39-8 of Command-Format in the Physical Layer Specification.

Offset: 0x8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**COMMAND_ARGUMENT**

SDMMC_CMD_XFER_MODE_0

Command and Transfer Mode Register

COMMAND_INDEX
These bits shall be set to the command number (CMD0-63, ACMD0-63) that is specified in bits 45-40 of the Command-Format in the Physical Layer Specification and SDIO Card Specification.
COMMAND_TYPE
There are three types of special commands: Suspend, Resume and Abort. These bits shall be set to 00b for all other commands.
(1) Suspend Command
If the Suspend command succeeds, the Host Controller shall assume the SD Bus has been released and that it is possible to issue the next command, which uses the DAT line. The Host Controller shall de-assert Read Wait for read transactions and stop checking busy for write transactions. The interrupt cycle shall start, in 4-bit mode. If the Suspend command fails, the Host Controller shall maintain its current state, and the Host Driver shall restart the transfer by setting Continue Request in the Block Gap Control register. (Refer to 3.12.1 Suspend Sequence)
(2) Resume Command
The Host Driver re-starts the data transfer by restoring the registers in the range of 000-00Dh. (Refer to Figure 1-4 in section 1.6 for the register map.) The Host Controller shall check for busy before starting write transfers.
(3) Abort Command
If this command is set when executing a read transfer, the Host Controller shall stop reads to the buffer. If this command is set when executing a write transfer, the Host Controller shall stop driving the DAT line. After issuing the Abort command, the Host Driver should issue a software reset. (Refer to 3.8 Abort Transaction)
11b Abort CMD12, CMD52 for writing "I/O Abort" in CCCR

10b Resume CMD52 for writing "Function Select" in CCCR
01b Suspend CMD52 for writing "Bus Suspend" in CCCR
00b Normal Other commands
DATA_PRESENT_SELECT
This bit is set to 1 to indicate that data is present and shall be transferred using the DAT line. It is set to 0 for the following:
(1) Commands using only CMD line (ex. CMD52).
(2) Commands with no data transfer but using busy signal on DAT[0] line (R1b or R5b ex. CMD38)
(3) Resume command
CMD_INDEX_CHECK_EN
If this bit is set to 1, the Host Controller shall check the Index field in the response to see if it has the same value as the command index. If it is not, it is reported as a Command Index Error. If this bit is set to 0, the Index field is not checked.
CMD_CRC_CHECK_EN
If this bit is set to 1, the Host Controller shall check the CRC field in the response. If an error is detected, it is reported as a Command CRC Error. If this bit is set to 0, the CRC field is not checked. The position of CRC field is determined according to the length of the response. (Refer to definition in D01-00 and Table 2-10 below.)
Sub Command Flag
This bit is added from Version 4.10 to distinguish a main command or sub command (Refer to Section 1.17). When issuing a main command, this bit is set to 0 and when issuing a sub command, this bit is set to 1. Setting of this bit is checked by Sub Command Status in the Present State register.
Host Driver manages whether main or sub command. Host Controller does not refer to this bit to issue a command.
1 Sub Command
0 Main Command
RESP_TYPE_SELECT
Normal Mode:
00 No Response
01 Response Length 136
10 Response Length 48
11 Response Length 48 check Busy after response
Response Interrupt Disable
Host Controller Version 4.00 supports response error check function to avoid overhead of response error check by Host Driver. Only R1 or R5 can be checked.
If Host Driver checks response error, sets this bit to 0 and waits Command Complete Interrupt and then check the response register.
If Host Controller checks response error, sets this bit to 1 and sets Response Error Check Enable to 1. Command Complete Interrupt is disabled by this bit regardless of Command Complete Signal Enable.
0 Response Interrupt is enabled
1 Response Interrupt is disabled
Response Error Check Enable

Host Controller Version 4.00 supports response error check function to avoid overhead of response error check by Host Driver. Only R1 or R5 can be checked. If Host Driver checks response error, this bit is set to 0 and Response Interrupt Disable is set to 0.
If Host Controller checks response error, sets this bit to 1 and sets Response Interrupt Disable to 1. Response Type R1 / R5 selects either R1 or R5 response type. If an error is detected, Response Error Interrupt is generated in the Response Error Interrupt Status register
0 Response Error Check is disabled
1 Response Error Check is enabled
Response Type R1 / R5
When response error check is enabled, this bit selects either R1 or R5 response types. Two types of response check is supported: R1 for memory and R5 for SDIO.
Error Statuses Checked in R1
Bit31 OUT_OF_RANGE
Bit30 ADDRESS_ERROR
Bit29 BLOCK_LEN_ERROR
Bit26 WP_VIOLATION
Bit25 CARD_IS_LOCKED
Bit23 COM_CRC_ERROR
Bit21 CARD_ECC_FAILED
Bit20 CC_ERROR
Bit19 ERROR
Response Flags Checked in R5
Bit07 COM_CRC_ERROR
Bit03 ERROR
Bit01 FUNCTION_NUMBER
Bit00 OUT_OF_RANGE
0 R1 (Memory)
1 R5 (SDIO)
MULTI_BLOCK_SELECT - Multi / Single Block Select
This bit is set when issuing multiple-block transfer commands using DAT line. For any other commands, this bit shall be set to 0. If this bit is 0, it is not necessary to set the Block Count register. (Refer to Table 2-8)
1 Multiple Block
0 Single Block
DATA_XFER_DIR_SEL - Data Transfer Direction Select
This bit defines the direction of DAT line data transfers. The bit is set to 1 by the Host Driver to transfer data from the SD card to the SD Host Controller and it is set to 0 for all other commands.
1 Read (Card to Host)
0 Write (Host to Card)
AUTO_CMD12_EN - Auto CMD Enable
This field determines use of auto command functions.
00b Auto Command Disabled

01b Auto CMD12 Enable
10b Auto CMD23 Enable
11b Auto CMD Auto Select
There are three methods to stop Multiple-block read/write operation by CMD23 or CMD12.
In the other operations (ex. single read/write operation), this field is set to 00b.

(1) Auto CMD12 Enable
When this field is set to 01b, the Host Controller issues CMD12 automatically when last block transfer is completed. Auto CMD12 error is indicated to the Auto CMD Error Status register. The Host Driver shall not set this bit if the command does not require CMD12. In particular, secure commands
defined in the Part 3 File Security specification do not require CMD12.
When Host Version 4 Enable =0, CMD12 is issued when 16-bit Block Count is expired.
When Host Version 4 Enable =1, CMD12 is issued when 16-bit Block Count or 32-bit Block Count is expired.

(2) Auto CMD23 Enable
When this bit field is set to 10b, the Host Controller issues a CMD23 automatically before issuing a command specified in the Command Register. The Host Controller Version 3.00 and later shall support this function. The following conditions are required to use the Auto CMD23.
Auto CMD23 Supported (Host Controller Version is 3.00 or later)
A memory card that supports CMD23 (SCR[33]=1)
If DMA is used, it shall be ADMA.
Only when CMD18 or CMD25 is issued
(Note, the Host Controller doesnt check command index.)
Auto CMD23 can be used with or without ADMA. By writing the Command register, the Host Controller issues a CMD23 first and then issues a command specified by the Command Index in Command register. If response errors of CMD23 are detected, the second command is not issued. A CMD23 error is indicated in the Auto CMD Error Status register.
32-bit block count value for CMD23 is set to 32-bit Block Count (SDMA System Address) register.

(3) Auto CMD Auto Select (Version 4.10)
As CMD23 is optional for SD Memory Card except UHS104 Card, If card supports CMD23, Auto CMD23 should be used instead of Auto CMD12. Host Controller Version 4.10 defines this "Auto CMD Auto Select" mode.
Selection of Auto CMD depends on setting of CMD23 Enable in the Host Control 2 register which indicates whether card supports CMD23. If CMD23 Enable =1, Auto CMD23 is used and if CMD23 Enable =0, Auto CMD12 is used. In case of Version 4.10 or later, use of Auto CMD Auto Select is recommended rather than use of Auto CMD12 Enable or Auto CMD23 Enable.
BLOCK_COUNT_EN - Block Count Enable
This bit is used to enable the Block Count register, which is only relevant for multiple block transfers. When this bit is 0, the Block Count register is disabled, which is useful in executing an infinite transfer. (Refer to Table 2-8)

Host Driver should set this bit to 0 when ADMA is used.
DMA_EN - DMA Enable
This bit enables DMA functionality as described in section 1.4. DMA can be enabled only if it is supported as indicated in the Capabilities register. One of the DMA modes can be selected by DMA Select in the Host Control register. If DMA is not supported, this bit is meaningless and shall always read 0. If this bit is set to 1, a DMA operation shall begin when the Host Driver writes to the upper byte of Command register (00Fh).

Offset: 0xc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
29:24	0x0	**COMMAND_INDEX**
23:22	0x0	**COMMAND_TYPE**: **0** = NORMAL **1** = SUSPEND **2** = RESUME **3** = ABORT
21	0x0	**DATA_PRESENT_SELECT**: **0** = NO_DATA_TRANSFER **1** = DATA_TRANSFER
20	0x0	**CMD_INDEX_CHECK_EN**: **0** = DISABLE **1** = ENABLE
19	0x0	**CMD_CRC_CHECK_EN**: **0** = DISABLE **1** = ENABLE
18	0x0	**SUB_CMD_FLAG**: **0** = MAIN_CMD **1** = SUB_CMD
17:16	0x0	**RESP_TYPE_SELECT**: **0** = NO_RESPONSE **1** = RESP_LENGTH_136 **2** = RESP_LENGTH_48 **3** = RESP_LENGTH_48BUSY
8	0x0	**RESP_INT_DIS**: **0** = ENABLE **1** = DISABLE
7	0x0	**RESP_ERR_CHK_EN**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
6	0x0	**RESP_TYPE**: **0** = R1 **1** = R5
5	0x0	**MULTI_BLOCK_SELECT**: **0** = DISABLE **1** = ENABLE
4	0x0	**DATA_XFER_DIR_SEL**: **0** = WRITE **1** = READ
3:2	0x0	**AUTO_CMD12_EN**: **0** = DISABLE **1** = CMD12 **2** = CMD23 **3** = AUTO_CMD_AUTO_SEL
1	0x0	**BLOCK_COUNT_EN**: **0** = DISABLE **1** = ENABLE
0	0x0	**DMA_EN**: **0** = DISABLE **1** = ENABLE

SDMMC_RESPONSE_R0_R1_0

Command Response Registers

The Table below describes the mapping of command responses from the SD Bus to this register for each response type. In the table, R[] refers to a bit range within the response data as transmitted on the SD Bus, REP[] refers to a bit range within the Response register.

Kind of Response	Meaning of Response	Response Field	Response Register
R1, R1b (normal response)	Card Status	R [39:8]	REP [31:0]
R1b (Auto CMD12 response)	Card Status for Auto CMD12	R [39:8]	REP [127:96]
R1 (Auto CMD23 response)	Card Status for Auto CMD23	R [39:8]	REP [127:96]
R2 (CID, CSD register)	CID or CSD reg. incl.	R [127:8]	REP [119:0]
R3 (OCR register)	OCR register for memory	R [39:8]	REP [31:0]
R4 (OCR register)	OCR register for I/O etc	R [39:8]	REP [31:0]
R5,R5b	SDIO response	R [39:8]	REP [31:0]
R6 (Published RCA response)	New published RCA[31:16] etc	R [39:8]	REP [31:0]

Response Bit Definition for Each Response Type

In UHS-II mode, the response of CM-TRAN abort CCMD (4-byte) is stored in offset 13h-10h and the response of SD-TRAN abort CCMD (8-byte) is stored in offset 1Fh-18h Command Response [31:0] (R0) Register

Offset: 0x10
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CMD_RESP_31_16**
15:0	0x0	**CMD_RESP_15_0**

SDMMC_RESPONSE_R2_R3_0

Command Response [63:32] (R2) Register

Offset: 0x14
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CMD_RESP_63_48**
15:0	0x0	**CMD_RESP_47_32**

SDMMC_RESPONSE_R4_R5_0

Command Response [95:64] (R4) Register

Offset: 0x18
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CMD_RESP_95_80**

Bit	Reset	Description
15:0	0x0	**CMD_RESP_79_64**

SDMMC_RESPONSE_R6_R7_0

Command Response [127:96] (R6) Register

Offset: 0x1c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CMD_RESP_127_112**
15:0	0x0	**CMD_RESP_111_96**

SDMMC_BUFFER_DATA_PORT_0

Buffer Data Port Register

The Host Controller Buffer can be accessed through this 32-bit Data Port Register.

Offset: 0x20
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**BUFFER_DATA**

SDMMC_PRESENT_STATE_0

Present State Register

CMD_LINE_LEVEL - CMD Line Signal Level
This status is used to check the CMD line level to recover from errors, and
for debugging.
DAT_3_0_LINE_LEVEL - DAT[3:0] Line Signal Level

This status is used to check the DAT line level to recover from errors, and for debugging. This is especially useful in detecting the busy signal level from DAT[0].
Sub Command Status
The Command register and Response register are commonly used for main command and sub command. This status is used to distinguish which response error statuses, main command or sub command, indicated in the Error Interrupt Status register or in the UHS-II Error Interrupt Status register. Refer to Section 1.17 about details of response error statuses. Just before reading of this register, the Sub Command Flag of the Command register or the UHS-II Command register is copied to this status. This status is effective not only when Response Error interrupt is generated but also when data error interrupt is generated with Command Not Issued by Error (D27 of this register) or Auto CMD Error interrupt is generated with Command Not Issued by Error by Auto CMD12 in the Auto CMD Error Status register.

1 Sub Command Status
0 Main Command Status
Command Not Issued by Error
Setting of this status indicates that a command cannot be issued due to an error except Auto CMD12 error. (Equivalent error status by Auto CMD12 error is defined as Command Not Issued By Auto CMD12 Error in the Auto CMD Error Status register.) This status is set to 1 when Host Controller cannot issue a command after setting Command register or UHS-II Command register. Refer to Section 3.10 about 2L-HD error case in UHS-II mode. Sub Command Status (D28) indicates which command is not issued (main or sub).

1 Command cannot be issued
0 No error for issuing a command

Host Regulator Voltage Stable
This status is added from Version 4.10 and is used to check whether host regulator voltage is stable for switching signal voltage of UHS-I mode.

1 Host Regulator Voltage is stable
0 Host Regulator Voltage is not stable

Support of this function is checked by reading this status after that Software Reset For All in the Software Reset register is cleared by the Host Controller in initialization. Setting this status to 1 means that this function is supported by the Host Controller.

This status may be related to 1.8V Signaling Enable in the Host Control 2 register. Changing 1.8V Signaling Enable causes unstable of host regulator voltage for I/O cell. Then once this status is set to 0 and retrieved to 1 when host regulator voltage is stable again. When executing power

cycle, Host Driver also executes Software Reset For All and it clears 1.8V Signaling Enable to go back signal voltage to 3.3V.

If this status is not supported, Host Driver should take more than 5ms for stable time of host voltage regulator from changing 1.8V Signaling Enable. Specific Host Driver may use a specific time, which is provided by Host System, instead of using 5ms.
WRITE_PROTECT_LEVEL - Write Protect Switch Pin Level
CARD_DETECT_PIN_LEVEL - Card Detect Pin Level
CARD_STATE_STABLE - Card State Stable
CARD_INSERTED - Card Inserted
BUFFER_READ_EN - Buffer Read Enable
This status is used for non-DMA read transfers.
BUFFER_WRITE_EN - Buffer Write Enable
This status is used for non-DMA write transfers.
READ_XFER_ACTIVE - Read Transfer Active
WRITE_XFER_ACTIVE - Write Transfer Active
RETUNING_REQUEST - Re-Tuning Request
DAT_LINE_ACTIVE - DAT Line Active
CMD_INHIBIT_DAT - Command Inhibit (DAT)
CMD_INHIBIT_CMD - Command Inhibit (CMD)

Offset: 0x24
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxx0,0x00,0000,0000,xxxx,0000,0000,0000)
PROD: 0x000b0000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	PROD	Description
28	0x0	_NONE_	**SUB_CMD_STATUS**
27	0x0	_NONE_	**CMD_NOT_ISSUED_BY_ERROR**
25	0x0	_NONE_	**HOST_REG_VOLTAGE_STABLE**
24	0x0	_NONE_	**CMD_LINE_LEVEL**: **0** = LOW **1** = HIGH
23:20	0x0	_NONE_	**DAT_3_0_LINE_LEVEL**
19	0x0	ENABLED	**WRITE_PROTECT_LEVEL**: **0** = PROTECTED **1** = ENABLED
18	0x0	_NONE_	**CARD_DETECT_PIN_LEVEL**: **0** = NO_CARD **1** = CARD

Bit	Reset	PROD	Description
17	0x0	INSERTED	**CARD_STATE_STABLE**: **0** = DEBOUNCE **1** = INSERTED
16	0x0	INSERTED	**CARD_INSERTED**: **0** = DEBOUNCE **1** = INSERTED
11	0x0	_NONE_	**BUFFER_READ_EN**: **0** = DISABLE **1** = ENABLE
10	0x0	_NONE_	**BUFFER_WRITE_EN**: **0** = DISABLE **1** = ENABLE
9	0x0	_NONE_	**READ_XFER_ACTIVE**: **0** = NO_DATA **1** = TRANSFERING
8	0x0	_NONE_	**WRITE_XFER_ACTIVE**: **0** = NO_DATA **1** = TRANSFERING
7:4	0x0	_NONE_	**DAT_7_4_LINE_LEVEL**
3	0x0	_NONE_	**RETUNING_REQUEST**: **0** = NOT_REQUIRED **1** = REQUIRED
2	0x0	_NONE_	**DAT_LINE_ACTIVE**: **0** = INACTIVE **1** = ACTIVE
1	0x0	_NONE_	**CMD_INHIBIT_DAT**: **0** = INACTIVE **1** = ACTIVE
0	0x0	_NONE_	**CMD_INHIBIT_CMD**: **0** = INACTIVE **1** = ACTIVE

SDMMC_POWER_CONTROL_HOST_0

Power Control / Host Control Register

WAKEUP_ON_CARD_REMOVAL - Wakeup Event Enable On SD Card Removal
This bit enables wakeup event via Card Removal assertion in the Normal Interrupt Status register. FN_WUS (Wake Up Support) in CIS does not affect this bit.
WAKEUP_ON_CARD_INSERTION - Wakeup Event Enable On SD Card Insertion
This bit enables wakeup event via Card Insertion assertion in the Normal Interrupt Status register. FN_WUS (Wake Up Support) in CIS does not affect this bit.
WAKEUP_ON_CARD_INTERRUPT - Wakeup Event Enable On Card Interrupt
This bit enables wakeup event via Card Interrupt assertion in the Normal Interrupt Status register. This bit can be set to 1 if FN_WUS (Wake Up Support) in CIS is set

to 1.
INTERRUPT_AT_BLOCK_GAP - Interrupt At Block Gap
This bit is valid only in 4-bit mode of the SDIO card and selects a sample point in the interrupt cycle. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer. Setting to 0 disables interrupt detection during a multiple block transfer. If the SD card cannot signal an interrupt during a multiple block transfer, this bit should be set to 0. When the Host Driver detects an SD card insertion, it shall set this bit according to the CCCR of the SDIO card.
READ_WAIT_CONTROL - Read Wait Control
The read wait function is optional for SDIO cards. If the card supports read wait, set this bit to enable use of the read wait protocol to stop read data using the DAT[2] line. Otherwise, the Host Controller has to stop the SD Clock to hold read data, which restricts commands generation. When the Host Driver detects an SD card insertion, it shall set this bit according to the CCCR of the SDIO card. If the card does not support read wait, this bit shall never be set to 1 otherwise DAT line conflict may occur. If this bit is set to 0, Suspend/Resume cannot be supported. In UHS-II mode, Read Wait is disabled and DAT[2] line is used for Interrupt Signal from UHS-II Card.
CONTINUE_REQUEST - Continue Request
This bit is used to restart a transaction, which was stopped using the Stop At Block Gap Request. To cancel stop at the block gap, set Stop At Block Gap Request to 0 and set this bit 1 to restart the transfer.
The Host Controller automatically clears this bit when the transaction re-starts. If Stop At Block Gap Request is set to 1, any write to this bit is ignored.
In SD mode, this bit is cleared in either of the following cases:
(1) In the case of a read transaction, the DAT Line Active changes from 0 to 1 as a read transaction restarts.
(2) In the case of a write transaction, the Write Transfer Active changes from 0 to 1 as the write transaction restarts.
Therefore, it is not necessary for Host Driver to set this bit to 0. If Stop At Block Gap Request is set to 1, any write to this bit is ignored.
STOP_AT_BLOCK_GAP_REQUEST - Stop At Block Gap Request
This bit is used to stop executing read and write transaction at the next block gap for non-DMA, SDMA and ADMA transfers. The Host Driver shall leave this bit set to 1 until the Transfer Complete is set to 1. Clearing both Stop At Block Gap Request and Continue Request shall not cause the transaction to restart. When Host Controller version is 1.00, the Host Driver can set this bit if the card supports Read Wait Control. When Host Controller version is 2.00 or higher, the Host Driver can set this bit regardless of the card supports Read Wait Control. The Host Controller shall stop read transfer by using Read Wait or stopping SD clock. In case of write transfers in which the Host Driver writes data to the Buffer Data Port register, the Host Driver shall set this bit after all block data is written. If this bit is set to 1, the Host Driver shall not write data to Buffer Data Port register. This bit affects Read Transfer Active, Write Transfer Active, DAT Line Active and Command Inhibit (DAT) in the Present State register.
Regarding detailed control of bits D01 and D00, refer to Section 3.8 and 3.12.

SD_BUS_VOLTAGE_SELECT - The Host doesnt support the bus voltage selections. For SDMMC1 Interfaces,
voltage switching between 3.3V to 1.8V is done by programming the PMU through I2C Interface.
SD_BUS_POWER - SD Bus Power
Before setting this bit, the SD Host Driver shall set SD Bus Voltage Select. If the Host Controller detects the No Card state, this bit shall be cleared.
If this bit is cleared, the Host Controller shall immediately stop driving CMD and DAT[3:0] (tri-state) and drive SDCLK to low level (refer to Section 2.2.14).
CARD_DETECT_SIGNAL_DETECT - Card Detect Signal Selection
This bit selects source for the card detection.
When the source for the card detection is switched, the interrupt should be disabled during the switching period by clearing the Interrupt Status/Signal Enable register in order to mask unexpected interrupt being caused by the glitch.
The Interrupt Status/Signal Enable should be disabled during over the period of debouncing.
CARD_DETECT_TEST_LVL - Card Detect Test Level
This bit is enabled while the Card Detect Signal Selection is set to 1 and it indicates card inserted or not.
EXTENDED_DATA_TRANSFER_WIDTH (VENDOR Bit)
1:8-bit Mode,DATA_XFER_WIDTH is ignored.
0:Card bus width is as per DATA_XFER_WIDTH value

DMA Select
This field is used to select DMA type. The Host Driver shall check support of DMA modes by referring the Capabilities register. Selected DMA is enabled by DMA Enable of the Transfer Mode register in SD mode and DMA Enable of UHS-II Transfer Mode register in UHS-II mode.

(1) Up to Version 3.00
When Host Version 4 Enable is set to 0, setting of this field is compatible to Host Controller Version 3.00.
SDMA is initiated by writing to the Command register when this field is set to 00b and the SDMA System Address register (32-bit) is used.
SDMA does not support 64-bit addressing.
ADMA2 is initiated by writing to the Command register when this field is set to 10b or 11b. Lower 32-bit of the ADMA System Address register is used when this field is set to 10b and 64-bit of the ADMA System Address register is used when this field is set to 11b. Support of 64-bit System Addressing is indicated by 64-bit System Address Support for V3 in the Capabilities register. 64-bit AMDA2 uses 96-bit Descriptor.

00 SDMA is selected
01 Reserved (New assignment is not allowed)
10 32-bit Address ADMA2 is selected
11 64-bit Address ADMA2 is selected (Optional)

(2) Version 4.00 or later
When Host Version 4 Enable is set to 1, setting of this field is changed as follows. SDMA is initiated by Host Driver writes to the Command register when this field is set to 00b.
ADMA2 is initiated by Host Driver writes to the Command register when this field is set to 10b or 11b and by ADMA3 sets to the ADMA System Address register when this field is set to 11b.
ADMA3 is initiated by Host Driver writes to the ADMA3 ID Address register when this field is set to 11b.

00 SDMA is selected
01 Not Used (New assignment is not allowed)
10 ADMA2 is selected (AMDA3 is not supported or disabled)
11 ADMA2 or ADMA3/CQE is selected

Support of 64-bit DMA and 128-bit Descriptor is indicated by 64-bit System Address Support for V4 in the Capabilities register. If the support bit is set to 1, all supported DMAs (depends on Support, ADMA2 Support and ADMA3 Support) shall support 64-bit addressing. 64-bit Addressing in the Host Controller 2 register selects either 32-bit or 64-bit system addressing of DMAs.

HIGH_SPEED_EN - High Speed Enable
This bit is optional. Before setting this bit, the Host Driver shall check the High Speed Support in the Capabilities register. If this bit is set to 0 (default), the Host Controller outputs CMD line and DAT lines at the falling edge of the SD Clock (up to 25MHz). If this bit is set to 1, the Host Controller outputs CMD line and DAT lines at the rising edge of the SD Clock (up to 50MHz).

If Preset Value Enable in the Host Control 2 register is set to 1, Host Driver needs to reset SD Clock Enable before changing this field to avoid generating clock glitches. After setting this field, the Host Driver sets SD Clock Enable again.
DATA_XFER_WIDTH - Data Transfer Width
This bit selects the data width of the Host Controller. The Host Driver shall set it to match the data width of the SD card.
LED_CONTROL - LED Control
This bit is used to caution the user not to remove the card while the SD card is being accessed. If the software is going to issue multiple SD commands, this bit can be set during all these transactions. It is not necessary to change for each transaction.

Offset: 0x28
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

Reset: 0x00000000 (0bxxxx,x000,xxxx,0000,xxxx,0000,0000,0000)
PROD: 0x00000002 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx0x,xx1x)

Bit	Reset	PROD	Description
26	0x0	_NONE_	**WAKEUP_ON_CARD_REMOVAL**: **0** = DISABLE **1** = ENABLE
25	0x0	_NONE_	**WAKEUP_ON_CARD_INSERTION**: **0** = DISABLE **1** = ENABLE
24	0x0	_NONE_	**WAKEUP_ON_CARD_INTERRUPT**: **0** = DISABLE **1** = ENABLE
19	0x0	_NONE_	**INTERRUPT_AT_BLOCK_GAP**: **0** = DISABLE **1** = ENABLE
18	0x0	_NONE_	**READ_WAIT_CONTROL**: **0** = DISABLE **1** = ENABLE
17	0x0	_NONE_	**CONTINUE_REQUEST**: **0** = IGNORED **1** = RESTART
16	0x0	_NONE_	**STOP_AT_BLOCK_GAP_REQUEST**: **0** = TRANSFER **1** = STOP
11:9	0x0	_NONE_	**SD_BUS_VOLTAGE_SELECT**: **5** = V1_8 **6** = V3_0 **7** = V3_3
8	0x0	_NONE_	**SD_BUS_POWER**: **0** = POWER_OFF **1** = POWER_ON
7	0x0	_NONE_	**CARD_DETECT_SIGNAL_DETECT**: **0** = SDCD **1** = CARD_DTECT_TST_LVL
6	0x0	_NONE_	**CARD_DETECT_TEST_LVL**: **0** = NO_CARD **1** = CARD_INSERTED
5	0x0	NOBIT_8	**EXTENDED_DATA_TRANSFER_WIDTH**: **0** = NOBIT_8 **1** = BIT_8
4:3	0x0	_NONE_	**DMA_SELECT**: SW should select ADMA3_CQE in eMMC CMD queuing mode. **0** = SDMA **1** = RSVD **2** = ADMA2 **3** = ADMA3_CQE

Bit	Reset	PROD	Description
2	0x0	_NONE_	**HIGH_SPEED_EN**: **0** = NORMAL_SPEED **1** = HIGH_SPEED
1	0x0	BIT_4	**DATA_XFER_WIDTH**: **0** = BIT_1 **1** = BIT_4
0	0x0	_NONE_	**LED_CONTROL**: **0** = OFF **1** = ON

SDMMC_SW_RESET_TIMEOUT_CTRL_CLOCK_CONTROL_0

Clock Control Register

SW_RESET_FOR_DAT_LINE - Software Reset For DAT Line
Only part of data circuit is reset. DMA circuit is also reset.
The following registers and bits are cleared by this bit:
Buffer Data Port register
Buffer is cleared and initialized.
Present State register
Buffer Read Enable
Buffer Write Enable
Read Transfer Active
Write Transfer Active
DAT Line Active
Command Inhibit (DAT)
Block Gap Control register
Continue Request
Stop At Block Gap Request
Normal Interrupt Status register
Buffer Read Ready
Buffer Write Ready
DMA Interrupt
Block Gap Event
Transfer Complete
SW_RESET_FOR_CMD_LINE - Software Reset For CMD Line
Only part of command circuit is reset to be able to issue a command. From Version 4.10, this bit is also used to initialize UHS-II command circuit. This reset is effective only command issuing circuit (including response error statuses related to Command Inhibit (CMD) control) and does not affect data transfer circuit. Host Controller can continue data transfer even this reset is executed during handling of sub command response errors.

The following registers and bits are cleared by this bit:
Present State register
Command Inhibit (CMD)
Normal Interrupt Status register
Command Complete
Error Interrupt Status (from Version 4.10)
Response error statuses related to Command Inhibit (CMD)
SW_RESET_FOR_ALL - Software Reset For All
This reset affects the entire Host Controller except for the card detection circuit. Register bits of type ROC, RW, RW1C, RWAC are cleared to 0. During its initialization, the Host Driver shall set this bit to 1 to reset the Host Controller. The Host Controller shall reset this bit to 0 when Capabilities registers are valid and the Host Driver can read them. Additional use of Software Reset For All may not affect the value of the Capabilities registers. If this bit is set to 1, the host driver should issue reset command and reinitialize the SD card.
DATA_TIMEOUT_COUNTER_VALUE - Data Timeout Counter Value
This value determines the interval by which DAT line timeouts are detected. For more information about timeout generation, refer to the Data Timeout Error in the Error Interrupt Status register. Timeout clock frequency will be generated by dividing the base clock TMCLK value by this value. When setting this register, prevent inadvertent timeout events by clearing the Data Timeout Error Status Enable (in the Error Interrupt Status Enable register)
1111b Reserved
1110b TMCLK x pow(2,27)
.....
0001b TMCLK x pow(2,14)
0000b TMCLK x pow(2,13)

There are two types of busy periods in a multiple block write operation.
(1) Write busy at block gap (without CMD12) is maximum 250ms
(2) Write busy after CMD12 is maximum 250ms (500ms for SDXC)

If CMD12 is issued during a multiple block write operation busy period, the host timeout counter is reset and the 250ms (500ms for SDXC) timeout period is measured from the response of CMD12.

The duration of an erase command can be estimated by the number of write blocks (WRITE_BL) to be
erased multiplied by 250 ms.

SDCLK_FREQUENCYSELECT - SDCLK Frequency Select
This register is used to select the frequency of SDCLK pin. The frequency is not programmed directly; rather this register holds the divisor of the Base Clock Frequency For SD Clock in the Capabilities register. Only the following settings are allowed.
UPPER_SDCLK_FREQUENCYSELECT - Upper Bits of SDCLK Frequency Select
Host Controller Version 1.00 and 2.00 do not support these bits and they are treated as 00b fixed value (ROC).

Host Controller Version 3.00 shall support these bits to expand SDCLK Frequency Select to 10-bit. Bit 07-06 is assigned to bit 09-08 of clock divider in SDCLK Frequency Select.
CLOCK_GENERATOR_SELECT - Clock Generator Select
Host Controller Version 3.00 supports this bit. This bit is used to select the clock generator mode in SDCLK Frequency Select.
If the Programmable Clock Mode is supported (non-zero value is set to Clock Multiplier in the Capabilities register), this bit attribute is RW, and if not supported, this bit attribute is RO and zero is read.
This bit depends on the setting of Preset Value Enable in the Host Control 2 register.
If the Preset Value Enable = 0, this bit is set by Host Driver.
If the Preset Value Enable = 1, this bit is automatically set to a value specified in one of Preset Value registers.
PLL Enable
This bit is added from Version 4.10 for Host Controller using PLL. This feature allows Host Controller to initialize clock generator in two steps: by Internal Clock Enable and PLL Enable and to minimize output latency (ex. SDCLK/RCLK, D0 lane) from SD Clock Enable. There are two modes to keep Host Drivers compatibility. In both modes, PL L Locked timing is indicated by Internal Clock Stable.

(1) When Host Version 4 Enable =0 (Host Driver Version 3, which does not support this bit) or this bit is not implemented, Internal Clock Enable (or SD Clock Enable) may activate PLL (exit low power mode and start locking clock).

(2) When Host Version 4 Enable =1 (Host Driver Version 4), Internal Clock Enable is set before setting this bit and then setting this bit may activate PLL (exit low power mode and start locking clock).
1 PLL is enabled
0 PLL is in low power mode
SD_CLK_EN
The Host Controller shall stop providing SDCLK or RCLK when writing this bit to 0. SDCLK/RCLK Frequency Select can be changed when this bit is 0. Then, the Host Controller shall maintain the same clock frequency until SDCLK is stopped (Stop at SDCLK=0). If the Card Inserted in the Present State register is cleared, this bit shall be cleared.

1 Enable providing SDCLK or RCLK
0 Disable providing SDCLK or RCLK

(1) SD Mode
This is the case when UHS-II Interface Enable is set to 0 in the Host Control 2 register. By setting this bit to 1, SDCLK is provided on pin number 5 (CLK). Refer to Section 1.12 Controlling SDCLK.
When PLL is used to generate clock, PLL is enabled by PLL Enable (if

supported) or by SD Clock Enable (if PLL Enable is not supported).
When PLL is enabled by PLL Enable, the clock synchronization is checked by Internal Clock Stable.

INTERNAL_CLOCK_STABLE - Internal Clock Stable
As PLL Enable is added from Version 4.10, this status is expanded to check two cases. Host Driver Version 4.10 checks clock stability by this status twice after Internal Clock Enable is set and after PLL Enable is set.
Refer to Figure 3-3 in SD host spec4.1.

(1) Internal Clock Stable (when PLL Enable = 0 or not supported)
This bit is set to 1 when internal clock is stable after writing to Internal Clock Enable in this register to 1.

(2) PLL Clock Stable (when PLL Enable = 1)
Host Controller which supports PLL Enable sets this status to 0 once when PLL Enable is changed 0 to 1 and then this status is set to 1 when PLL is locked. (PLL uses an internal clock in stable as a reference clock which is enabled by Internal Clock Enable). After this bit is set to 1, Host Driver may set SD Clock Enable.

1 Ready
0 Not Ready

INTERNAL_CLOCK_EN - Internal Clock Enable
This bit is set to 0 when the Host Driver is not using the Host Controller or the Host Controller awaits a wakeup interrupt. The Host Controller should stop its internal clock to go very low power state. Still, registers shall be able to be read and written. Clock starts to oscillate when this bit is set to 1. When clock oscillation is stable, the Host Controller shall set Internal Clock Stable in this register to 1. This bit shall not affect card detection.

Offset: 0x2c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,x000,xxxx,0000,0000,0000,000x,0000)
PROD: 0x00000002 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	PROD	Description
26	RW	0x0	_NONE_	**SW_RESET_FOR_DAT_LINE**: **0** = WORK **1** = RESETED
25	RW	0x0	_NONE_	**SW_RESET_FOR_CMD_LINE**: **0** = WORK **1** = RESETED

Bit	R/W	Reset	PROD	Description
24	RW	0x0	_NONE_	**SW_RESET_FOR_ALL**: **0** = WORK **1** = RESETED
19:16	RW	0x0	_NONE_	**DATA_TIMEOUT_COUNTER_VALUE**
15:8	RW	0x0	_NONE_	**SDCLK_FREQUENCYSELECT**: **128** = DIV256 **64** = DIV128 **32** = DIV64 **16** = DIV32 **8** = DIV16 **4** = DIV8 **2** = DIV4 **1** = DIV2 **0** = BASE
7:6	RW	0x0	_NONE_	**UPPER_SDCLK_FREQUENCYSELECT**
5	RW	0x0	_NONE_	**CLOCK_GENERATOR_SELECT**
3	RW	0x0	_NONE_	**PLL_EN**: In legacy SD mode, no separate PLL is used to generate SDCLK. SW should set this as a part of standard SDCLK generation process. PLL is used to generate RCLK in UHS-II mode
2	RW	0x0	_NONE_	**SD_CLOCK_EN**: **0** = DISABLE **1** = ENABLE
1	RO	0x0	READY	**INTERNAL_CLOCK_STABLE**: **0** = NOT_READY **1** = READY
0	RW	0x0	_NONE_	**INTERNAL_CLOCK_EN**: when disabled turns off PLL in uhsII IOBRICK in uhsII mode **0** = STOP **1** = OSCILLATE

SDMMC_INTERRUPT_STATUS_0

Normal Interrupt Status Register

VEND_SPEC_ERR[1:0]
1:BOOT_ACK_ERR - Occurs When Boot Ack Status is not equal to '010'
0:BOOT_ACK_TIMEOUT_ERR - Occurs When Boot Ack is not received within the programmed number of cycles.
TARGET_RESP_ERROR - Not supported for Tegra
SPI_ERR
Indicate when SPI Error has occurred. The SPI Errors are registered in SPI_INTERRUPT_STATUS register.

Response Error
Host Controller Version 4.00 supports response error check function to avoid overhead of response error check by Host Driver during DMA execution. If Response Error Check Enable is set to1 in the Transfer Mode register, Host Controller Checks R1 or R5 response. If an error is detected in a response, this bit is set to 1.
TUNING_ERR
This bit is set when an unrecoverable error is detected in a tuning circuit except during tuning procedure (Occurrence of an error during tuning procedure is indicated by Sampling Select). By detecting Tuning Error, Host Driver needs to abort a command executing and perform tuning. To reset tuning circuit, Sampling Clock shall be set to 0 before executing tuning procedure.
ADMA_ERR
This bit is set when the Host Controller detects errors during ADMA based data transfer. The state of the ADMA at an error occurrence is saved in the ADMA Error Status Register,
In addition, the Host Controller generates this Interrupt when it detects invalid descriptor data (Valid=0) at the ST_FDS state. ADMA Error State in the ADMA Error Status indicates that an error occurs in ST_FDS state. The Host Driver may find that Valid bit is not set at the error descriptor.
AUTO_CMD12_ERR
Auto CMD12 and Auto CMD23 use this error status. This bit is set when detecting that any of the bits D00 to D05 in Auto CMD Error Status register has changed from 0 to 1. D07 is effective in case of Auto CMD12. Auto CMD Error Status register is valid while this bit is set to 1 and may be cleared with clearing of this bit (another implementation is also allowed).
CURRENT_LIMIT_ERR
By setting the SD Bus Power bit in the Power Control register, the Host Controller is requested to supply power for the SD Bus. If the Host Controller supports the Current Limit function, it can be protected from an illegal card by stopping power supply to the card in which case this bit indicates a failure status. Reading 1 means the Host Controller is not supplying power to SD card due to some failure. Reading 0 means that the Host Controller is supplying power and no error has occurred. The Host Controller may require some sampling time to detect the current limit. If the Host Controller does not support this function, this bit shall always be set to 0.
DATA_END_BIT_ERR
Occurs either when detecting 0 at the end bit position of read data which uses the DAT line or at the end bit position of the CRC Status.
DATA_CRC_ERR
Occurs when detecting CRC error when transferring read data which uses the DAT line or when detecting the Write CRC status having a value of other than "010".
DATA_TIMEOUT_ERR
Occurs when detecting one of following timeout conditions.
(1) Busy timeout for R1b,R5b type
(2) Busy timeout after Write CRC status

(3) Write CRC Status timeout
(4) Read Data timeout.
COMMAND_INDEX_ERR
Occurs if a Command Index error occurs in the command response.
COMMAND_END_BIT_ERR
Occurs when detecting that the end bit of a command response is 0.
COMMAND_CRC_ERR
Command CRC Error is generated in two cases.
(1) If a response is returned and the Command Timeout Error is set to 0 (indicating no timeout), this bit is set to 1 when detecting a CRC error in the command response.
(2) The Host Controller detects a CMD line conflict by monitoring the CMD line when a command is issued. If the Host Controller drives the CMD line to 1 level, but detects 0 level on the CMD line at the next SDCLK edge, then the Host Controller shall abort the command (Stop driving CMD line) and set this bit to 1. The Command Timeout Error shall also be set to 1 to distinguish CMD line conflict
COMMAND_TIMEOUT_ERR
Occurs only if no response is returned within 64 SDCLK cycles from the end bit of the command. If the Host Controller detects a CMD line conflict, in which case Command CRC Error shall also be set as shown in Table 2-25, this bit shall be set without waiting for 64 SDCLK cycles because the command will be aborted by the Host Controller.
ERR_INTERRUPT
If any of the bits in the Error Interrupt Status register are set, then this bit is set. Therefore the Host Driver can efficiently test for an error by checking this bit first. This bit is read only.
FX_EVENT
This status is added from Version 4.10. Bit06 of response data will be stored in the R[14] of the Response register.
Basically, this interrupt is used with response check function. In this case, this status is set when R[14] of Response register is set to 1 and Response Type R1 / R5 is set to 0 in the Transfer Mode register or UHSII Transfer Mode register. If response check is disabled, this status is set when R[14] of Response register is set to 1. Host Driver needs to screen FX Event interrupt by checking response type is R1.

1 FX_EVENT is detected
0 No Event
RETUNING_EVENT - Re-Tuning Event
This status is set if Re-Tuning Request in the Present State register changes from 0 to 1.
Host Controller requests Host Driver to perform re-tuning for next data transfer. Current data transfer (not large block count) can be completed without re-tuning.
1 Re-Tuning should be performed
0 Re-Tuning is not required

CARD_INTERRUPT
Writing this bit to 1 does not clear this bit. It is cleared by resetting the SD card interrupt factor. In 1-bit mode, the Host Controller shall detect the Card Interrupt without SD Clock to support wakeup. In 4-bit mode, the card interrupt signal is sampled during the interrupt cycle, so there are some sample delays between the interrupt signal from the SD card and the interrupt to the Host System. It is necessary to define how to handle this delay.
CARD_REMOVAL
This status is set if the Card Inserted in the Present State register changes from 1 to 0. When the Host Driver writes this bit to 1 to clear this status, the status of the Card Inserted in the Present State register should be confirmed. Because the card detect state may possibly be changed when the Host Driver clear this bit and interrupt event may not be generated.
CARD_INSERTION
This status is set if the Card Inserted in the Present State register changes from 0 to 1. When the Host Driver writes this bit to 1 to clear this status, the status of the Card Inserted in the Present State register should be confirmed. Because the card detect state may possibly be changed when the Host Driver clear this bit and interrupt event may not be generated.
BUFFER_READ_READY
This status is set if the Buffer Read Enable changes from 0 to 1. Refer to the Buffer Read Enable in the Present State register.
BUFFER_WRITE_READY
This status is set if the Buffer Write Enable changes from 0 to 1. Refer to the Buffer Write Enable in the Present State register.
DMA_INTERRUPT
This status is set if the Host Controller detects the Host DMA Buffer boundary during transfer. Refer to the Host DMA Buffer Boundary in the Block Size register. Other DMA interrupt factors may be added in the future. This interrupt shall not be generated after the Transfer Complete.
BLOCK_GAP_EVENT
If the Stop At Block Gap Request in the Block Gap Control register is set, this bit is set when both a read / write transaction is stopped at a block gap. If Stop At Block Gap Request is not set to 1, this bit is not set to 1.
XFER_COMPLETE
This bit indicates stop of transaction on three cases:
(1) Completion of a data transfer
(2) Completion of a command pairing with response-with-busy (R1b, R5b)
(3) Stop of data transfer by setting Stop At Block Gap Request in the Block Gap Control register
CMD_COMPLETE
This bit is set when get the end bit of the command response. (Except Auto CMD12)
Refer to Command Inhibit (CMD) in the Present State register.

Offset: 0x30
Read/Write: See table below
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0x00,xxx0,0000,0000)

Bit	R/W	Reset	Description
31:30	RW	0x0	**VEND_SPEC_ERR**: **0** = DISABLE **3** = ENABLE
29	RW	0x0	**SPI_ERR**: **0** = NO_ERR **1** = ERR
28	RW	0x0	**TARGET_RESP_ERROR**: **0** = NO_ERROR **1** = ERROR
27	RW	0x0	**RESP_ERR**: **0** = NO_ERR **1** = ERR
26	RW	0x0	**TUNING_ERR**: **0** = NO_ERR **1** = ERR
25	RW	0x0	**ADMA_ERR**: **0** = NO_ERR **1** = ERR
24	RW	0x0	**AUTO_CMD12_ERR**: **0** = NO_ERR **1** = ERR
23	RW	0x0	**CURRENT_LIMIT_ERR**: **0** = NO_ERR **1** = POWER_FAIL
22	RW	0x0	**DATA_END_BIT_ERR**: **0** = NO_ERR **1** = ERR
21	RW	0x0	**DATA_CRC_ERR**: **0** = NO_ERR **1** = ERR
20	RW	0x0	**DATA_TIMEOUT_ERR**: **0** = NO_ERR **1** = TIMEOUT
19	RW	0x0	**COMMAND_INDEX_ERR**: **0** = NO_ERR **1** = ERR
18	RW	0x0	**COMMAND_END_BIT_ERR**: **0** = NO_ERR **1** = END_BIT_ERR_GENERATED
17	RW	0x0	**COMMAND_CRC_ERR**: **0** = NO_ERR **1** = CRC_ERR_GENERATED

Bit	R/W	Reset	Description
16	RW	0x0	**COMMAND_TIMEOUT_ERR**: **0** = NO_ERR **1** = TIMEOUT
15	RO	0x0	**ERR_INTERRUPT**: **0** = NO_ERR **1** = ERR
13	RO	0x0	**FX_EVENT**: **0** = NO_EVENT **1** = FX_EVENT_DETECTED
12	RO	0x0	**RETUNING_EVENT**: **0** = NO_INT **1** = GEN_INT
8	RO	0x0	**CARD_INTERRUPT**: **0** = NO_INT **1** = GEN_INT
7	RW	0x0	**CARD_REMOVAL**: **0** = NO_INT **1** = GEN_INT
6	RW	0x0	**CARD_INSERTION**: **0** = NO_INT **1** = GEN_INT
5	RW	0x0	**BUFFER_READ_READY**: **0** = NO_INT **1** = GEN_INT
4	RW	0x0	**BUFFER_WRITE_READY**: **0** = NO_INT **1** = GEN_INT
3	RW	0x0	**DMA_INTERRUPT**: **0** = NO_INT **1** = GEN_INT
2	RW	0x0	**BLOCK_GAP_EVENT**: **0** = NO_INT **1** = GEN_INT
1	RW	0x0	**XFER_COMPLETE**: **0** = NO_INT **1** = GEN_INT
0	RW	0x0	**CMD_COMPLETE**: **0** = NO_INT **1** = GEN_INT

SDMMC_INTERRUPT_STATUS_ENABLE_0

Normal Interrupt Status Enable Register

This register is used to select which interrupt status is indicated to the Host System as the

interrupt. These status bits all share the same 1-bit interrupt line. Setting any of these bits to 1 enables interrupt generation.

Offset: 0x34
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,xx00,xxx0,0000,0000)

Bit	Reset	Description
31:30	0x0	**VENDOR_SPECIFIC_ERR**: **0** = DISABLE **3** = ENABLE
29	0x0	**SPI_ERR**: **0** = DISABLE **1** = ENABLE
28	0x0	**TARGET_RESP_ERROR**: **0** = NO_ERROR **1** = ERROR
27	0x0	**RESP_ERROR**: **0** = NO_ERROR **1** = ERROR
26	0x0	**TUNING_ERR**: **0** = DISABLE **1** = ENABLE
25	0x0	**ADMA_ERR**: **0** = DISABLE **1** = ENABLE
24	0x0	**AUTO_CMD12_ERR**: **0** = DISABLE **1** = ENABLE
23	0x0	**CURRENT_LIMIT_ERR**: **0** = DISABLE **1** = ENABLE
22	0x0	**DATA_END_BIT_ERR**: **0** = DISABLE **1** = ENABLE
21	0x0	**DATA_CRC_ERR**: **0** = DISABLE **1** = ENABLE
20	0x0	**DATA_TIMEOUT_ERR**: **0** = DISABLE **1** = ENABLE
19	0x0	**COMMAND_INDEX_ERR**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
18	0x0	**COMMAND_END_BIT_ERR**: **0** = DISABLE **1** = ENABLE
17	0x0	**COMMAND_CRC_ERR**: **0** = DISABLE **1** = ENABLE
16	0x0	**COMMAND_TIMEOUT_ERR**: **0** = DISABLE **1** = ENABLE
13	0x0	**FX_EVENT**: **0** = DISABLE **1** = ENABLE
12	0x0	**RETUNING_EVENT**: **0** = DISABLE **1** = ENABLE
8	0x0	**CARD_INTERRUPT**: **0** = DISABLE **1** = ENABLE
7	0x0	**CARD_REMOVAL**: **0** = DISABLE **1** = ENABLE
6	0x0	**CARD_INSERTION**: **0** = DISABLE **1** = ENABLE
5	0x0	**BUFFER_READ_READY**: **0** = DISABLE **1** = ENABLE
4	0x0	**BUFFER_WRITE_READY**: **0** = DISABLE **1** = ENABLE
3	0x0	**DMA_INTERRUPT**: **0** = DISABLE **1** = ENABLE
2	0x0	**BLOCK_GAP_EVENT**: **0** = DISABLE **1** = ENABLE
1	0x0	**TRANSFER_COMPLETE**: **0** = DISABLE **1** = ENABLE
0	0x0	**COMMAND_COMPLETE**: **0** = DISABLE **1** = ENABLE

SDMMC_INTERRUPT_SIGNAL_ENABLE_0

Normal Interrupt Signal Enable Register

This register is used to select which interrupt status is notified to the Host System as the interrupt. These status bits all share the same 1-bit interrupt line. Setting any of these bits to 1 enables interrupt generation.

Offset: 0x38
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,xx00,xxx0,0000,0000)

Bit	Reset	Description
31:30	0x0	**VENDOR_SPECIFIC_ERR**: **0** = DISABLE **3** = ENABLE
29	0x0	**SPI_ERR**: **0** = DISABLE **1** = ENABLE
28	0x0	**TARGET_RESP_ERROR**: **0** = NO_ERROR **1** = ERROR
27	0x0	**RESP_ERROR**: **0** = DISABLE **1** = ENABLE
26	0x0	**TUNING_ERR**: **0** = DISABLE **1** = ENABLE
25	0x0	**ADMA_ERR**: **0** = DISABLE **1** = ENABLE
24	0x0	**AUTO_CMD12_ERR**: **0** = DISABLE **1** = ENABLE
23	0x0	**CURRENT_LIMIT_ERR**: **0** = DISABLE **1** = ENABLE
22	0x0	**DATA_END_BIT_ERR**: **0** = DISABLE **1** = ENABLE
21	0x0	**DATA_CRC_ERR**: **0** = DISABLE **1** = ENABLE
20	0x0	**DATA_TIMEOUT_ERR**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
19	0x0	**COMMAND_INDEX_ERR**: **0** = DISABLE **1** = ENABLE
18	0x0	**COMMAND_END_BIT_ERR**: **0** = DISABLE **1** = ENABLE
17	0x0	**COMMAND_CRC_ERR**: **0** = DISABLE **1** = ENABLE
16	0x0	**COMMAND_TIMEOUT_ERR**: **0** = DISABLE **1** = ENABLE
13	0x0	**FX_EVENT**: **0** = DISABLE **1** = ENABLE
12	0x0	**RETUNING_EVENT**: **0** = DISABLE **1** = ENABLE
8	0x0	**CARD_INTERRUPT**: **0** = DISABLE **1** = ENABLE
7	0x0	**CARD_REMOVAL**: **0** = DISABLE **1** = ENABLE
6	0x0	**CARD_INSERTION**: **0** = DISABLE **1** = ENABLE
5	0x0	**BUFFER_READ_READY**: **0** = DISABLE **1** = ENABLE
4	0x0	**BUFFER_WRITE_READY**: **0** = DISABLE **1** = ENABLE
3	0x0	**DMA_INTERRUPT**: **0** = DISABLE **1** = ENABLE
2	0x0	**BLOCK_GAP_EVENT**: **0** = DISABLE **1** = ENABLE
1	0x0	**TRANSFER_COMPLETE**: **0** = DISABLE **1** = ENABLE
0	0x0	**COMMAND_COMPLETE**: **0** = DISABLE **1** = ENABLE

SDMMC_AUTO_CMD12_ERR_STATUS_0

Host Control2 Register / Auto CMD Error Status Register

PRESET_VALUE_ENABLE - Preset Value Enable
Host Controller Version 3.00 supports this bit.
As the operating SDCLK frequency and I/O driver strength depend on the Host System implementation, it is difficult to determine these parameters in the Standard Host Driver. When Preset Value Enable is set, automatic SDCLK frequency generation and driver strength selection is performed without considering system specific conditions. This bit enables the functions defined in the Preset Value registers.
1 Automatic Selection by Preset Value are Enabled
0 SDCLK and Driver Strength are controlled by Host Driver
If this bit is set to 0, SDCLK Frequency Select, Clock Generator Select in the Clock Control register and Driver Strength Select in Host Control 2 register are set by Host Driver.
If this bit is set to 1, SDCLK Frequency Select, Clock Generator Select in the Clock Control register and Driver Strength Select in Host Control 2 register are set by Host Controller as specified in the Preset Value registers.
ASYNC_INTR_EN - Asynchronous Interrupt Enable
This bit can be set to 1 if a card support asynchronous interrupt and Asynchronous Interrupt Support is set to 1 in the Capabilities register. Asynchronous interrupt is effective when DAT[1] interrupt is used in 4-bit SD mode (and zero is set to Interrupt Pin Select in the Shared Bus Control register). If this bit is set to 1, the Host Driver can stop the SDCLK during asynchronous interrupt period to save power. During this period, the Host Controller continues to deliver Card Interrupt to the host when it is asserted by the Card.
1 Enabled
0 Disabled
ADDRESSING_64BIT_EN - 64bit addressing enable
Host Controller selects either of 32-bit or 64-bit addressing modes to access system memory. Whether 32-bit or 64-bit is determined by OS installed in a host system. Host Driver sets this bit depends on addressing mode of installed OS. Refer to 64-bit System Address Support in the Capabilities register.
1 64 bits addressing
0 32 bits addressing
HOST_VERSION_4_EN - Host Version 4.00 Enable
This bit selects either Version 3.00 compatible mode or Ver4.00 mode. In Version 4.00, support of 64-bit System Addressing is modified. All DMAs support 64-bit System Addressing. UHS-II supported Host Driver shall

enable this bit.
In Version 4.10, supported 32-bit Block Count for all operations.

Functions of following fields are modified.
(1) SDMA Address
SDMA uses ADMA System Address register (05Fh-058h) instead of
SDMA System Address register (Offset 003-000h)
(2) ADMA2 / ADMA3 Selection
ADMA3/CQE is selected by DMA Select in the Host Control 1 register.
This bit should be set to 1 to use ADMA3/CQE.
(3) 64bit ADMA Descriptor Size
128bit descriptor is used instead of 96-bit descriptor when 64-bit
Addressing is set to 1.
(4) Selection of 32-bit / 64-bit System Addressing
Either 32-bit or 64-bit system addressing is selected by 64-bit
Addressing bit in this register instead of DMA Select in the Host Control 1 register.
(5) 32-bit Block Count
SDMA System Address register (003h-000h) is modified to 32-bit Block Count register.

1 Version 4.00 Mode
0 Version 3.00 Compatible Mode
CMD23_EN
In memory card initialization, Host Driver Version 4.10 checks whether card
supports CMD23 by checking a bit SCR[33]. If the card supports CMD23
(SCR[33]=1), this bit is set to 1. This bit is used to select Auto CMD23 or Auto CMD12 for ADMA3
data transfer. Refer to Auto CMD Enable in the
Transfer Mode register.
ADMA2 Length Mode
This bit selects one of ADMA2 Length Modes either 16-bit or 26-bit.
1 26-bit Data Length Mode
0 16-bit Data Length Mode
UHS2_IF_EN - UHS-II Interface Enable
This bit is used to enable UHS-II Interface. Before trying to start UHS-II
initialization, this bit shall be set to 1. SD 4-bit Interface signals shall be
tri-state (input or bi-directional) or drive to low (output). Before trying to start
SD mode initialization, this bit shall be set to.
1 UHS-II Interface Enabled
0 4-bit SD Interface Enabled
SAMPLING_CLK_SEL - Sampling Clock Select
Host Controller uses this bit to select sampling clock to receive CMD and
DAT. This bit is set by tuning procedure and valid after the completion of
tuning (when Execute Tuning is cleared). Setting 1 means that tuning is
completed successfully and setting 0 means that tuning is failed. Writing 1 to
this bit is meaningless and ignored. A tuning circuit is reset by writing to 0.
This bit can be cleared with setting Execute Tuning. Once the tuning circuit
is reset, it will take time to complete tuning sequence. Therefore, Host Driver
should keep this bit to 1 to perform re-tuning sequence to compete re-tuning

sequence in a short time. Change of this bit is not allowed while the Host Controller is receiving response or a read data block.
1 Tuned clock is used to sample data
0 Fixed clock is used to sample data
EXECUTE_TUNING - Execute Tuning
This bit is set to 1 to start tuning procedure and automatically cleared when tuning procedure is completed. The result of tuning is indicated to Sampling Clock Select. Tuning procedure is aborted by writing 0.
for more detail about tuning procedure.
1 Execute Tuning
0 Not Tuned or Tuning Completed
DRIVE_STRENGTH_SEL - Driver Strength Select
Host Controller output driver in 1.8V signaling is selected by this bit. In 3.3V signaling, this field is not effective. This field can be set depends on Driver Type A, C and D support bits in the Capabilities register.
This bit depends on setting of Preset Value Enable.
If Preset Value Enable = 0, this field is set by Host Driver.
If Preset Value Enable = 1, this field is automatically set by a value specified in the one of Preset Value registers.
00b Driver Type B is Selected (Default)
01b Driver Type A is Selected
10b Driver Type C is Selected
11b Driver Type D is Selected
VOLT_18_EN - 1.8V Signaling Enable
This bit controls voltage regulator for I/O cell. 3.3V is supplied to the card regardless of signaling voltage.
Setting this bit from 0 to 1 starts changing signal voltage from 3.3V to 1.8V. 1.8V regulator output shall be stable within 5ms. Host Controller clears this bit if switching to 1.8V signaling fails.
Clearing this bit from 1 to 0 starts changing signal voltage from 1.8V to 3.3V. 3.3V regulator output shall be stable within 5ms.
Host Driver can set this bit to 1 when Host Controller supports 1.8V signaling (One of support bits is set to 1: SDR50, SDR104 or DDR50 in the Capabilities register) and the card or device supports UHS-I (S18R=1. Refer to Bus Signal Voltage Switch Sequence in the Physical Layer Specification Version 3.00).
1 1.8V Signaling
0 3.3V Signaling
UHS_MODE_SEL - UHS Mode Select
This field is used to select one of UHS-I modes and effective when 1.8V Signaling Enable is set to 1.
If Preset Value Enable in the Host Control 2 register is set to 1, Host Controller sets SDCLK Frequency Select, Clock Generator Select in the Clock Control register and Driver Strength Select according to Preset Value registers. In this case, one of preset value registers is selected by this field. Host Driver needs to reset SD Clock Enable before changing this field

to avoid generating clock glitch. After setting this field, Host Driver sets SD Clock Enable again.
000b SDR12
001b SDR25
010b SDR50
011b SDR104(SD/SDIO)/HS200(eMMC)
100b DDR50(SD/SDIO)/DDR52(eMMC)
101b HS400(eMMC)
110b Reserved
111b UHS2
When SDR50, SDR104 or DDR50 is selected for SDIO card, interrupt detection at the block gap shall not be used. Read Wait timing is changed for these modes. Refer to the SDIO Specification Version 3.00 for more detail
COMMAND_NOT_ISSUED - Command Not Issued By Auto CMD12 Error
Setting this bit to 1 means CMD_wo_DAT is not executed due to an Auto CMD12 Error (D04-D01) in this register.
INDEX_ERR - Auto CMD12 Index Error
This bit is set if the Command Index error occurs in response to a command.
END_BIT_ERR - Auto CMD12 End Bit Error
This bit is set when detecting that the end bit of command response is 0.
CRC_ERR - Auto CMD12 CRC Error
This bit is set when detecting a CRC error in the command response.
TIMEOUT_ERR - Auto CMD12 Timeout Error
This bit is set if no response is returned within 64 SDCLK cycles from the end bit of command. If this bit is set to 1, the other error status bits (D04-D02) are meaningless.
NOT_EXECUTED - Auto CMD12 Not Executed
If memory multiple block data transfer is not started due to command error, this bit is not set because it is not necessary to issue Auto CMD12. Setting this bit to 1 means the Host Controller cannot issue Auto CMD12 to stop memory multiple block data transfer due to some error. If this bit is set to 1, other error status bits (D04-D01) are meaningless.
The relation between Auto CMD12 CRC Error and Auto CMD12 Timeout Error is shown below

Auto CMD12 CRC Error	Auto CMD12 Timeout Error	Kinds of error
0	0	No Error
0	1	Response Timeout Error
1	0	Response CRC Error
1	1	CMD line conflict

RESP_ERR - Auto CMD Response Error
This bit is set when Response Error Check Enable in the Transfer Mode register is set to 1 and an error is detected in either R1 response of either Auto CMD12 or Auto CMD23.

1 Error
0 No Error

Offset: 0x3c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,00x0,0000,0000,xxxx,xxxx,0x00,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PRESET_VALUE_ENABLE**: **0** = HW_SEL **1** = SW_SEL
30	RW	0x0	**ASYNC_INTR_EN**: **0** = DISABLE **1** = ENABLE
29	RW	0x0	**ADDRESSING_64BIT_EN**: **0** = DISABLE **1** = ENABLE
28	RW	0x0	**HOST_VERSION_4_EN**: **0** = DISABLE **1** = ENABLE
27	RW	0x0	**CMD23_EN**: **0** = DISABLE **1** = ENABLE
26	RW	0x0	**ADMA2_LEN_MODE**: **0** = LEN_16BIT **1** = LEN_26BIT
24	RW	0x0	**UHS2_IF_EN**: **0** = DISABLE **1** = ENABLE
23	RW	0x0	**SAMPLING_CLK_SEL**: **0** = FIXED **1** = TUNED
22	RW	0x0	**EXECUTE_TUNING**: **0** = NOT_TUNED **1** = EXECUTE
21:20	RW	0x0	**DRIVE_STRENGTH_SEL**: **0** = TYPE_B **1** = TYPE_A **2** = TYPE_C **3** = TYPE_D
19	RW	0x0	**VOLT_18_EN**: **0** = V33 **1** = V18

Bit	R/W	Reset	Description
18:16	RW	0x0	**UHS_MODE_SEL**: The following table shows the PROD mnemonic mapping to different SD and eMMC speed modes |-------------|--------------|--------------------| |PROD mnemonic | SD speed mode | eMMC speed mode | |-------------|--------------|--------------------| |prod_c_ds | DS | Legacy (SDR at 26M) | |prod_c_hs | HS | NA | |prod_c_sdr12 | SDR12 | NA | |prod_c_sdr25 | SDR25 | NA | |prod_c_sdr50 | SDR50 | SDR52 | |prod_c_sdr104 | SDR104 | NA | |prod_c_ddr52 | DDR50 | DDR52 | |prod_c_hs200 | NA | HS200 | |prod_c_hs400 | NA | HS400 | |----------| **0** = SDR12 **1** = SDR25 **2** = SDR50 **3** = SDR104 **4** = DDR50 **5** = HS400 **6** = RSVD **7** = UHS2
7	RO	0x0	**COMMAND_NOT_ISSUED**: **0** = NO_ERR **1** = NOT_ISSUED
5	RO	0x0	**RESP_ERR**: **0** = NO_ERR **1** = ERR
4	RO	0x0	**INDEX_ERR**: **0** = NO_ERR **1** = ERR
3	RO	0x0	**END_BIT_ERR**: **0** = NO_ERR **1** = END_BIT_ERR_GENERATED
2	RO	0x0	**CRC_ERR**: **0** = NO_ERR **1** = CRC_ERR_GENERATED
1	RO	0x0	**TIMEOUT_ERR**: **0** = NO_ERR **1** = TIMEOUT
0	RO	0x0	**NOT_EXECUTED**: **0** = EXECUTED **1** = NOT_EXECUTED

SDMMC_CAPABILITIES_0

Lower Capabilities Register

SLOT_TYPE

00b Removable Card Slot
01b Embedded Slot for One Device
10b Shared Bus Slot
11b UHS-II Multiple Embedded Devices
ASYNC_INTR
1 Asynchronous Interrupt Supported
0 Asynchronous Interrupt Not Supported
SYSTEM_BUS_64BIT_SUPPORT - 64-bit System Bus Support for V3
Meaning of this bit is different depends on Versions
Host Controller Version 3.00 and Ver4.10 use this bit as 64-bit System Address support for V3 mode.
Host Controller Version 4.00 uses this bit as 64-bit System Address support for both V3 and V4 modes.
SDMA cannot be used in 64-bit Addressing in Version 3 mode.
If this bit is set to 1, 64-bit ADMA2 with using 96-bit Descriptor may be enabled as follows:
In case of Host Controller Version 3, 64-bit ADMA2 is enabled by DMA Select =11b in the Host Control 1 register.
In case of Host Controller Version 4, 64-bit ADMA2 for Version 3 is enabled by setting Host Version 4 Enable =0 and DMA Select = 11b.

1 64-bit System Address for V3 is Supported
0 64-bit System Address for V3 is not Supported

SYSTEM_BUS_64BIT_SUPPORT_V4 - 64-bit System Bus Support for V4
This bit is added from Version 4.10. Setting 1 to this bit indicates that the Host Controller supports 64-bit System Addressing of Version 4 mode
When this bit is set to 1, full or a part of 64-bit address should be used to decode Host Controller Registers so that Host Controller Registers can be placed above system memory area. 64-bit address decode of Host Controller Registers is effective regardless of setting to 64bit Addressing in Host Control 2.
If this bit is set to 1, 64-bit DMA Addressing for Version 4 is enabled by setting Host Version 4 Enable =1,
64-bit Addressing =1 in the Host Control 2 register. SDMA can be used and ADMA2 uses 128-bit Descriptor.

1 64-bit System Address for V4 is Supported
0 64-bit System Address for V4 is not Supported

VOLTAGE_SUPPORT_1_8_V - Voltage Support 1.8V,The Voltage Support to Card is dependent on System & Slot. The platfrom datasheet has this.
VOLTAGE_SUPPORT_3_0_V - Voltage Support 3.0V,The Voltage Support to Card is dependent on System & Slot. The platfrom datasheet has this.
VOLTAGE_SUPPORT_3_3_V - Voltage Support 3.3V,The Voltage Support to Card is dependent on System & Slot. The platfrom datasheet has this.
SUSPEND_RESUME_SUPPORT - Suspend/Resume Support
This bit indicates whether the Host Controller supports Suspend / Resume functionality. If this bit is 0, the Host Driver shall not issue either Suspend or

Resume commands because the Suspend and Resume mechanism (Refer to 1.6) is not supported.
DMA_SUPPORT - SDMA Support
This bit indicates whether the Host Controller is capable of using SDMA to transfer data between system memory and the Host Controller directly. Version 4.10 Host Controller shall support SDMA if ADMA2 is supported.
HIGH_SPEED_SUPPORT - High Speed Support
This bit indicates whether the Host Controller and the Host System support High Speed mode and they can supply SD Clock frequency from 25MHz to 50MHz.
ADMA1_SUPPORT - ADMA1 Support
This bit indicates whether the Host Controller is capable of using ADMA1.
ADMA2_SUPPORT - ADMA2 Support
This bit indicates whether the Host Controller is capable of using ADMA2. Version 4.10 Host Controller shall support ADMA2 if ADMA3 is supported.
EXTENDED_MEDIA_BUS_SUPPORT
Setting to 1,indicates 8-bit data bus is supported.
MAX_BLOCK_LENGTH - Max Block Length
This value indicates the maximum block size that the Host Driver can read and write to the buffer in the Host Controller. The buffer shall transfer this block size without wait cycles. Three sizes can be defined as indicated below. It is noted that transfer block length shall be always 512 bytes for SD Memory Cards regardless this field.
BASE_CLOCK_FREQUENCY - Base Clock Frequency For SD Clock
This value indicates the base (maximum) clock frequency for the SD Clock. Unit values are 1MHz. If the real frequency is 16.5MHz, the lager value shall be set 01 0001b (17MHz) because the Host Driver use this value to calculate the clock divider value (Refer to the SDCLK Frequency Select in the Clock Control register.) and it shall not exceed upper limit of the SD Clock frequency. The supported clock range is 10MHz to 63MHz. If these bits are all 0, the Host System has to get information via another method.
Not 0 - 1MHz to 63MHz
000000b- Get information via another method
TIMEOUT_CLOCK_UNIT - Timeout Clock Unit
This bit shows the unit of base clock frequency used to detect Data Timeout Error.
0 KHz
1 MHz
TIMEOUT_CLOCK_FREQUENCY - Timeout Clock Frequency
This bit shows the base clock frequency used to detect Data Timeout Error. The Timeout Clock Unit defines the unit of this fields value.
Timeout Clock Unit =0 [KHz] unit: 1KHz to 63KHz
Timeout Clock Unit =1 [MHz] unit: 1MHz to 63MHz
Not 0 - 1KHz to 63KHz or 1MHz to 63MHz
000000b - Get information via another method

Offset: 0x40
Read/Write: RO

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x3f6cd08c (0b0011,1111,011x,1100,1101,0000,1x00,1100)

Bit	Reset	Description
31:30	0x0	**SLOT_TYPE**: **0** = REMOVABLE **1** = EMBEDDED **2** = SHARED **3** = UHS2_MULTIPLE_EMBEDDED
29	0x1	**ASYNC_INTR**: **0** = NOT_SUPPORTED **1** = SUPPORTED
28	0x1	**SYSTEM_BUS_64BIT_SUPPORT**: **0** = NOT_SUPPORTED **1** = SUPPORTED
27	0x1	**SYSTEM_BUS_64BIT_SUPPORT_V4**: **0** = NOT_SUPPORTED **1** = SUPPORTED
26	0x1	**VOLTAGE_SUPPORT_1_8_V**: **0** = NOT_SUPPORTED **1** = SUPPORTED
25	0x1	**VOLTAGE_SUPPORT_3_0_V**: **0** = NOT_SUPPORTED **1** = SUPPORTED
24	0x1	**VOLTAGE_SUPPORT_3_3_V**: **0** = NOT_SUPPORTED **1** = SUPPORTED
23	0x0	**SUSPEND_RESUME_SUPPORT**: **0** = NOT_SUPPORTED **1** = SUPPORTED
22	0x1	**DMA_SUPPORT**: **0** = NOT_SUPPORTED **1** = SUPPORTED
21	0x1	**HIGH_SPEED_SUPPORT**: **0** = NOT_SUPPORTED **1** = SUPPORTED
19	0x1	**ADMA2_SUPPORT**: **0** = NOT_SUPPORTED **1** = SUPPORTED
18	0x1	**EXTENDED_MEDIA_BUS_SUPPORT**: **0** = NOT_SUPPORTED **1** = SUPPORTED
17:16	0x0	**MAX_BLOCK_LENGTH**: **0** = BYTE512 **1** = BYTE1024 **2** = BYTE2048 **3** = RESERVED

Bit	Reset	Description
15:8	0xd0	**BASE_CLOCK_FREQUENCY**
7	0x1	**TIMEOUT_CLOCK_UNIT**: MHz **0** = KHZ **1** = MHZ
5:0	0xc	**TIMEOUT_CLOCK_FREQUENCY**: 12MHz TMCLK is used in legacy SD/eMMC mode TMCLK freq value will be advertised based on UHS2_IF_EN and USE_TMCLK_FOR_DATA_TIMEOUT

SDMMC_CAPABILITIES_HIGHER_0

Higher Capabilities Register

1.8V VDD2 Support
This bit indicates that support of VDD2 on the Host System.
0b 1.8V VDD2 is not supported
1b 1.8V VDD2 is supported
ADMA3_SUPPORT - ADMA3 Support
This bit indicates that support of ADMA3 on Host Controller.
0b ADMA3 is not supported
1b ADMA3 is supported
Clock Multiplier
This field indicates clock multiplier value of programmable clock generator.
Refer to Clock Control register. Setting 00h means that Host Controller
does not support programmable clock generator.
00h Clock Multiplier is Not Supported
01h Clock Multiplier M = 2
02h Clock Multiplier M = 3
....
Timer Count for Re-Tuning
0h Re-Tuning timer disabled
1h 1 seconds
2h 2 seconds
3h 4 seconds
. .....
. .....
nh 2**(n-1) seconds
. .....
. .....
Fh Get information from other source
Driver Type D Support
This bit indicates support of Driver Type D for 1.8 Signaling.
1 Driver Type D is Supported

0 Driver Type D is Not Supported
Driver Type C Support
This bit indicates support of Driver Type C for 1.8 Signaling.
1 Driver Type C is Supported
0 Driver Type C is Not Supported
Driver Type A Support
This bit indicates support of Driver Type A for 1.8 Signaling.
1 Driver Type A is Supported
0 Driver Type A is Not Supported
UHS2_SUPPORT - UHS-II Support (UHS-II only)
This bit indicates whether Host Controller supports UHS-II. If this bit is set to 1, 1.8V VDD2 Support shall be set to 1 (Host System shall support VDD2 power supply).
1 UHS-II is supported
0 UHS-II is not supported
DDR50 Support
1 DDR50 is Supported
0 DDR50 is Not Supported
SDR104 Support
SDR104 requires tuning.
1 SDR104 is Supported
0 SDR104 is Not Supported
SDR50 Support
If SDR104 is supported, this bit shall be set to 1. Bit 40 indicates whether SDR50 requires tuning or not.
1 SDR50 is Supported
0 SDR50 is Not Supported

Offset: 0x44
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x18002f73 (0bxxx1,1xxx,0000,0000,001x,1111,x111,0011)

Bit	Reset	Description
28	0x1	**VDD2_1_8V_SUPPORT**: **0** = NOT_SUPPORTED **1** = SUPPORTED
27	0x1	**ADMA3_SUPPORT**: **0** = NOT_SUPPORTED **1** = SUPPORTED
23:16	0x0	**CLOCK_MULTIPLIER**

Bit	Reset	Description
15:14	0x0	**RETUNING_MODES**: **0** = MODE1 **1** = MODE2 **2** = MODE3 **3** = MODE4
13	0x1	**SDR50_TUNING**: **0** = NOT_REQUIRED **1** = REQUIRED
11:8	0xf	**RETUNING_TIMER_COUNT**
6	0x1	**TYPE_D_DRIVER**: **0** = NOT_SUPPORTED **1** = SUPPORTED
5	0x1	**TYPE_C_DRIVER**: **0** = NOT_SUPPORTED **1** = SUPPORTED
4	0x1	**TYPE_A_DRIVER**: **0** = NOT_SUPPORTED **1** = SUPPORTED
3	0x0	**UHS2_SUPPORT**: **0** = NOT_SUPPORTED **1** = SUPPORTED
2	0x0	**DDR50**: **0** = NOT_SUPPORTED **1** = SUPPORTED
1	0x1	**SDR104**: **0** = NOT_SUPPORTED **1** = SUPPORTED
0	0x1	**SDR50**: **0** = NOT_SUPPORTED **1** = SUPPORTED

SDMMC_MAXIMUM_CURRENT_0

Maximum Current Capabilities Register

Offset: 0x48
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:16	0x0	**MAXIMUM_CURRENT_FOR_1_8V**: Maximum Current for 1.8V VDD1

Bit	Reset	Description
15:8	0x0	**MAXIMUM_CURRENT_FOR_3_0V**: Maximum Current for 3.0V VDD1
7:0	0x0	**MAXIMUM_CURRENT_FOR_3_3V**: Maximum Current for 3.3V VDD1

SDMMC_MAXIMUM_CURRENT_HI_0

Maximum Current Capabilities2 Register

Offset: 0x4c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**MAXIMUM_CURRENT_FOR_1_8V_VDD2**: Maximum Current for 1.V VDD2

SDMMC_FORCE_EVENT_0

Force Event for Auto CMD12 Error Status Register

The Force Event Register is not a physically implemented register. Rather, it is an address at which the
Auto CMD12 Error Status Register can be written.
Writing 1 : set each bit of the Auto CMD12 Error Status Register
Writing 0 : no effect

Offset: 0x50
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,xxxx,xxxx,0x00,0000)

Bit	Reset	Description
31:30	0x0	**VENDOR_SPECIFIC_ERR_STATUS**: **0** = DISABLE **3** = ENABLE

Bit	Reset	Description
28	0x0	**TARGET_RESP_ERROR**: **0** = NO_ERROR **1** = ERROR
27	0x0	**RESP_ERROR**: **0** = NO_ERROR **1** = ERROR
26	0x0	**TUNING_ERR**: **0** = DISABLE **1** = ENABLE
25	0x0	**ADMA_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
24	0x0	**AUTOCMD12_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
23	0x0	**CURRENTLIMIT_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
22	0x0	**DATA_END_BIT_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
21	0x0	**DATACRC_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
20	0x0	**DATATIMEOUT_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
19	0x0	**COMMAND_INDEX_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
18	0x0	**COMMAND_END_BIT_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
17	0x0	**COMMAND_CRC_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
16	0x0	**COMMAND_TIMEOUT_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
7	0x0	**AUTO_CMD12_NOT_ISSUED**: **0** = NO_INTERRUPT **1** = INTERRUPT
5	0x0	**AUTO_CMD12_RESP_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT

Bit	Reset	Description
4	0x0	**AUTO_CMD12_INDEX_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
3	0x0	**AUTO_CMD12_END_BIT_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
2	0x0	**AUTO_CMD12_CRC_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
1	0x0	**AUTO_CMD12_TIMEOUT_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
0	0x0	**AUTO_CMD12_NOT_EXECUTED**: **0** = NO_INTERRUPT **1** = INTERRUPT

SDMMC_ADMA_ERR_STATUS_0

ADMA Error Status Register

ADMA_LENGTH_MISMATCH_ERR - ADMA Length Mismatch Error
This error occurs in the following 2 cases.
(1) While Block Count Enable being set, the total data length specified by the Descriptor table is different from that specified by the Block Count and Block Length.
(2) Total data length can not be divided by the block length.
ADMA_ERR_STATE - ADMA Error State
This field indicates the state of ADMA when error is occurred during ADMA data transfer. This field never indicates "10" because ADMA never stops in this state.

|D01 - D00 |ADMA Error State when | Contents of SYS_SDR register |
| | error is occurred | |
---|
| 00 | ST_STOP (Stop DMA) | Points next of the error descriptor|
| 01 | ST_FDS (Fetch Descriptor) | Points the error descriptor |
| 10 | Never set this state | (Not used) |
| 11 | ST_TFR (Transfer Data) | Points the next of the error |
| | | descriptor |
---|

Offset: 0x54
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ADMA_LENGTH_MISMATCH_ERR**: **0** = NO_ERR **1** = ERR
1:0	0x0	**ADMA_ERR_STATE**

SDMMC_ADMA_SYSTEM_ADDRESS_0

ADMA System Address Register
The 32-bit addressing Host Driver uses lower 32-bit of this register (upper 32-bit should be set to 0) and shall program Descriptor Table on 32-bit boundary and set 32-bit boundary address to this register. DMA2/3 ignores lower 2-bit of this register and assumes it to be 00b.
DMA in 64-bit addressing. The 64-bit addressing Host Driver uses all bits of this register and shall program Descriptor Table on 64-bit boundary and set 64-bit boundary address to this register. DMA2/3 ignores lower 3-bit of this register and assumes it to be 000b.
(1) SDMA
If Host Version 4.00 Enable is set to 1, SDMA use this register to indicate System Address of data location instead of using SDMA System Address register (Offset 003-000h). SDMA can be used in 32-bit and 64-bit addressing in Version 4.00.
(2) ADMA2
This register holds byte address of executing command of the Descriptor table. At the start of
ADMA2, the Host Driver shall set start address of the Descriptor table. The ADMA increments this register address, which points to next line, when every fetching a Descriptor line. When the ADMA Error Interrupt is generated, this register shall hold the Descriptor address depending on the ADMA state.
(3) ADMA3
This register is set by ADMA3. Host Driver is not necessary to set this register. The ADMA3 increments address of this register, which points to next line, when every time fetching a Descriptor line. When Error Interrupt is generated, this register shall hold the Descriptor address depending on the ADMA state.

Offset: 0x58
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ADMA_SYSTEM_ADDRESS**

SDMMC_UPPER_ADMA_SYSTEM_ADDRESS_0

Upper ADMA System Address Register

This register is used by 64-bit address descriptor.
Upper bits of 64bit address - ADMA system address[63:32] is set in this register

Offset: 0x5c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UPPER_ADMA_SYSTEM_ADDRESS**

SDMMC_PRESET_DEFAULT_AND_INIT_0

Preset Value Register Indexes

15-14 HwInit Driver Strength Select Value
Driver Strength is supported by 1.8V signaling bus speed modes. This field
is meaningless for 3.3V signaling.
11b Driver Type D is Selected
10b Driver Type C is Selected
01b Driver Type A is Selected
00b Driver Type B is Selected
13-11 Rsvd Reserved
Clock Generator Select Value
This bit is effective when Host Controller supports programmable clock
generator.
1 Programmable Clock Generator
0 Host Controller Ver2.00 Compatible Clock Generator
SDCLK Frequency Select Value
10-bit preset value to set SDCLK Frequency Select in the Clock Control
Register is described by a host system.
Preset Value for Default Speed and Initialization

Offset: 0x60
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00040000 (0b00xx,x000,0000,0100,00xx,x000,0000,0000)

Bit	Reset	Description
31:30	0x0	**DRIVE_STRENGTH_VAL_HIGH**
26	0x0	**CLK_GEN_VAL_HIGH**
25:16	0x4	**SDCLK_FREQ_VAL_HIGH**
15:14	0x0	**DRIVE_STRENGTH_VAL_LOW**
10	0x0	**CLK_GEN_VAL_LOW**
9:0	0x0	**SDCLK_FREQ_VAL_LOW**

SDMMC_PRESET_SDR12_AND_HIGH_0

Preset Value for SDR12 Speed and HIGHSPEED

Offset: 0x64
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00040002 (0b00xx,x000,0000,0100,00xx,x000,0000,0010)

Bit	Reset	Description
31:30	0x0	**DRIVE_STRENGTH_VAL_HIGH**
26	0x0	**CLK_GEN_VAL_HIGH**
25:16	0x4	**SDCLK_FREQ_VAL_HIGH**
15:14	0x0	**DRIVE_STRENGTH_VAL_LOW**
10	0x0	**CLK_GEN_VAL_LOW**
9:0	0x2	**SDCLK_FREQ_VAL_LOW**

SDMMC_PRESET_SDR50_AND_SDR25_0

Preset Value for SDR50 and SDR25

Offset: 0x68
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00010002 (0b00xx,x000,0000,0001,00xx,x000,0000,0010)

Bit	Reset	Description
31:30	0x0	**DRIVE_STRENGTH_VAL_HIGH**
26	0x0	**CLK_GEN_VAL_HIGH**
25:16	0x1	**SDCLK_FREQ_VAL_HIGH**
15:14	0x0	**DRIVE_STRENGTH_VAL_LOW**
10	0x0	**CLK_GEN_VAL_LOW**
9:0	0x2	**SDCLK_FREQ_VAL_LOW**

SDMMC_PRESET_DDR50_AND_SDR104_0

Preset Value for DDR50 and SDR104

Offset: 0x6c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00020000 (0b00xx,x000,0000,0010,00xx,x000,0000,0000)

Bit	Reset	Description
31:30	0x0	**DRIVE_STRENGTH_VAL_HIGH**
26	0x0	**CLK_GEN_VAL_HIGH**
25:16	0x2	**SDCLK_FREQ_VAL_HIGH**
15:14	0x0	**DRIVE_STRENGTH_VAL_LOW**
10	0x0	**CLK_GEN_VAL_LOW**
9:0	0x0	**SDCLK_FREQ_VAL_LOW**

SDMMC_ADMA3_INT_DESC_LOWER_ADDRESS_0

ADMA3 Integrated Descriptor Address 31:0 Register .

The start address of Integrated DMA Descriptor is set to this register. Writing to a specific address starts ADMA3 depends on 32-bit/64-bit addressing. The ADMA3 fetches one Descriptor Address and increments this field to indicate the next Descriptor address.
The 32-bit addressing Host Driver uses lower 32-bit of this register and shall program Descriptor Table on 32-bit boundary. ADMA3 ignores lower 2-bit of this register and assumes it to be 00b. Writing to 07Bh starts ADMA3 data transfer.
The 64-bit addressing Host Driver uses all 64-bit of this register and shall program Descriptor Table on 64-bit boundary. ADMA3 ignores lower 3-bit of this register and assumes it to be 000b. Writing to 07Fh starts ADMA3 data transfer.

Register Value Addressing Mode
00000000_xxxxxxxxh 32-bit System Address
xxxxxxxx_xxxxxxxxh 64-bit System Address

Offset: 0x78
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ADDRESS_31_0**

SDMMC_ADMA3_INT_DESC_UPPER_ADDRESS_0

ADMA3 Integrated Descriptor Address 63:32 Register .

Offset: 0x7c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ADDRESS_63_32**

SDMMC_VENDOR_REGS_PTR_0

Pointer Registers to 1FFh-100h Area - vendor specific area
Area of offset mFFh-m00h is defined as re-locatable area. The locations of following register sets are pointed by offset address.
vendor registers start at 100h. => m=1
Vendor registers pointer

Offset: 0xe8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000100 (0bxxxx,xxxx,xxxx,xxxx,xxxx,0001,0000,0000)

Bit	Reset	Description
11:0	0x100	**VENDOR_PTR**: Vendor regs start at 0x100 - offset[11:0]=0x100

SDMMC_SLOT_INTERRUPT_STATUS_0

Slot Interrupt Status Register

VENDOR_VERSION_NUMBER - Vendor Version Number
This status is reserved for the vendor version number. The Host Driver should not use this status.
SPECIFICATION_VERSION_NUMBER - Specification Version Number
This status indicates the Host Controller Spec. Version. The upper and lower 4-bits indicate the version.
00 SD Host Specification Version 1.00
01 SD Host Specification Version 2.00
Including the feature of the ADMA and Test Register,
02 SD Host Specification Version 3.00
03 SD Host Specification Version 4.00
04h SD Host Controller Specification Version 4.10
05h SD Host Controller Specification Version 4.20
others Reserved
INTERRUPT_SIGNAL_FOR_EACH_SLOT - Interrupt Signal For Each Slot
These status bits indicate the logical OR of Interrupt Signal and Wakeup Signal for each slot. A maximum of 8 slots can be defined. If one interrupt signal is associated with multiple slots, the Host Driver can know which interrupt is generated by reading these status bits. By a power on reset or by setting Software Reset For All, the interrupt signal shall be de-asserted and this status shall read 00h.
Bit 00 Slot 1
Bit 01 Slot 2
Bit 02 Slot 3

....
Bit 07 Slot 8

Offset: 0xfc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x05050000 (0b0000,0101,0000,0101,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31:24	0x5	**VENDOR_VERSION_NUMBER**
23:16	0x5	**SPECIFICATION_VERSION_NUMBER**
7:0	0x0	**INTERRUPT_SIGNAL_FOR_EACH_SLOT**

SDMMC_VENDOR_CLOCK_CNTRL_0

The following Registers are Vendor Specific Registers and are mapped to Vendor Specific Address Space(0x100 - 0x1FF)

Offset: 0x100
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0804d06d (0bxx00,1000,0000,0100,1101,0000,x110,1101)
PROD: 0x05090020 (0bxxx0,0101,0000,1001,xxxx,xxxx,x01x,0x0x)

Bit	Reset	PROD	Description
29	0x0	_NONE_	**DIFF_CLK_SEL**: If set, selects differential CLK and DQS; Used by eMMC IOBRICK only. SW should set this appropriately based on eMMC part used. E_INPUT_* (CMD/DAT/CLK) of emmc IOBRICK should be set to 0 when differential signaling is used. default is '0' - selects single ended signaling for clk/dqs
28:24	0x8	0x5	**TRIM_VAL**: Trimmer tap value for the output data path trimmer This determines the trimmer value needed to drive the output data correctly. The tap for outbound trimmer is single MUX. The trim settings required are within very small (0-3) with absolute delay requirement of ~400ps. Minimal change with PVT variations.

Bit	Reset	PROD	Description
23:16	0x4	0x9	**TAP_VAL**: Tap value for input data path trimmer This determines the tap value needed to sample the input data correctly. Delay per each tap can range from 70ps (hv_ff) to 505ps (lv_ss).
15:8	0xd0	_NONE_	**BASE_CLK_FREQ**: SW driver should write core clock frequency value in MHz to this field to advertise base frequency in SDMMCA_CAPABILITIES_0_ BASE_CLOCK_FREQUENCY for standard SD driver usage.
6	0x1	NORMAL	**LEGACY_CLKEN_OVERRIDE**: Override for sdmmc_legacy_g_clk clken; Set this to 0 to save power **0** = NORMAL :0 -> sdmmc_legacy_g_clk is gated **1** = OVERRIDE :1 -> sdmmc_legacy_g_clk is not gated
5	0x1	OVERRIDE	**SDR50_TUNING_OVERRIDE**: override the SDR50_TUNING capabilities bit. Software should only set this bit if it is required to use Tuning for SDR50. (only supported for SDMMC1) **0** = NORMAL :0 -> No Tuning support advertised for SDR50 mode. **1** = OVERRIDE :1 -> Tuning support is enabled for SDR50 mode.
4	0x0	_NONE_	**UHS2_CAPABILITY_OVERRIDE**: override the UHS-II capabilities bit.
3	0x1	0x0	**PADPIPE_CLKEN_OVERRIDE**: Override for padmacro and pipemacro clken. **0** = NORMAL :0 -> CLKEN is de-asserted when internal CLKEN is de-asserted. **1** = OVERRIDE :1 -> CLKEN is kept asserted even when internal CLKEN is de-asserted.
2	0x1	_NONE_	**SPI_MODE_CLKEN_OVERRIDE**: This mode is not supported.
1	0x0	FEEDBACK	**INPUT_IO_CLK**: Feedback clock is selected by default. Software should not change this. Disabling Feedback clock will select Internal Clock that requires different TAP Value Programming. **0** = FEEDBACK **1** = INTERNAL

Bit	Reset	PROD	Description
0	0x1	_NONE_	**SDMMC_CLK**: This is set when sdmmc_clk is supplied by the CAR module. Prior to sdmmc_clk switch OFF, this bit should be written as '0'. By writing zero, the asynchronous card interrupt is routed to the Interrupt controller. **0** = DISABLE **1** = ENABLE

SDMMC_VENDOR_SYS_SW_CNTRL_0

Offset: 0x104
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x38600002 (0b0011,1000,011x,xxxx,x000,0000,x000,0010)

Bit	Reset	Description
31	0x0	**ENHANCED_STROBE_MODE**: Enables enhanced strobe mode in HS400 mode for eMMC5.x devices 0 - cmd_in(Resp) is sampled by loopback clock which requires tuning 1 - cmd_in(Resp) is sampled by DQS_in which requires no tuning SW has to set this bit appropriately based on device capability since this is an optional feature for eMMC5.x devices. If device supports this feature, SW should set this bit to avoid tuning.
30	0x0	**USE_TMCLK_FOR_WR_CRC_STATUS_TIMEOUT**: When set, uses TMCLK data timeout counter for generating wr_crc_status data-timeout When cleared, uses sdmmc_clk for maintaining wr_crc_status data timeout counter
29	0x1	**USE_NCRC_FOR_WR_CRC_STATUS_TIMEOUT_VAL**: This field is valid only when USE_TMCLK_FOR_WR_CRC_STATUS_TIMEOUT is set to 0. When cleared, uses data timeout value as wr crc status timeout value (spec defined one) When set, uses Ncrc cycles as timeout value
28	0x1	**USE_TMCLK_FOR_DATA_TIMEOUT**: When set, uses TMCLK data timeout counter for generating legacy data timeout error (except wr_crc_status timeout) When cleared, uses sdmmc_clk for maintaining data timeout counter
27:24	0x8	**DEVICE_BUSY_WAIT_CYCLES**: This register field is used to load wait_cycles counter before device busy sampling in HS400 mode. Please note that this counter is used only in HS400 mode.
23	0x0	**ALLOW_CARD_CLK_STALLS_IN_WR**: When set, allows card clock stopping during transfer of data within a block in DDR52/HS400 writes.

Bit	Reset	Description
22	0x1	**EMMC_IOBRICK_CLK_DATA**: Used to drive AP_CLK and AN_CLK input of iobrick. 0x1 - clk_out will be same as iobrick_clk_in 0x0 - clk_out will be inverted iobrick_clk_in
21	0x1	**QUALIFY_WITH_RD_DATA_VLD**: We have async FIFOs in both cmd_in and dat_in paths in padmacro which are used in tunable modes. When this bit set, rdata from FIFO is treated as valid data only when rd_req is high. This is needed to handle bubbles on 'rd_req' when MTBF is high.
14	0x0	**SD_BUS_POWER_ON_OFF_INT_STATUS**: SD_BUS_POWER was changed. System software can use this interrupt to implement power switch.
13	0x0	**VOLT_SWITCH_INT_STATUS**: VOLT_18_EN was changed. System software can use this interrupt to implement a UHS-I voltage switch procedure for a standard SD Host driver **0** = NO_INT **1** = GEN_INT
12	0x0	**TUNING_SYS_INT_STATUS**: CMD19 was issued while EXECUTE_TUNING was set. System software can use this interrupt to implement a UHS-I tuning procedure for a standard SD Host driver. **0** = NO_INT **1** = GEN_INT
11:8	0x0	**TUNING_ASYNC_FIFO_ADDNL_DELAY**: This register field holds the additional delay in cycles which should be added to round trip delay Default value is - zero. SW should not update this field unless a new PROD setting is given.
6	0x0	**SD_BUS_POWER_ON_OFF_INT_ENABLE**: Enables a separate system interrupt (i.e., not the standard SDHCI interrupt) when SD_BUS_POWER is changed. **0** = DISABLE **1** = ENABLE
5	0x0	**VOLT_SWITCH_INT_ENABLE**: Enables a separate system interrupt (i.e., not the standard SDHCI interrupt) when VOLT_18_EN is changed. **0** = DISABLE **1** = ENABLE
4	0x0	**TUNING_SYS_INT_ENABLE**: Enables a separate system interrupt (i.e., not the standard SDHCI interrupt) when CMD19 is issued while EXECUTE_TUNING is set. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
3	0x0	**ASSERT_BUFF_RD_RDY_INT**: Write a 1 to this field to assert sdmmc_interrupt_status_0_buffer_read_ready. Used by the system software that implements the tuning procedure to signal to the standard SD driver that the tuning process has completed **0** = DISABLE **1** = ENABLE
2	0x0	**IO_TRIM_BYPASS**: Override bit for selecting between core trimmer (Vcore dependent) and io trimmer (custom trimmer) in IB clock path Default option is IO trimmer; SW should not set this field.
1	0x1	**INT_MASK_WHILE_TUNING**: As per spec, Host should not generate any interrupts (including cmd_complete and data_xfer_complete) except buffer_read_ready interrupt during tuning sequence is being performed SW can override this behavior by clearing this bit - but this leads to a spec violation **0** = DISABLE **1** = ENABLE
0	0x0	**SPI_MODE**: This mode is not supported.

SDMMC_VENDOR_ERR_INTR_STATUS_0

Legacy Interrupt Status Register

The fields are valid when a error interrupt has occurred.

Offset: 0x108
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000X0000 (0bxxxx,xxxx,xxxx,xx00,0000,xxx0,0000,0000)

Bit	Reset	Description
18	X	**SDMMC_LEGACY_CTLR_IDLE**: indicates legacy SD interface controller is idle - no active data transfers on legacy SD interface
17	0x0	**READ_DATA_TIMEOUT**: valid when a data timeout error occurs
16	0x0	**WRITE_CRC_STATUS_TIMEOUT**: valid when a data timeout error occurs

Bit	Reset	Description
15	0x0	**WRITE_BUSY_TIMEOUT**: valid when a data timeout error occurs
14	0x0	**RESP_BUSY_TIMEOUT**: valid when a data timeout error occurs
13	0x0	**SPI_WRITE_BUSY_TIMEOUT**
12	0x0	**SPI_RX_START_TOKEN_TIMEOUT**
8:5	0x0	**SPI_DAT_ERR_TOKEN**: Data Error Token,while read from card.
4:0	0x0	**SPI_DAT_RESPONSE**: Data Response while write to card **5** = DATA_ACCEPTED **11** = CRC_ERR **13** = WRITE_ERR

SDMMC_VENDOR_CAP_OVERRIDES_0

Capabilities override bits

Offset: 0x10c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00001107 (0bxxxx,xxxx,xxxx,xxxx,xx01,0001,xxxx,0111)

Bit	Reset	Description
13:8	0x11	**DQS_TRIM_VAL**: Tap value for incoming DQS path trimmer - used in HS400 modes
3	0x0	**DRV_LPBK_CLK_ON_CMD_LINE**: Loopback trimmed clock will be driven onto cmd line, if this bit set to 1. Should be set to zero during normal data transfers. Useful in debug.
2	0x1	**VOLTAGE_3_3_V_SUPPORT_OVERRIDE**: Voltage support 3_3_V override
1	0x1	**VOLTAGE_3_0_V_SUPPORT_OVERRIDE**: Voltage support 3_0_V override
0	0x1	**VOLTAGE_1_8_V_SUPPORT_OVERRIDE**: Voltage support 1_8_V override

SDMMC_VENDOR_DEBOUNCE_COUNT_0

Debounce Counter Value Register

The Debounce Counter runs on 32KHz clock. Keeping the default value to 100ms = (100 * 32cycles/1ms) = 3200 cycles for 100ms = 0xC80

Offset: 0x11c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000c80 (0bxxxx,xxxx,0000,0000,0000,1100,1000,0000)

Bit	Reset	Description
23:0	0xc80	**VALUE**: The number of 32KHz clock cycles is programed to meet Debounce period of the card slot. This register is valid for only SDMMC1.

SDMMC_VENDOR_MISC_CNTRL_0

Misc Vendor Cntrl Register

SDMMC_SPARE0: Spare register bits with reset value of 0

- SDMMC_SPARE0[0] : SW_RESET_CLKEN_OVERRIDE, override the sdmmc_clken when doing SW_RESET if set to 1.
- SDMMC_SPARE0[1] : When set, allows SD clock to be stopped in the middle of a read data block while in SDR104/HS400 modes(allow_sdr104_intrablock_stalls).
- Unsafe for some SD/eMMC cards, but may improve SDR104 DMA read performance in some cases.
- SDMMC_SPARE0[2] : When set, SDR104 support is advertised in SDMMC_CAPABILITIES_HIGHER_0_SDR104
- SDMMC_SPARE0[3] : When set, SDR50 support is advertised in SDMMC_CAPABILITIES_HIGHER_0_SDR50
- SDMMC_SPARE0[5] : When 0, masks the pad macro's "high speed" enable to 0, causing the pad macro to always launch
- data on the falling edge of the clock. This prevents the SD Host driver's setting of
- SDMMC_POWER_CONTROL_HOST_x_HIGH_SPEED_EN from undesirably affecting the output timing.
- SDMMC_SPARE0[7:6] : Number of pipe stages.
- SDMMC_SPARE0[8] : When set, DDR50 support is advertised in SDMMC_CAPABILITIES_HIGHER_0_DDR50

SDMMC_SPARE1: Spare register bits with reset value of 1

- SDMMC_SPARE1[0] : Reserved

- SDMMC_SPARE1[1] : Reserved

Offset: 0x120
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffff0098 (0b1111,1111,1111,1111,0000,0000,1001,1000)

Bit	Reset	Description
31:16	0xffff	**SDMMC_SPARE1**: Spare register bits with reset value of 1
15:1	0x4c	**SDMMC_SPARE0**: Spare register bits with reset value of 0x4C
0	0x0	**ERASE_TIMEOUT_LIMIT**: Erase timeout value. **0** = FINITE :Finite, It is limited to the programmed value in the DATA_TIMEOUT_VALUE **1** = INFINITE :Infinite, Controller would be monitoring until the card is busy.

SDMMC_VENDOR_MISC_CNTRL1_0

Offset: 0x124
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:16	0x0	**OVERRIDE_FOR_1_8V**: Maximum override for 1.8V VDD1
15:8	0x0	**OVERRIDE_FOR_3_0V**: Maximum override for 3.0V VDD1
7:0	0x0	**OVERRIDE_FOR_3_3V**: Maximum override for 3.3V VDD1

SDMMC_VENDOR_MISC_CNTRL2_0

Offset: 0x128
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

Reset: 0x62600000 (0b0110,001x,x11x,0000,0000,0000,0000,0000)
PROD: 0x00000000 (0bx0xx,xx0x,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	PROD	Description
31	0x0	_NONE_	**VGPIO_MODE_EN**: SW should set this to 1 when CD and WP pins are connected to external PMIC/VGPIO and not connected to SoC pins
30	0x1	0x0	**SDMMC_CLK_OVR_ON**: Master clk en Override bit for all SLCGs
29	0x1	_NONE_	**SD_CARD_DETECT_STATUS_N**: SW should read CD_N status from external PMIC/VGPIO controller and update this field to get SDMMC present state register gets updated. Present state register is read by standard SDHC driver to know card status. 0 - card detected 1 - no card present in slot
28	0x0	_NONE_	**SD_CARD_WP_STATUS**: SW should read Write Protect status from external PMIC/VGPIO controller and update this field to get SDMMC present state register gets updated. Present state register is read by standard SDHC driver to know card status. 0 - card is not write protected 1 - card is write protected
27	0x0	_NONE_	**CMD_TFIFO_HOT_RESET**: SW can reset CMD tuning FIFO present in padmacro incase there is any error or hang condition. Reset duration should be at least 20 cycles. Set this bit to 1 and clear it after 20 cycles to reset FIFO.
26	0x0	_NONE_	**DAT_TFIFO_HOT_RESET**: SW can reset DAT tuning FIFO present in padmacro incase there is any error or hang condition. Reset duration should be at least 20 cycles. Set this bit to 1 and clear it after 20 cycles to reset FIFO.
25	0x1	NORMAL	**ADMA3_CLKEN_OVERRIDE**: Override for sdmmc_adma3_g_clk clken; **0** = NORMAL :0 -> sdmmc_adma3_g_clk is gated in nonADMA3 modes **1** = OVERRIDE :1 -> sdmmc_adma3_g_clk is not gated in nonADMA3 modes
22	0x1	_NONE_	**ADMA3_DESC_PREFETCH_EN**: When set to 1, enables ADMA3 descriptors pre-fetch feature. It helps improving perf when system memory is heavily fragmented (number of descriptors programmed per transfer will be more)
21	0x1	_NONE_	**ADMA2_DESC_PREFETCH_EN**: When set to 1, enables ADMA2 descriptors pre-fetch feature. It helps improving perf when system memory is heavily fragmented (number of descriptors programmed per transfer will be more)

Bit	Reset	PROD	Description
19:16	0x0	_NONE_	**DATA_TIMEOUT_VAL_MULTIPLIER**: Used when SDMMC IO clock is used instead TMCLK for running data timeout counter (USE_TMCLK_FOR_DATA_TIMEOUT is set in VENDOR_SYS_SW_CNTRL register). Effective data timeout val = (multiplier+1) * data_timeout_val 0 - no multiplier
15:12	0x0	_NONE_	**DAT_TUNING_ASYNC_FIFO_ADDNL_DELAY**: This register field holds the additional delay in cycles which should be added to wdata/crc token round trip delay Default value is - zero. NOTE: SW should not update this field unless a new PROD setting is given.
11:8	0x0	_NONE_	**ADDITIONAL_NCR_CYCLES**: Additional Ncr wait time - useful for HW debug Default is 0 - SW should not modify this.
7:0	0x0	_NONE_	**OVERRIDE_FOR_1_8V_VDD2**: Maximum override for 1.8V VDD2

SDMMC_VENDOR_IO_TRIM_CNTRL_0

Vendor IO trimmer control register

Used to configure IO trimmer
Truth table
Input Pins Output Comments
E_DPD SEL_VREG SEL_VREF CLKOUT
1 x x 0 The cell is in deep power down mode
0 1 x based on ip_clk_select selected clock input Trimmer is powered by VAUXC
0 0 x based on ip_clk_select selected clock input Trimmer is powered by regulated voltage
* 'x' indicates don't care

Offset: 0x1ac
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000205 (0b0000,xxxx,xxxx,xxxx,xxxx,x010,000x,x101)

Bit	Reset	Description
31:28	0x0	**TRIM_PAD_RFU_IN**: Unused comp pad input pins. Reserved

Bit	Reset	Description
10:8	VREF_750_MV	**SEL_VREF_LEVEL**: Selects Vref voltage level **0** = VREF_700_MV **1** = VREF_725_MV **2** = VREF_750_MV **3** = VREF_775_MV **4** = VREF_800_MV **5** = VREF_850_MV **6** = VREF_900_MV **7** = VREF_950_MV
7:6	0x0	**TRIM_SEL_ATEST**: For testing purpose only - should be used when sdmmc is in idle state Select analog test signals to send to comp pad. 0x0: Not used (float) 0x1: Regulator input voltage 0x2: Regulator output voltage before analog mux 0x3: Regulator output after analog mux
5	ENABLE	**TRIM_PWRSAVE**: Enables power saving mode by clock gating the unused taps in delay chain Active low signal, 0 - power saving mode enabled - clock gating is enabled for unused trimmer taps - may affect tap delay 1 - no power saving - all the trimmer taps are not clock gated **0** = ENABLE **1** = DISABLE
2	0x1	**SEL_VREG**: By default, BG is disabled to save power if interface is not used. SW should select BG for error free SD/eMMC operation. For BG <-> VAUXC switching, SW should follow the switching sequence given in TRM/IAS. Select voltage supply for delay chain present in both Trimmer and DLLs PROD value: 0x0 ***SW should set this to 0x0 before accessing SD/eMMC. This setting makes IB trimmer delay independent of VDD_CORE*** 0 - selects regulated reference voltage for trimmer supply - default (recommended option for tunable SD/eMMC modes) 1 - selects VAUXC for trimmer supply and shut down BG+REG circuit (can be used in non-Tunable modes for power saving) Power up time for BG+REG is ~3us(worst case). Power down time for BG+REG is ~1us(worst case). If SW wants to turn on/off BG+REG when SDMMC is idle, it has to take hit of 3us power on time. When SEL_VREG is toggled, both DLL and rx clock trimmer output could glitch irrespective of input clock state which could cause corresponding rx CMD and DATA FIFOs to go into bad state. SW should issue SW_RESET_DAT and SW_RESET_CMD to reset host FIFOs after BG <-> VAUXC switching. This would ensure error free data transfers from there on. Powering down BG would need 3usec turn ON time which may cause IOPS reduction, if SW shut downs BG after every transfer and enables it on seeing new xfer req. Hence, it may not be possible to do dynamic shut down of BG without stalling new requests.

Bit	Reset	Description
1	0x0	**SEL_VREF**: Select reference voltage for voltage regulator 0 - selects Bandgap Voltage Reference (recommended option for SD/eMMC tunable modes) 1 - selects resistor divider voltage reference and power down bandgap Switching time between the supplies is 1us. When switching from one supply to other supply, we need to wait for at least 1us before doing any data transfers. Providing reference voltage from R divider network is just a backup plan, if A. Bandgap does not work or B. Bandgap works very well but we want to save bandgap power when Silicon Characterization results shows that the eMMC/ SDMMC interface perform well even by using R divider+REG+TRIMMER
0	0x1	**PD_BGREG**: Not used - Dummy control Power down Band Gap voltage reference, voltage regulator and resistor chain voltage ref (BG+REG) present in custom IO trimmer used for SD/eMMC bus tuning. Active High signal, PD_BGREG=1 => Power down BG+REG; PD_BGREG=0 => power up Power down and up time for BG+REG is 1us. If SW wants to turn on/off BG+REG when SDMMC is idle, it has to follow 1us power on/off time. Back-up option for powering down BG. SW should clear this bit when it wants to turn ON BG by setting SEL_VREG=0. The original idea to have PD_BGREG pin is to provide power saving feature when the eMMC/SDMMC is in IDLE state, but not in DPD mode. This pin function is actually merged into SEL_VREG function -BG+REG circuit is shut-down when SEL_VREG=1 (trimmer powered by VAUXC)

SDMMC_VENDOR_TUNING_CNTRL0_0

Vendor Tuning Control0 register

Offset: 0x1c0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x74020090 (0bx111,0100,0000,001x,0000,0000,1001,0000)
PROD: 0x00000040 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,01xx,xxxx)

Bit	Reset	PROD	Description
30	0x1	_NONE_	**RD_DATA_CRC_ERR_EN**: If cleared, Re-tuning request/tuning error is not generated on read data crc error
29	0x1	_NONE_	**WR_DATA_CRC_ERR_EN**: If cleared, Re-tuning request/tuning error is not generated on write crc error

Bit	Reset	PROD	Description
28	0x1	_NONE_	**CMD_CRC_ERR_EN**: If cleared, Re-tuning request/tuning error is not generated on cmd crc error
27	0x0	_NONE_	**RETUNING_REQ_EN_ON_CRC_ERR_DETECTION**: Re-tuning request is generated when set to initiate re-tuning by SW to compensate for temperature drift when any data or cmd CRC errors are detected in SDR50/SDR104/HS200 modes
26	0x1	_NONE_	**TUNING_ERR_EN_ON_CRC_ERR_DETECTION**: Tuning error is generated to initiate re-tuning by SW to compensate for temperature drift when any data or cmd CRC errors are detected in SDR50/SDR104/HS200 modes
25:18	0x0	_NONE_	**START_TAP_VAL**: start tap value to be used by tuning; start_tap should be multiple of step_size chosen and its valid range is 0-255; Not valid when TAP_VAL_UPDATED_BY_HW is set to zero. Tuning algorithm uses this as the start tap value for scanning through trimmer taps.
17	0x1	_NONE_	**TAP_VAL_UPDATED_BY_HW**: This bit is functional only in tunable modes (SDR50, SDR104 and HS200) SW can choose to update the tap val by itself by clearing this bit; Preferred value is 1 - tap val is updated by HW. If this bit is cleared, HW does not update tap_val per every tuning iteration. SW can update it as desired. Tuning pattern match is indicated by sampling_clock_select per every tuning iteration. SW has to maintain the status of each tuning iteration and determine the best PASS window to fix the final sampling point. And SW can program NUM_TUNING_ITERATIONS as desired. Once the number of tuning commands issued reaches number of tuning iterations programmed, execute_tuning bit will be cleared to indicate the completion of tuning procedure. Please note that using this option violates Host Spec but provided for legacy reasons. It helps us in using legacy SW tuning solution incase HW solution does not work.
15:13	TRIES_40	_NONE_	**NUM_TUNING_ITERATIONS**: The number of tuning iterations to be used by tuning circuit. **0** = TRIES_40 **1** = TRIES_64 **2** = TRIES_128 **3** = TRIES_192 **4** = TRIES_256

Bit	Reset	PROD	Description
12:6	0x2	0x1	**MUL_M**: implements a multiplier - M+1 Final tap value is derived from best passing window and calculated as follows. Final tap value = first_pass + ((last_pass - first_pass)*Q); where Q = percentage of pass window;default-75% Q = M+1/(2^N); N:1...7 M:should be in range [0:2^N-1];
5:3	0x2	_NONE_	**DIV_N**: implements a divider - 2^N; max div is 2^7 =>128
2:0	0x0	_NONE_	**TUNING_WORD_SEL**: Selects desired word from 256-bit tuning status bitmap status_word[31:0] = status[255:0] >> (tuning_word_sel * 32)

SDMMC_VENDOR_TUNING_CNTRL1_0

Vendor Tuning Control1 register
Different step size is required in SDR50 mode to cover two UI (100MHz => 2*10ns)
With 70ps/tap trimmer resolution, we can cover almost 2UI using step_size=8 in SDR50 and step_size=4 in SDR104.
Tuning will be done in HS200 - SDR mode only.
HS200 - tuning @200MHz - UHS_SEL should be SDR104 for executing tuning
Before initiating data transfers in HS400 mode, tuning procedure should be executed in HS200 mode with IO clock running @200MHz
DQ_OFFSET: offset between even and odd bits. Td is per-tap delay in trimmer. When the DQ offset function is turned off (DQ_OFFSET[1:0]=00), there is no offset between ZI0~7. When the DQ offset function is turned on
(DQ_OFFSET[1:0]=01, 10, 11),
extra delays are added at the odd bits (DQ1, 3, 5,7). Thus, there is an offset between even bits and odd bits. DQ offset function is turned on during auto-tuning to avoid the window-merged issue and turned off during normal operation.
00 no offset
01 1*Td
10 2*Td
11 3*Td

Offset: 0x1c4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000400 (0b00xx,xxxx,xxxx,xx00,0000,0100,x000,x000)

Bit	Reset	Description
31:30	0x0	**DQ_OFFSET**: offset between even and odd bits
17	0x0	**FIRST_PASS_WINDOW_SEL**: This enables the selection of the first pass window by the HW tuning engine. First pass window select feature is enabled only when STEP_SIZE is set to 0.
16	0x0	**FALSE_PASS_MASK**: This enables masking of the false pass windows from the tap value selection. False pass mask feature is enabled only when STEP_SIZE is set to 0.
15:8	0x4	**MIN_PASS_WINDOW_WIDTH**: This is used to mask false passes. Tuning engine considers PASS windows of size > MIN_WIDTH for tap value calculation. Allowed range is 1<= MIN_PASS_WINDOW_WIDTH <=4.
6:4	0x0	**STEP_SIZE_SDR104_HS200**: tap_val is incremented by step_size for every tuning iteration - used in SDR104/HS200/HS400 mode increment = 2^step_size; step_size should be in range 0-4. Others are RSVD
2:0	0x0	**STEP_SIZE_SDR50**: tap_val is incremented by step_size for every tuning iteration - used in SDR50 mode increment = 2^step_size; step_size should be in range 0-4. Others are RSVD

SDMMC_VENDOR_TUNING_STATUS0_0

Vendor Tuning Status0 register

Offset: 0x1c8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**STATUS_WORD**: Each bit indicates the status of each tuning iteration, when tap value is updated by HW (TAP_VAL_UPDATED_BY_HW=1); 0-Tuning pattern not matched 1-tuning pattern matched We have a total of 256 tap values. SW can issue a max. of 256 tuning commands for debug. SW needs to read this register eight times to get status of all 256 iterations by changing tuning_word_sel status[255:0] is left shifted and loaded into this register every time when tuning_word_sel is changed status_word[31:0] = status[255:0] >> (tuning_word_sel * 32)

SDMMC_VENDOR_TUNING_STATUS1_0

Vendor Tuning Status1 register

Offset: 0x1cc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**PASS_WINDOW_END_BEFORE_FINE_TUNING**: End tap value of best PASS window found by scan FSM
23:16	0x0	**PASS_WINDOW_START_BEFORE_FINE_TUNING**: Start tap value of best PASS window found by scan FSM
15:8	0x0	**PASS_WINDOW_END_AFTER_FINE_TUNING**: End tap value of best PASS window after fine tuning
7:0	0x0	**PASS_WINDOW_START_AFTER_FINE_TUNING**: Start tap value of best PASS window after fine tuning

SDMMC_VENDOR_CLK_GATE_HYSTERESIS_COUNT_0

Vendor Clk gating Hysteresis Counter initial value

Offset: 0x1d0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,1111)

Bit	Reset	Description
5:0	0xf	**CLK_COUNT**: Before gating second level clocks controller will wait for these many cycles. we can recover if any idle windows are missed in clken equation

SDMMC_VENDOR_PRESET_VAL0_0

Vendor Preset Value Registers

SD host spec defines one preset value register for each bus speed mode which should be set by host by some unique method.
Preset values vary based on the base frequency used which is in SW (SoC system driver) control. System driver supposed to set BASE_CLK_FREQ in VENDOR_CLOCK_CNTRL register before handling over the control to SD host standard driver.
In the similar way, system driver should set below vendor preset values based on the base clock frequency and the desired card clock frequency in each
bus speed mode
This should be done after every time SDMMC is reset and after every soft reset.
This is important as all SDMMC controllers follow the same register map, but could be programmed with different frequencies depending on the use case.
Default values are set assuming base clock frequency=208 MHz.

Offset: 0x1d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00201000 (0bxx00,0000,0010,0000,0001,0000,0000,0000)

Bit	Reset	Description
29:20	0x2	**SDCLK_FREQ_SEL_HIGH_SPEED**: System software programs 10-bit divider value to generate SD clk in default speed mode (<50MHz,3.3Vsignaling) This value is readable in the standard via PRESET_SDR12_AND_HIGH_0_SDCLK_FREQ_VAL_LOW register field Default val is 0x2 assuming 208MHz base clock
19:10	0x4	**SDCLK_FREQ_SEL_DEFAULT**: System software programs 10-bit divider value to generate SD clk in default speed mode (<25MHz,3.3Vsignaling) This value is readable in the standard via PRESET_DEFAULT_AND_INIT_0_SDCLK_FREQ_VAL_HIGH register field Default val is 0x4 assuming 208MHz base clock
9:0	0x0	**SDCLK_FREQ_SEL_INIT**: System software programs 10-bit divider value to generate desired SD clk frequency during initialization This value is readable in the standard via PRESET_DEFAULT_AND_INIT_0_SDCLK_FREQ_VAL_LOW register field For Eg., if 400KHz SDCLK is desired @base clk freq=48MHz, this register should be programmed with 0x3C

SDMMC_VENDOR_PRESET_VAL1_0

Offset: 0x1d8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00100804 (0bxx00,0000,0001,0000,0000,1000,0000,0100)

Bit	Reset	Description
29:20	0x1	**SDCLK_FREQ_SEL_SDR50**: System software programs 10-bit divider value to generate SD clk in SDR50 mode (<100MHz,1.8Vsignaling) This value is readable in the standard via PRESET_SDR50_AND_SDR25_0_SDCLK_FREQ_VAL_HIGH register field Default val is 0x1 (gives 2N divider) assuming 208MHz base clock
19:10	0x2	**SDCLK_FREQ_SEL_SDR25**: System software programs 10-bit divider value to generate SD clk in SDR25 mode (<50MHz,1.8Vsignaling) This value is readable in the standard via PRESET_SDR50_AND_SDR25_0_SDCLK_FREQ_VAL_LOW register field Default val is 0x2 assuming 208MHz base clock
9:0	0x4	**SDCLK_FREQ_SEL_SDR12**: System software programs 10-bit divider value to generate SD clk in SDR12 mode (<25MHz,1.8Vsignaling) This value is readable in the standard via PRESET_SDR12_AND_HIGH_0_SDCLK_FREQ_VAL_HIGH register field

SDMMC_VENDOR_PRESET_VAL2_0

Offset: 0x1dc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000800 (0bxxxx,xxxx,xxxx,0000,0000,1000,0000,0000)

Bit	Reset	Description
19:10	0x2	**SDCLK_FREQ_SEL_DDR50**: System software programs 10-bit divider value to generate SD clk in DDR50 mode (<50MHz,1.8Vsignaling) This value is readable in the standard via PRESET_DDR50_AND_SDR104_0_SDCLK_FREQ_VAL_HIGH register field Default val is 0x2 assuming 208MHz base clock
9:0	0x0	**SDCLK_FREQ_SEL_SDR104**: System software programs 10-bit divider value to generate SD clk in SDR104 mode (<208MHz,1.8Vsignaling) This value is readable in the standard via PRESET_DDR50_AND_SDR104_0_SDCLK_FREQ_VAL_LOW register field

SDMMC_SDMEMCOMPPADCTRL_0

SDMEMCOMP Pad control register

This register is used to control COMP pad inputs.

Offset: 0x1e0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x08000000 (0b00x0,1xx0,0000,xxx0,0000,0000,0000,0000)
PROD: 0x00007000 (0bxxxx,xxxx,xxxx,xxx0,0111,xxxx,xxxx,xxxx)

Bit	Reset	PROD	Description
31	0x0	_NONE_	**PAD_E_INPUT_OR_E_PWRD**: used to control E_INPUT(for SDMMC1/3) and E_PWRD (for SDMMC4) input of pu/pd comp pad should be set at least 1usec before starting auto-cal and cleared once auto-calibration is done (for power saving) NOTE: E_PWRD = !PAD_E_INPUT_OR_E_PWRD and E_INPUT = PAD_E_INPUT_OR_E_PWRD
30	0x0	_NONE_	**COMP_PAD_E_PBIAS_BUF**: Active high. Enables internally generated bias levels for driver PMOS. Reserved
28:27	0x1	_NONE_	**COMP_PAD_DRV_TYPE**: used to control drv_type input of BDSDMEMLVCOMP_C pad
24:20	0x0	_NONE_	**COMP_PAD_DRVUP_OVR**: used to drive DRVUP input of COMP pad if AUTO_CAL_ENABLE is disabled
16:12	0x0	0x7	**COMP_PAD_DRVDN_OVR**: used to drive DRVDN input of COMP pad if AUTO_CAL_ENABLE is disabled
11	0x0	_NONE_	**COMP_PAD_E_TEST_OUT**: used to control e_test_out input of COMP pad
10:7	0x0	_NONE_	**COMP_PAD_RFU_IN**: Unused comp pad input pins. Reserved
6:4	0x0	_NONE_	**COMP_PAD_TEST_SEL**: used to control test_sel input of COMP pad
3:0	0x0	_NONE_	**SDMMC2TMC_CFG_SDMEMCOMP_VREF_SEL**: Select different bias levels for driver PMOS when E_PBIAS_BUF=1. Reserved

SDMMC_AUTO_CAL_CONFIG_0

SDMEMCOMP pad auto-calibration settings
AUTO_CAL_SLW_OVERRIDE
0 (Normal operation) pad DRVDN/UP_SLWR/F tied to AUTO_CAL output
DRDVDN/UP_SLWR/F[1:0] = AUTO_CAL_PULLDOWN/UP[4:3]
1 (override) use CFG2TMC_SDIO[1|3]*_DRVDN/UP_SLWR/F pins to control pad slew inputs
AUTO_CAL_OVERRIDE

0 (normal operation): use AUTO_CAL_PU/PD_OFFSET as an offset to the calibration state machine setting
1 (override) : use AUTO_CAL_PU/PD_OFFSET register values directly

Offset: 0x1e4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00010000 (0b0000,xxxx,xxxx,x001,xxx0,0000,xxx0,0000)
PROD: 0x20000000 (0bxx1x,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	PROD	Description
31	0x0	_NONE_	**AUTO_CAL_START**: Writing a one to this bit starts the calibration state machine. This bit must be set even if the override is set in order to latch in the override value.
30	0x0	_NONE_	**AUTO_CAL_OVERRIDE**: AUTOCAL override. **0** = NORMAL :0 (normal operation): use AUTO_CAL_PU/PD_OFFSET as an offset to the calibration state machine setting **1** = OVERRIDE :1 (override) : use AUTO_CAL_PU/PD_OFFSET register values directly
29	DISABLED	ENABLED	**AUTO_CAL_ENABLE**: AUTOCAL enable. **0** = DISABLED :0 (disabled): use sdmmc2tmc_cfg* register settings for pullup/dn **1** = ENABLED :1 (normal operation): use SDMMC generated pullup/dn (override or AUTOCAL)
28	0x0	_NONE_	**AUTO_CAL_SLW_OVERRIDE**: AUTOCAL slew rate override **0** = NORMAL :0 (Normal operation) pad DRVDN/UP_SLWR/F tied to AUTO_CAL output DRDVDN/UP_SLWR/F[1:0] = AUTO_CAL_PULLDOWN/UP[4:3] **1** = OVERRIDE :1 (override) use CFG2TMC_SDIO[1|3]*_DRVDN/UP_SLWR/F pins to control pad slew inputs
18:16	0x1	_NONE_	**AUTO_CAL_STEP**: calibration step interval (in microseconds)
12:8	0x0	_NONE_	**AUTO_CAL_PD_OFFSET**: 2's complement offset for pull-down value
4:0	0x0	_NONE_	**AUTO_CAL_PU_OFFSET**: 2's complement offset for pull-up value

SDMMC_AUTO_CAL_INTERVAL_0

SDMEMCOMP pad calibration interval

Offset: 0x1e8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**AUTO_CAL_INTERVAL**: 0: do calibration once Otherwise, auto-calibration occurs at intervals equivalent to the programmed number of microseconds.

SDMMC_AUTO_CAL_STATUS_0

SDMEMCOMP pad calibration status

Offset: 0x1ec
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0xx0,0000,xxx0,0000,xxx0,0000,xxx0,0000)

Bit	Reset	Description
31	0x0	**AUTO_CAL_ACTIVE**: One when auto calibrate is active - valid only after auto calibrate sequence has completed (AUTO_CAL_ACTIVE == 0)
28:24	0x0	**AUTO_CAL_PULLDOWN_ADJ**: Pulldown code sent to pads
20:16	0x0	**AUTO_CAL_PULLUP_ADJ**: Pullup code sent to pads
12:8	0x0	**AUTO_CAL_PULLDOWN**: Pulldown code generated by auto-calibration
4:0	0x0	**AUTO_CAL_PULLUP**: Pullup code generated by auto-calibration

SDMMC_IO_SPARE_0

These SPARE_OUT bits go to pipe -> pad and then come back as SPARE_IN
SPARE_OUT[3] : IO_SPARE[19] - used as MUX select which selects between one cycle delay and two cycle delay versions of cmd_oen to mask wdata of IB

capture flop.
0x0 : selects zero cycle delayed version
0x1 : selects one cycle delayed version - recommended
SPARE_OUT[2] : IO_SPARE[18] - used as active low enable for gating both CMD_IN and DAT_IN aysnc FIFOs wdata when we are driving CMD/DAT lines.
0x0 : write 1 when OEN is active and Zi value when OEN is not active into FIFO (default)
0x1 : write Zi value into FIFO irrespective of OEN state.

Offset: 0x1f0
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0008XXXX (0b0000,0000,0000,1000,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31:16	RW	0x8	**SPARE_OUT**
15:0	RO	X	**SPARE_IN**

SDMMC_CIF2AXI_CTRL_0

SDMMC CIF2AXI control register DMA transaction (MC transaction) attributes

Offset: 0x1fc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:8	0x0	**MC_WRITE_REQ_STREAM_ID**: MC write transaction stream ID
7:0	0x0	**MC_READ_REQ_STREAM_ID**: MC read transaction stream ID

SDMMC_TZ_DMA_CTRL_0

SDMMC DMA requests security attribute control register for DMA transaction (MC transaction) security attributes.
This register is used to control write/read access to secure memory region.
This register can be accessed only by TZ. Non-secure writes to this register are dropped by controller and reads return all ones.
Controller will assert PSLVERR when this register is accessed by a non-secure master.

Usage:
wsb_ns = AWPROT[1] = ~{SDMMCxx_TZ_DMA_CTRL_MC_WRITE_REQ_TZ_ACCESS_EN};
rsb_ns = ARPROT[1] = ~{SDMMCxx_TZ_DMA_CTRL_MC_READ_REQ_TZ_ACCESS_EN};

Offset: 0x200
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**MC_WRITE_REQ_TZ_ACCESS_EN**: TZ must set this bit to 1 to enable write access to secure memory region. TZ must clear this bit to disable write access to secure memory region.
0	0x0	**MC_READ_REQ_TZ_ACCESS_EN**: TZ must set this bit to 1 to enable read access to secure memory region. TZ must clear this bit to disable read access to secure memory region.

SDMMC_VENDOR_MISC_CNTRL3_0

Offset: 0x204
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x4f01000f (0b0100,1111,xxxx,xxx1,0000,0000,xxx0,1111)

Bit	Reset	Description
31	0x0	**ALLOW_INTRABLOCK_CLK_STALLING_IN_SDR50**: This disables the intrablock clock stopping in SDR50 mode
30	0x1	**STOP_TRIM_IN_CLK_DURING_TUNING**: When set to 1, rx clk trimmer and rx fifos input clock is stopped when tap value is changed during tuning process. When set to 0, rx clk trimmer and rx fifos input clock is not stopped during tap value change. Trimmer output is also clamped to zero during tuning tap val change.
29:24	0xf	**TUNING_TRIMMER_RECOVERY_TIME**: Trimmer output may glitch for 2 or 3 cycles, when tap value is changed irrespective of its input clock state. Do not use trimmer output during this uncertainty window. If STOP_TRIM_IN_CLK_DURING_TUNING is set to 1, - trimmer input clock is stopped for TUNING_TRIMMER_RECOVERY_TIME cycles and - rx fifos input clock is stopped (trimmer output clock is clamped) for TUNING_TRIMMER_RECOVERY_TIME cycles. This is required not to propagate glitch into FIFOs and other downstream logic.

Bit	Reset	Description
16	0x1	**EXTEND_SYNC_INTR_MASK_DURING_ABORT_OR_STOP_CMD**: When an async abort is issued during read operation, dat_fsm will move to idle state as soon as CMD12/CMD52 END bit is sent by core. This would start sync_intr_period. But due to the intermediate delay stages present in pipemacro and padmacro, END bit reaches device after some cycles. So, device would not stop data transmission till it sees END bit of ABORT CMD. During this time, core controller receives the data already driven by the device. If DAT[1] line has any zeroes during this period, SDIO sync interrupt detector will generate a spurious card interrupt. This register bit is used to mask sync intr detection period to avoid spurious interrupts. SW should not write into this field unless it is published in TRM/IAS.
15:8	0x0	**E_DIFF_DQ**: diff/Vref rx selection for DAT[7:0] If set to 1, enables differential amplitude receiver for DAT lines in EMMC IOBRICK. If set to 0, enables vref receiver for DAT lines in EMMC IOBRICK. (default)
4	0x0	**E_DIFF_CMD**: diff/Vref rx selection for CMD If set to 1, enables differential amplitude receiver for CMD in EMMC IOBRICK. If set to 0, enables vref receiver for CMD in EMMC IOBRICK. (default)
3	0x1	**DAT_OE_POSTAMBLE_EN**: If set, DAT pads output driver will be disabled one cycle after END bit of DATA pkt sent. If cleared, DAT pads output driver will be disabled in the same cycle END bit is sent.
2	0x1	**DAT_OE_PREAMBLE_EN**: If set, DAT pads output driver will be enabled one cycle before START bit of DATA pkt is transmitted on DAT lines. If cleared, DAT pads output driver will be enabled in the same cycle START bit is sent.
1	0x1	**CMD_OE_POSTAMBLE_EN**: If set, CMD pad output driver will be disabled one cycle after END bit of CMD pkt sent. If cleared, CMD pad output driver will be disabled in the same cycle END bit is sent.
0	0x1	**CMD_OE_PREAMBLE_EN**: If set, CMD pad output driver will be enabled one cycle before START bit of CMD pkt is transmitted on DAT lines. If cleared, CMD pad output driver will be enabled in the same cycle START bit is sent.

SDMMC_VENDOR_MISC_CNTRL4_0

Offset: 0x20c
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0xffff0000 (0b1111,1111,1111,1111,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0xffff	**SDMMC_SPARE3**: Reserved
15:0	0x0	**SDMMC_SPARE2**: Reserved

SDMMC_SDMEMCOMPPADCTRL_MISC_CTL_0

SDMEMCOMP Pad misc control register

This register is used to control COMP pad inputs.

Offset: 0x214
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:4	0x0	**COMP_PAD_SPARE_VDD**: Comp pad spare inputs
3:0	0x0	**COMP_PAD_SPARE_VAUXC**: Comp pad spare inputs

9.5.4.1 SDMMCAB Registers

SDMMCAB_SYSTEM_ADDRESS_0

32-bit Block Count (SDMA System Address) Register

When Host Version 4 Enable is set to 0 in the Host Control 2 register, SDMA
uses this register as system address in only 32-bit addressing mode. Auto CMD23 cannot be used with SDMA.
When Host Version 4 Enable is set to 1, SDMA uses ADMA System Address
register (05Fh-058h) instead of using this register to support both 32-bit and 64bit addressing.
This register is re-assigned to 32-bit Block Count and then SDMA may use Auto CMD23.

(1) SDMA System Address (Host Version 4 Enable = 0)
This register contains the system memory address for a SDMA transfer in 32bit addressing mode.

When the Host Controller stops a SDMA transfer, this register shall point to the system address of the next
contiguous data position. It can be accessed only if no transaction is executing (i.e., after a transaction has stopped).
Reading this register during SDMA transfers may return an invalid value. The Host Driver shall initialize this register before starting a SDMA
transaction.
After SDMA has stopped, the next system address of the next contiguous data position can be read from this register. The SDMA transfer waits at the
every boundary specified by the SDMA Buffer Boundary in the Block Size register.
The Host Controller generates DMA Interrupt to request the Host Driver to update this register. The Host Driver sets the next system address
of the next data position to this register. When the most upper byte of this register (003h) is written, the Host Controller
restarts the SDMA transfer. When restarting SDMA by setting Continue Request in the Block Gap Control register, the Host Controller
shall start at the next contiguous address stored here in the SDMA System Address register.
ADMA does not use this register.

(2) 32-bit Block Count (Host Version 4 Enable = 1)
Host Controller Version 4.10 re-defines this register as 32-bit Block Count
(Refer to Section 1.15 in SD Host spec4.1 for more details about block count extension). In version 4.00, this register may be used
as 32-bit block count only for Auto CMD23 to set the argument of the CMD23 while executing Auto CMD23.

FFFF_FFFFh 4G - 1 block
… …
0000_0002h 2 blocks
0000_0001h 1 block
0000_0000h Stop Count

The Host Controller would decrement the block count of this register every
block transfer and data transfer stops when the count reaches zero.
This register should be accessed only when no transaction is executing.
Reading this register during data transfers may return invalid value.

Offset: 0x0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ADDRESS**

SDMMCAB_BLOCK_SIZE_BLOCK_COUNT_0

Block Size Register

HOST_DMA_BUFFER_SIZE
The large contiguous memory space may not be available in the virtual memory system. To perform long SDMA transfer, SDMA System Address register shall be updated at every system memory boundary during SDMA transfer.
These bits specify the size of contiguous buffer in the system memory. The SDMA transfer shall wait at the every boundary specified by these fields and the Host Controller generates the DMA Interrupt to request the Host Driver to update the SDMA System Address register. At the end of transfer, the Host Controller may issue or may not issue DMA Interrupt. In particular, DMA Interrupt shall not be issued after Transfer Complete Interrupt is issued.
In case of this register is set to 0 (buffer size = 4K bytes), lower 12-bit of byte address points data in the contiguous buffer and the upper 20-bit points the location of the buffer in the system memory. The SDMA transfer stops when the Host Controller detects carry out of the address from bit 11 to 12.
These bits shall be supported when the SDMA Support in the Capabilities register is set to 1 and this function is active when the DMA Enable in the Transfer Mode register is set to 1.
ADMA does not use this register.
000b 4K bytes (Detects A11 carry out)
001b 8K bytes (Detects A12 carry out)
010b 16K Bytes (Detects A13 carry out)
011b 32K Bytes (Detects A14 carry out)
100b 64K bytes (Detects A15 carry out)
101b 128K Bytes (Detects A16 carry out)
110b 256K Bytes (Detects A17 carry out)
111b 512K Bytes (Detects A18 carry out)
XFER_BLOCK_SIZE_11_0
This register specifies the block size of data transfers for CMD17, CMD18, CMD24, CMD25, and CMD53. Values ranging from 1 up to the maximum buffer size can be set. In case of memory, it shall be set up to 512 bytes (Refer to Implementation Note in Section 1.7.2). It can be accessed only if no transaction is executing (i.e., after a transaction has stopped). Read operations during transfers may return an invalid value, and write operations shall be ignored.
0800h 2048 Bytes
.....
0200h 512 Bytes
01FFh 511 Bytes

.....
0004h 4 Bytes
0003h 3 Bytes
0002h 2 Bytes
0001h 1 Byte
0000h No data transfer
16-bit BLOCKS_COUNT
Host Controller Version 4.10 extends block count to 32-bit (Refer to Section 1.15 in SD host spec4.1).
Selection of either 16-bit Block Count register or 32-bit Block Count register is defined as follows:
(1) If Host Version 4 Enable in the Host Control 2 register is set to 0 or 16-bit Block Count register is set to non-zero, 16-bit Block Count register is selected.
(2) If Host Version 4 Enable is set to 1 and 16-bit Block Count register is set to zero, 32-bit Block Count register is selected.
Use of 16-bit/32-bit Block Count register is enabled when Block Count Enable in the Transfer Mode register is set to 1 and is valid only for multiple block transfers. The Host Driver shall set this
register to a value between 1 and the maximum block count. The Host Controller decrements the block count after each block transfer and stops when the count reaches zero. Setting the block count to 0 results in no data blocks is transferred. This register should be accessed only when no transaction is executing (i.e., after transactions are stopped). During data transfer, read operations on this register may return an invalid value and write operations are ignored.
When a suspend command is completed, the number of blocks yet to be transferred can be determined by reading this register. Before issuing a resume command, the Host Driver shall restore the previously saved block count.
FFFFh 65535 blocks
.....
0002h 2 blocks
0001h 1 block
0000h Stop Count

Offset: 0x4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,x000,0000,0000,0000)
PROD: 0x00000200 (0bxxxx,xxxx,xxxx,xxxx,xxxx,0010,0000,0000)

Bit	Reset	PROD	Description
31:16	0x0	_NONE_	**BLOCKS_COUNT**

Bit	Reset	PROD	Description
14:12	0x0	_NONE_	**HOST_DMA_BUFFER_SIZE**: **0** = DMA4K **1** = DMA8K **2** = DMA16K **3** = DMA32K **4** = DMA64K **5** = DMA128K **6** = DMA256K **7** = DMA512K
11:0	0x0	0x200	**XFER_BLOCK_SIZE_11_0**

SDMMCAB_ARGUMENT_0

Argument 1 Register

COMMAND_ARGUMENT
The SD Command Argument is specified as bit39-8 of Command-Format in the Physical Layer Specification.

Offset: 0x8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**COMMAND_ARGUMENT**

SDMMCAB_CMD_XFER_MODE_0

Command and Transfer Mode Register

COMMAND_INDEX
These bits shall be set to the command number (CMD0-63, ACMD0-63) that is specified in bits 45-40 of the Command-Format in the Physical Layer Specification and SDIO Card Specification.
COMMAND_TYPE
There are three types of special commands: Suspend, Resume and Abort. These bits shall be set to 00b for all other commands.
(1) Suspend Command

If the Suspend command succeeds, the Host Controller shall assume the SD Bus has been released and that it is possible to issue the next command, which uses the DAT line. The Host Controller shall de-assert Read Wait for read transactions and stop checking busy for write transactions. The interrupt cycle shall start, in 4-bit mode. If the Suspend command fails, the Host Controller shall maintain its current state, and the Host Driver shall restart the transfer by setting Continue Request in the Block Gap Control register. (Refer to 3.12.1 Suspend Sequence)
(2) Resume Command
The Host Driver re-starts the data transfer by restoring the registers in the range of 000-00Dh. (Refer to Figure 1-4 in section 1.6 for the register map.) The Host Controller shall check for busy before starting write transfers.
(3) Abort Command
If this command is set when executing a read transfer, the Host Controller shall stop reads to the buffer. If this command is set when executing a write transfer, the Host Controller shall stop driving the DAT line. After issuing the Abort command, the Host Driver should issue a software reset. (Refer to 3.8 Abort Transaction)
11b Abort CMD12, CMD52 for writing "I/O Abort" in CCCR
10b Resume CMD52 for writing "Function Select" in CCCR
01b Suspend CMD52 for writing "Bus Suspend" in CCCR
00b Normal Other commands
DATA_PRESENT_SELECT
This bit is set to 1 to indicate that data is present and shall be transferred using the DAT line. It is set to 0 for the following:
(1) Commands using only CMD line (ex. CMD52).
(2) Commands with no data transfer but using busy signal on DAT[0] line (R1b or R5b ex. CMD38)
(3) Resume command
CMD_INDEX_CHECK_EN
If this bit is set to 1, the Host Controller shall check the Index field in the response to see if it has the same value as the command index. If it is not, it is reported as a Command Index Error. If this bit is set to 0, the Index field is not checked.
CMD_CRC_CHECK_EN
If this bit is set to 1, the Host Controller shall check the CRC field in the response. If an error is detected, it is reported as a Command CRC Error. If this bit is set to 0, the CRC field is not checked. The position of CRC field is determined according to the length of the response. (Refer to definition in D01-00 and Table 2-10 below.)
Sub Command Flag
This bit is added from Version 4.10 to distinguish a main command or sub command (Refer to Section 1.17). When issuing a main command, this bit is set to 0 and when issuing a sub command, this bit is set to 1. Setting of this bit is checked by Sub Command Status in the Present State register.
Host Driver manages whether main or sub command. Host Controller does

not refer to this bit to issue a command.
1 Sub Command
0 Main Command
RESP_TYPE_SELECT
Normal Mode:
00 No Response
01 Response Length 136
10 Response Length 48
11 Response Length 48 check Busy after response
Response Interrupt Disable
Host Controller Version 4.00 supports response error check function to avoid overhead of response error check by Host Driver. Only R1 or R5 can be checked. If Host Driver checks response error, sets this bit to 0 and waits Command Complete Interrupt and then check the response register.
If Host Controller checks response error, sets this bit to 1 and sets Response Error Check Enable to 1. Command Complete Interrupt is disabled by this bit regardless of Command Complete Signal Enable.
0 Response Interrupt is enabled
1 Response Interrupt is disabled
Response Error Check Enable
Host Controller Version 4.00 supports response error check function to avoid overhead of response error check by Host Driver. Only R1 or R5 can be checked. If Host Driver checks response error, this bit is set to 0 and Response Interrupt Disable is set to 0.
If Host Controller checks response error, sets this bit to 1 and sets Response Interrupt Disable to 1. Response Type R1 / R5 selects either R1 or R5 response type. If an error is detected, Response Error Interrupt is generated in the Response Error Interrupt Status register
0 Response Error Check is disabled
1 Response Error Check is enabled
Response Type R1 / R5
When response error check is enabled, this bit selects either R1 or R5 response types. Two types of response check is supported: R1 for memory and R5 for SDIO.
Error Statuses Checked in R1
Bit31 OUT_OF_RANGE
Bit30 ADDRESS_ERROR
Bit29 BLOCK_LEN_ERROR
Bit26 WP_VIOLATION
Bit25 CARD_IS_LOCKED
Bit23 COM_CRC_ERROR
Bit21 CARD_ECC_FAILED
Bit20 CC_ERROR
Bit19 ERROR
Response Flags Checked in R5
Bit07 COM_CRC_ERROR

Bit03 ERROR
Bit01 FUNCTION_NUMBER
Bit00 OUT_OF_RANGE
0 R1 (Memory)
1 R5 (SDIO)
MULTI_BLOCK_SELECT - Multi / Single Block Select
This bit is set when issuing multiple-block transfer commands using DAT line. For any other commands, this bit shall be set to 0. If this bit is 0, it is not necessary to set the Block Count register. (Refer to Table 2-8)
1 Multiple Block
0 Single Block
DATA_XFER_DIR_SEL - Data Transfer Direction Select
This bit defines the direction of DAT line data transfers. The bit is set to 1 by the Host Driver to transfer data from the SD card to the SD Host Controller and it is set to 0 for all other commands.
1 Read (Card to Host)
0 Write (Host to Card)
AUTO_CMD12_EN - Auto CMD Enable
This field determines use of auto command functions.
00b Auto Command Disabled
01b Auto CMD12 Enable
10b Auto CMD23 Enable
11b Auto CMD Auto Select
There are three methods to stop Multiple-block read/write operation by CMD23 or CMD12. In the other operations (ex. single read/write operation), this field is set to 00b.

(1) Auto CMD12 Enable
When this field is set to 01b, the Host Controller issues CMD12 automatically when last block transfer is completed. Auto CMD12 error is indicated to the Auto CMD Error Status register. The Host Driver shall not set this bit if the command does not require CMD12. In particular, secure commands
defined in the Part 3 File Security specification do not require CMD12.
When Host Version 4 Enable =0, CMD12 is issued when 16-bit Block Count is expired.
When Host Version 4 Enable =1, CMD12 is issued when 16-bit Block Count or 32-bit Block Count is expired.

(2) Auto CMD23 Enable
When this bit field is set to 10b, the Host Controller issues a CMD23 automatically before issuing a command specified in the Command Register. The Host Controller Version 3.00 and later shall support this function. The following conditions are required to use the Auto CMD23.
Auto CMD23 Supported (Host Controller Version is 3.00 or later)
A memory card that supports CMD23 (SCR[33]=1)
If DMA is used, it shall be ADMA.
Only when CMD18 or CMD25 is issued
(Note, the Host Controller doesn't check command index.)
Auto CMD23 can be used with or without ADMA. By writing the Command

register, the Host Controller issues a CMD23 first and then issues a command specified by the Command Index in Command register. If response errors of CMD23 are detected, the second command is not issued. A CMD23 error is indicated in the Auto CMD Error Status register.
32-bit block count value for CMD23 is set to 32-bit Block Count (SDMA System Address) register.

(3) Auto CMD Auto Select (Version 4.10)
As CMD23 is optional for SD Memory Card except UHS104 Card, If card supports CMD23, Auto CMD23 should be used instead of Auto CMD12. Host Controller Version 4.10 defines this "Auto CMD Auto Select" mode.
Selection of Auto CMD depends on setting of CMD23 Enable in the Host Control 2 register which indicates whether card supports CMD23. If CMD23 Enable =1, Auto CMD23 is used and if CMD23 Enable =0, Auto CMD12 is used. In case of Version 4.10 or later, use of Auto CMD Auto Select is recommended rather than use of Auto CMD12 Enable or Auto CMD23 Enable.
BLOCK_COUNT_EN - Block Count Enable
This bit is used to enable the Block Count register, which is only relevant for multiple block transfers. When this bit is 0, the Block Count register is disabled, which is useful in executing an infinite transfer. (Refer to Table 2-8)
Host Driver should set this bit to 0 when ADMA is used.
DMA_EN - DMA Enable
This bit enables DMA functionality as described in section 1.4. DMA can be enabled only if it is supported as indicated in the Capabilities register. One of the DMA modes can be selected by DMA Select in the Host Control register. If DMA is not supported, this bit is meaningless and shall always read 0. If this bit is set to 1, a DMA operation shall begin when the Host Driver writes to the upper byte of Command register (00Fh).

Offset: 0xc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
29:24	0x0	**COMMAND_INDEX**
23:22	0x0	**COMMAND_TYPE**: **0** = NORMAL **1** = SUSPEND **2** = RESUME **3** = ABORT
21	0x0	**DATA_PRESENT_SELECT**: **0** = NO_DATA_TRANSFER **1** = DATA_TRANSFER

Bit	Reset	Description
20	0x0	**CMD_INDEX_CHECK_EN**: **0** = DISABLE **1** = ENABLE
19	0x0	**CMD_CRC_CHECK_EN**: **0** = DISABLE **1** = ENABLE
18	0x0	**SUB_CMD_FLAG**: **0** = MAIN_CMD **1** = SUB_CMD
17:16	0x0	**RESP_TYPE_SELECT**: **0** = NO_RESPONSE **1** = RESP_LENGTH_136 **2** = RESP_LENGTH_48 **3** = RESP_LENGTH_48BUSY
8	0x0	**RESP_INT_DIS**: **0** = ENABLE **1** = DISABLE
7	0x0	**RESP_ERR_CHK_EN**: **0** = DISABLE **1** = ENABLE
6	0x0	**RESP_TYPE**: **0** = R1 **1** = R5
5	0x0	**MULTI_BLOCK_SELECT**: **0** = DISABLE **1** = ENABLE
4	0x0	**DATA_XFER_DIR_SEL**: **0** = WRITE **1** = READ
3:2	0x0	**AUTO_CMD12_EN**: **0** = DISABLE **1** = CMD12 **2** = CMD23 **3** = AUTO_CMD_AUTO_SEL
1	0x0	**BLOCK_COUNT_EN**: **0** = DISABLE **1** = ENABLE
0	0x0	**DMA_EN**: **0** = DISABLE **1** = ENABLE

SDMMCAB_RESPONSE_R0_R1_0

Command Response Registers
The Table below describes the mapping of command responses from the SD Bus to this register for each response type. In the table, R[] refers to a bit range within the

response data as transmitted on the SD Bus, REP[] refers to a bit range within the Response register.

Kind of Response	Meaning of Response	Response Field	Response Register
R1, R1b (normal response)	Card Status	R [39:8]	REP [31:0]
R1b (Auto CMD12 response)	Card Status for Auto CMD12	R [39:8]	REP [127:96]
R1 (Auto CMD23 response)	Card Status for Auto CMD23	R [39:8]	REP [127:96]
R2 (CID, CSD register)	CID or CSD reg. incl.	R [127:8]	REP [119:0]
R3 (OCR register)	OCR register for memory	R [39:8]	REP [31:0]
R4 (OCR register)	OCR register for I/O etc	R [39:8]	REP [31:0]
R5,R5b	SDIO response	R [39:8]	REP [31:0]
R6 (Published RCA response)	New published RCA[31:16] etc	R [39:8]	REP [31:0]

Response Bit Definition for Each Response Type
In UHS-II mode, the response of CM-TRAN abort CCMD (4-byte) is stored in offset 13h-10h and the response of SD-TRAN abort CCMD (8-byte) is stored in offset 1Fh-18h
Command Response [31:0] (R0) Register

Offset: 0x10
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CMD_RESP_31_16**
15:0	0x0	**CMD_RESP_15_0**

SDMMCAB_RESPONSE_R2_R3_0

Command Response [63:32] (R2) Register

Offset: 0x14
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CMD_RESP_63_48**
15:0	0x0	**CMD_RESP_47_32**

SDMMCAB_RESPONSE_R4_R5_0

Command Response [95:64] (R4) Register

Offset: 0x18
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CMD_RESP_95_80**
15:0	0x0	**CMD_RESP_79_64**

SDMMCAB_RESPONSE_R6_R7_0

Command Response [127:96] (R6) Register

Offset: 0x1c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CMD_RESP_127_112**
15:0	0x0	**CMD_RESP_111_96**

SDMMCAB_BUFFER_DATA_PORT_0

Buffer Data Port Register

The Host Controller Buffer can be accessed through this 32-bit Data Port Register.

Offset: 0x20
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**BUFFER_DATA**

SDMMCAB_PRESENT_STATE_0

Present State Register

CMD_LINE_LEVEL - CMD Line Signal Level
This status is used to check the CMD line level to recover from errors, and for debugging.
DAT_3_0_LINE_LEVEL - DAT[3:0] Line Signal Level
This status is used to check the DAT line level to recover from errors, and for debugging. This is especially useful in detecting the busy signal level from DAT[0].
Sub Command Status
The Command register and Response register are commonly used for main command and sub command. This status is used to distinguish which response error statuses, main command or sub command, indicated in the Error Interrupt Status register or in the UHS-II Error Interrupt Status register. Refer to Section 1.17 about details of response error statuses. Just before reading of this register, the Sub Command Flag of the Command register or the UHS-II Command register is copied to this status. This status is effective not only when Response Error interrupt is generated but also when data error interrupt is generated with Command Not Issued by Error (D27 of this register) or Auto CMD Error interrupt is generated with Command Not Issued by Error by Auto CMD12 in the Auto CMD Error Status register.

1 Sub Command Status
0 Main Command Status
Command Not Issued by Error
Setting of this status indicates that a command cannot be issued due to an error except Auto CMD12 error. (Equivalent error status by Auto CMD12 error is defined as Command Not Issued By Auto CMD12 Error in the Auto CMD Error Status register.) This status is set to 1 when Host Controller cannot issue a

command after setting Command register or
UHS-II Command register. Refer to Section 3.10 about 2L-HD error case in UHS-II mode.
Sub Command Status (D28) indicates which command is not issued (main or sub).

1 Command cannot be issued
0 No error for issuing a command

Host Regulator Voltage Stable
This status is added from Version 4.10 and is used to check whether host regulator voltage is stable for switching signal voltage of UHS-I mode.

1 Host Regulator Voltage is stable
0 Host Regulator Voltage is not stable

Support of this function is checked by reading this status after that Software Reset For All in the Software Reset register is cleared by the Host Controller in initialization. Setting this status to 1 means that this function is supported by the Host Controller.

This status may be related to 1.8V Signaling Enable in the Host Control 2 register. Changing 1.8V Signaling Enable causes unstable of host regulator voltage for I/O cell. Then once this status is set to 0 and retrieved to 1 when host regulator voltage is stable again. When executing power cycle, Host Driver also executes Software Reset For All and it clears 1.8V Signaling Enable to go back signal voltage to 3.3V.

If this status is not supported, Host Driver should take more than 5ms for stable time of host voltage regulator from changing 1.8V Signaling Enable. Specific Host Driver may use a specific time, which is provided by Host System, instead of using 5ms.
WRITE_PROTECT_LEVEL - Write Protect Switch Pin Level
CARD_DETECT_PIN_LEVEL - Card Detect Pin Level
CARD_STATE_STABLE - Card State Stable
CARD_INSERTED - Card Inserted
BUFFER_READ_EN - Buffer Read Enable
This status is used for non-DMA read transfers.
BUFFER_WRITE_EN - Buffer Write Enable
This status is used for non-DMA write transfers.
READ_XFER_ACTIVE - Read Transfer Active
WRITE_XFER_ACTIVE - Write Transfer Active
RETUNING_REQUEST - Re-Tuning Request
DAT_LINE_ACTIVE - DAT Line Active
CMD_INHIBIT_DAT - Command Inhibit (DAT)
CMD_INHIBIT_CMD - Command Inhibit (CMD)

Offset: 0x24
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxx0,0x00,0000,0000,xxxx,0000,0000,0000)
PROD: 0x000b0000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	PROD	Description
28	0x0	_NONE_	**SUB_CMD_STATUS**
27	0x0	_NONE_	**CMD_NOT_ISSUED_BY_ERROR**
25	0x0	_NONE_	**HOST_REG_VOLTAGE_STABLE**
24	0x0	_NONE_	**CMD_LINE_LEVEL**: **0** = LOW **1** = HIGH
23:20	0x0	_NONE_	**DAT_3_0_LINE_LEVEL**
19	0x0	ENABLED	**WRITE_PROTECT_LEVEL**: **0** = PROTECTED **1** = ENABLED
18	0x0	_NONE_	**CARD_DETECT_PIN_LEVEL**: **0** = NO_CARD **1** = CARD
17	0x0	INSERTED	**CARD_STATE_STABLE**: **0** = DEBOUNCE **1** = INSERTED
16	0x0	INSERTED	**CARD_INSERTED**: **0** = DEBOUNCE **1** = INSERTED
11	0x0	_NONE_	**BUFFER_READ_EN**: **0** = DISABLE **1** = ENABLE
10	0x0	_NONE_	**BUFFER_WRITE_EN**: **0** = DISABLE **1** = ENABLE
9	0x0	_NONE_	**READ_XFER_ACTIVE**: **0** = NO_DATA **1** = TRANSFERING
8	0x0	_NONE_	**WRITE_XFER_ACTIVE**: **0** = NO_DATA **1** = TRANSFERING
7:4	0x0	_NONE_	**DAT_7_4_LINE_LEVEL**
3	0x0	_NONE_	**RETUNING_REQUEST**: **0** = NOT_REQUIRED **1** = REQUIRED

Bit	Reset	PROD	Description
2	0x0	_NONE_	**DAT_LINE_ACTIVE**: **0** = INACTIVE **1** = ACTIVE
1	0x0	_NONE_	**CMD_INHIBIT_DAT**: **0** = INACTIVE **1** = ACTIVE
0	0x0	_NONE_	**CMD_INHIBIT_CMD**: **0** = INACTIVE **1** = ACTIVE

SDMMCAB_POWER_CONTROL_HOST_0

Power Control / Host Control Register

WAKEUP_ON_CARD_REMOVAL - Wakeup Event Enable On SD Card Removal
This bit enables wakeup event via Card Removal assertion in the Normal Interrupt Status register. FN_WUS (Wake Up Support) in CIS does not affect this bit.
WAKEUP_ON_CARD_INSERTION - Wakeup Event Enable On SD Card Insertion
This bit enables wakeup event via Card Insertion assertion in the Normal Interrupt Status register. FN_WUS (Wake Up Support) in CIS does not affect this bit.
WAKEUP_ON_CARD_INTERRUPT - Wakeup Event Enable On Card Interrupt
This bit enables wakeup event via Card Interrupt assertion in the Normal Interrupt Status register. This bit can be set to 1 if FN_WUS (Wake Up Support) in CIS is set to 1.
INTERRUPT_AT_BLOCK_GAP - Interrupt At Block Gap
This bit is valid only in 4-bit mode of the SDIO card and selects a sample point in the interrupt cycle. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer. Setting to 0 disables interrupt detection during a multiple block transfer. If the SD card cannot signal an interrupt during a multiple block transfer, this bit should be set to 0. When the Host Driver detects an SD card insertion, it shall set this bit according to the CCCR of the SDIO card.
READ_WAIT_CONTROL - Read Wait Control
The read wait function is optional for SDIO cards. If the card supports read wait, set this bit to enable use of the read wait protocol to stop read data using the DAT[2] line. Otherwise, the Host Controller has to stop the SD Clock to hold read data, which restricts commands generation. When the Host Driver detects an SD card insertion, it shall set this bit according to the CCCR of the SDIO card. If the card does not support read wait, this bit shall never be set to 1 otherwise DAT line conflict may occur. If this bit is set to 0, Suspend/Resume cannot be supported. In UHS-II mode, Read Wait is disabled and DAT[2] line is used for Interrupt Signal from UHS-II Card.
CONTINUE_REQUEST - Continue Request
This bit is used to restart a transaction, which was stopped using the Stop At

Block Gap Request. To cancel stop at the block gap, set Stop At Block Gap Request to 0 and set this bit 1 to restart the transfer.
The Host Controller automatically clears this bit when the transaction re-starts.
If Stop At Block Gap Request is set to 1, any write to this bit is ignored.
In SD mode, this bit is cleared in either of the following cases:
(1) In the case of a read transaction, the DAT Line Active changes from 0 to 1 as a read transaction restarts.
(2) In the case of a write transaction, the Write Transfer Active changes from 0 to 1 as the write transaction restarts.
Therefore, it is not necessary for Host Driver to set this bit to 0. If Stop At Block Gap Request is set to 1, any write to this bit is ignored.
STOP_AT_BLOCK_GAP_REQUEST - Stop At Block Gap Request
This bit is used to stop executing read and write transaction at the next block gap for non-DMA, SDMA and ADMA transfers. The Host Driver shall leave this bit set to 1 until the Transfer Complete is set to 1. Clearing both Stop At Block Gap Request and Continue Request shall not cause the transaction to restart. When Host Controller version is 1.00, the Host Driver can set this bit if the card supports Read Wait Control. When Host Controller version is 2.00 or higher, the Host Driver can set this bit regardless of the card supports Read Wait Control. The Host Controller shall stop read transfer by using Read Wait or stopping SD clock.
In case of write transfers in which the Host Driver writes data to the Buffer Data Port register, the Host Driver shall set this bit after all block data is written. If this bit is set to 1, the Host Driver shall not write data to Buffer Data Port register.
This bit affects Read Transfer Active, Write Transfer Active, DAT Line Active and Command Inhibit (DAT) in the Present State register.
Regarding detailed control of bits D01 and D00, refer to Section 3.8 and 3.12.
SD_BUS_VOLTAGE_SELECT - The Host doesn't support the bus voltage selections. For SDMMC1 Interfaces,
voltage switching between 3.3V to 1.8V is done by programming the PMU through I2C Interface.
SD_BUS_POWER - SD Bus Power
Before setting this bit, the SD Host Driver shall set SD Bus Voltage Select. If the Host Controller detects the No Card state, this bit shall be cleared.
If this bit is cleared, the Host Controller shall immediately stop driving CMD and DAT[3:0] (tri-state) and drive SDCLK to low level (refer to Section 2.2.14).
CARD_DETECT_SIGNAL_DETECT - Card Detect Signal Selection
This bit selects source for the card detection.
When the source for the card detection is switched, the interrupt should be disabled during the switching period by clearing the Interrupt Status/Signal Enable register in order to mask unexpected interrupt being caused by the glitch.
The Interrupt Status/Signal Enable should be disabled during over the period of debouncing.
CARD_DETECT_TEST_LVL - Card Detect Test Level
This bit is enabled while the Card Detect Signal Selection is set to 1 and it indicates card inserted or not.
EXTENDED_DATA_TRANSFER_WIDTH (VENDOR Bit)

1:8-bit Mode,DATA_XFER_WIDTH is ignored.
0:Card bus width is as per DATA_XFER_WIDTH value

DMA Select
This field is used to select DMA type. The Host Driver shall check support of DMA modes by referring the Capabilities register. Selected DMA is enabled by DMA Enable of the Transfer Mode register in SD mode and DMA Enable of UHS-II Transfer Mode register in UHS-II mode.

(1) Up to Version 3.00
When Host Version 4 Enable is set to 0, setting of this field is compatible to Host Controller Version 3.00.
SDMA is initiated by writing to the Command register when this field is set to 00b and the SDMA System Address register (32-bit) is used.
SDMA does not support 64-bit addressing.
ADMA2 is initiated by writing to the Command register when this field is set to 10b or 11b. Lower 32-bit of the ADMA System Address register is used when this field is set to 10b and 64-bit of the ADMA System Address register is used when this field is set to 11b. Support of 64-bit System Addressing is indicated by 64-bit System Address Support for V3 in the Capabilities register. 64-bit AMDA2 uses 96-bit Descriptor.

00 SDMA is selected
01 Reserved (New assignment is not allowed)
10 32-bit Address ADMA2 is selected
11 64-bit Address ADMA2 is selected (Optional)

(2) Version 4.00 or later
When Host Version 4 Enable is set to 1, setting of this field is changed as follows. SDMA is initiated by Host Driver writes to the Command register when this field is set to 00b.
ADMA2 is initiated by Host Driver writes to the Command register when this field is set to 10b or 11b and by ADMA3 sets to the ADMA System Address register when this field is set to 11b.
ADMA3 is initiated by Host Driver writes to the ADMA3 ID Address register when this field is set to 11b.

00 SDMA is selected
01 Not Used (New assignment is not allowed)
10 ADMA2 is selected (AMDA3 is not supported or disabled)
11 ADMA2 or ADMA3/CQE is selected

Support of 64-bit DMA and 128-bit Descriptor is indicated by 64-bit System Address Support for V4 in the Capabilities register. If the support bit is set to 1, all supported DMAs (depends on Support, ADMA2 Support and

ADMA3 Support) shall support 64-bit addressing. 64-bit Addressing in the Host Controller 2 register selects either 32-bit or 64-bit system addressing of DMAs.

HIGH_SPEED_EN - High Speed Enable
This bit is optional. Before setting this bit, the Host Driver shall check the High Speed Support in the Capabilities register. If this bit is set to 0 (default), the Host Controller outputs CMD line and DAT lines at the falling edge of the SD Clock (up to 25MHz). If this bit is set to 1, the Host Controller outputs CMD line and DAT lines at the rising edge of the SD Clock (up to 50MHz).

If Preset Value Enable in the Host Control 2 register is set to 1, Host Driver needs to reset SD Clock Enable before changing this field to avoid generating clock glitches. After setting this field, the Host Driver sets SD Clock Enable again.
DATA_XFER_WIDTH - Data Transfer Width
This bit selects the data width of the Host Controller. The Host Driver shall set it to match the data width of the SD card.
LED_CONTROL - LED Control
This bit is used to caution the user not to remove the card while the SD card is being accessed. If the software is going to issue multiple SD commands, this bit can be set during all these transactions. It is not necessary to change for each transaction.

Offset: 0x28
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,x000,xxxx,0000,xxxx,0000,0000,0000)
PROD: 0x00000020 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx1x,xxxx)

Bit	Reset	PROD	Description
26	0x0	_NONE_	**WAKEUP_ON_CARD_REMOVAL**: **0** = DISABLE **1** = ENABLE
25	0x0	_NONE_	**WAKEUP_ON_CARD_INSERTION**: **0** = DISABLE **1** = ENABLE
24	0x0	_NONE_	**WAKEUP_ON_CARD_INTERRUPT**: **0** = DISABLE **1** = ENABLE
19	0x0	_NONE_	**INTERRUPT_AT_BLOCK_GAP**: **0** = DISABLE **1** = ENABLE
18	0x0	_NONE_	**READ_WAIT_CONTROL**: **0** = DISABLE **1** = ENABLE

Bit	Reset	PROD	Description
17	0x0	_NONE_	**CONTINUE_REQUEST**: **0** = IGNORED **1** = RESTART
16	0x0	_NONE_	**STOP_AT_BLOCK_GAP_REQUEST**: **0** = TRANSFER **1** = STOP
11:9	0x0	_NONE_	**SD_BUS_VOLTAGE_SELECT**: **5** = V1_8 **6** = V3_0 **7** = V3_3
8	0x0	_NONE_	**SD_BUS_POWER**: **0** = POWER_OFF **1** = POWER_ON
7	0x0	_NONE_	**CARD_DETECT_SIGNAL_DETECT**: **0** = SDCD **1** = CARD_DTECT_TST_LVL
6	0x0	_NONE_	**CARD_DETECT_TEST_LVL**: **0** = NO_CARD **1** = CARD_INSERTED
5	0x0	BIT_8	**EXTENDED_DATA_TRANSFER_WIDTH**: **0** = NOBIT_8 **1** = BIT_8
4:3	0x0	_NONE_	**DMA_SELECT**: SW should select ADMA3_CQE in eMMC CMD queuing mode. **0** = SDMA **1** = RSVD **2** = ADMA2 **3** = ADMA3_CQE
2	0x0	_NONE_	**HIGH_SPEED_EN**: **0** = NORMAL_SPEED **1** = HIGH_SPEED
1	0x0	_NONE_	**DATA_XFER_WIDTH**: **0** = BIT_1 **1** = BIT_4
0	0x0	_NONE_	**LED_CONTROL**: **0** = OFF **1** = ON

SDMMCAB_SW_RESET_TIMEOUT_CTRL_CLOCK_CONTROL_0

Clock Control Register

SW_RESET_FOR_DAT_LINE - Software Reset For DAT Line
Only part of data circuit is reset. DMA circuit is also reset.

The following registers and bits are cleared by this bit:
Buffer Data Port register
Buffer is cleared and initialized.
Present State register
Buffer Read Enable
Buffer Write Enable
Read Transfer Active
Write Transfer Active
DAT Line Active
Command Inhibit (DAT)
Block Gap Control register
Continue Request
Stop At Block Gap Request
Normal Interrupt Status register
Buffer Read Ready
Buffer Write Ready
DMA Interrupt
Block Gap Event
Transfer Complete
SW_RESET_FOR_CMD_LINE - Software Reset For CMD Line
Only part of command circuit is reset to be able to issue a command. From Version 4.10, this bit is also used to initialize UHS-II command circuit. This reset is effective only command issuing circuit (including response error statuses related to Command Inhibit (CMD) control) and does not affect data transfer circuit. Host Controller can continue data transfer even this reset is executed during handling of sub command response errors.

The following registers and bits are cleared by this bit:
Present State register
Command Inhibit (CMD)
Normal Interrupt Status register
Command Complete
Error Interrupt Status (from Version 4.10)
Response error statuses related to Command Inhibit (CMD)
SW_RESET_FOR_ALL - Software Reset For All
This reset affects the entire Host Controller except for the card detection circuit. Register bits of type ROC, RW, RW1C, RWAC are cleared to 0. During its initialization, the Host Driver shall set this bit to 1 to reset the Host Controller. The Host Controller shall reset this bit to 0 when Capabilities registers are valid and the Host Driver can read them. Additional use of Software Reset For All may not affect the value of the Capabilities registers. If this bit is set to 1, the host driver should issue reset command and reinitialize the SD card.
DATA_TIMEOUT_COUNTER_VALUE - Data Timeout Counter Value
This value determines the interval by which DAT line timeouts are detected. For more information about timeout generation, refer to the Data Timeout Error in the Error Interrupt Status register. Timeout clock frequency will be generated by

dividing the base clock TMCLK value by this value. When setting this register, prevent inadvertent timeout events by clearing the Data Timeout Error Status Enable (in the Error Interrupt Status Enable register)
1111b Reserved
1110b TMCLK x pow(2,27)
.....
0001b TMCLK x pow(2,14)
0000b TMCLK x pow(2,13)

There are two types of busy periods in a multiple block write operation.
(1) Write busy at block gap (without CMD12) is maximum 250ms
(2) Write busy after CMD12 is maximum 250ms (500ms for SDXC)

If CMD12 is issued during a multiple block write operation busy period, the host timeout counter is reset and the 250ms (500ms for SDXC) timeout period is measured from the response of CMD12.

The duration of an erase command can be estimated by the number of write blocks (WRITE_BL) to be
erased multiplied by 250 ms.

SDCLK_FREQUENCYSELECT - SDCLK Frequency Select
This register is used to select the frequency of SDCLK pin. The frequency is not programmed directly; rather this register holds the divisor of the Base Clock Frequency For SD Clock in the Capabilities register. Only the following settings are allowed.
UPPER_SDCLK_FREQUENCYSELECT - Upper Bits of SDCLK Frequency Select
Host Controller Version 1.00 and 2.00 do not support these bits and they are treated as 00b fixed value (ROC).
Host Controller Version 3.00 shall support these bits to expand SDCLK Frequency Select to 10-bit. Bit 07-06 is assigned to bit 09-08 of clock divider in SDCLK Frequency Select.
CLOCK_GENERATOR_SELECT - Clock Generator Select
Host Controller Version 3.00 supports this bit. This bit is used to select the clock generator mode in SDCLK Frequency Select.
If the Programmable Clock Mode is supported (non-zero value is set to Clock Multiplier in the Capabilities register), this bit attribute is RW, and if not supported, this bit attribute is RO and zero is read.
This bit depends on the setting of Preset Value Enable in the Host Control 2 register.
If the Preset Value Enable = 0, this bit is set by Host Driver.
If the Preset Value Enable = 1, this bit is automatically set to a value specified in one of Preset Value registers.
PLL Enable
This bit is added from Version 4.10 for Host Controller using PLL. This feature allows Host Controller to initialize clock generator in two steps: by Internal Clock Enable and PLL Enable and to minimize output latency (ex. SDCLK/RCLK, D0 lane) from SD Clock Enable. There are two modes to

keep Host Drivers compatibility. In both modes, PL L Locked timing is indicated by Internal Clock Stable.

(1) When Host Version 4 Enable =0 (Host Driver Version 3, which does not support this bit) or this bit is not implemented, Internal Clock Enable (or SD Clock Enable) may activate PLL (exit low power mode and start locking clock).

(2) When Host Version 4 Enable =1 (Host Driver Version 4), Internal Clock Enable is set before setting this bit and then setting this bit may activate PLL (exit low power mode and start locking clock).
1 PLL is enabled
0 PLL is in low power mode
SD_CLK_EN
The Host Controller shall stop providing SDCLK or RCLK when writing this bit to 0. SDCLK/RCLK Frequency Select can be changed when this bit is 0. Then, the Host Controller shall maintain the same clock frequency until SDCLK is stopped (Stop at SDCLK=0). If the Card Inserted in the Present State register is cleared, this bit shall be cleared.

1 Enable providing SDCLK or RCLK
0 Disable providing SDCLK or RCLK

(1) SD Mode
This is the case when UHS-II Interface Enable is set to 0 in the Host Control 2 register. By setting this bit to 1, SDCLK is provided on pin number 5 (CLK). Refer to Section 1.12 Controlling SDCLK.
When PLL is used to generate clock, PLL is enabled by PLL Enable (if supported) or by SD Clock Enable (if PLL Enable is not supported).
When PLL is enabled by PLL Enable, the clock synchronization is checked by Internal Clock Stable.

INTERNAL_CLOCK_STABLE - Internal Clock Stable
As PLL Enable is added from Version 4.10, this status is expanded to check two cases. Host Driver Version 4.10 checks clock stability by this status twice after Internal Clock Enable is set and after PLL Enable is set.
Refer to Figure 3-3 in SD host spec4.1.

(1) Internal Clock Stable (when PLL Enable = 0 or not supported)
This bit is set to 1 when internal clock is stable after writing to Internal Clock Enable in this register to 1.

(2) PLL Clock Stable (when PLL Enable = 1)
Host Controller which supports PLL Enable sets this status to 0 once when PLL Enable is changed 0 to 1 and then this status is set to 1 when PLL is locked. (PLL uses an internal clock in stable as a reference clock which is enabled by Internal Clock Enable). After this bit is set to 1, Host Driver may set SD Clock Enable.

1 Ready
0 Not Ready

INTERNAL_CLOCK_EN - Internal Clock Enable
This bit is set to 0 when the Host Driver is not using the Host Controller or the Host Controller awaits a wakeup interrupt. The Host Controller should stop its internal clock to go very low power state. Still, registers shall be able to be read and written. Clock starts to oscillate when this bit is set to 1. When clock oscillation is stable, the Host Controller shall set Internal Clock Stable in this register to 1. This bit shall not affect card detection.

Offset: 0x2c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,x000,xxxx,0000,0000,0000,000x,0000)
PROD: 0x00000002 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	PROD	Description
26	RW	0x0	_NONE_	**SW_RESET_FOR_DAT_LINE**: **0** = WORK **1** = RESETED
25	RW	0x0	_NONE_	**SW_RESET_FOR_CMD_LINE**: **0** = WORK **1** = RESETED
24	RW	0x0	_NONE_	**SW_RESET_FOR_ALL**: **0** = WORK **1** = RESETED
19:16	RW	0x0	_NONE_	**DATA_TIMEOUT_COUNTER_VALUE**
15:8	RW	0x0	_NONE_	**SDCLK_FREQUENCYSELECT**: **128** = DIV256 **64** = DIV128 **32** = DIV64 **16** = DIV32 **8** = DIV16 **4** = DIV8 **2** = DIV4 **1** = DIV2 **0** = BASE
7:6	RW	0x0	_NONE_	**UPPER_SDCLK_FREQUENCYSELECT**
5	RW	0x0	_NONE_	**CLOCK_GENERATOR_SELECT**
3	RW	0x0	_NONE_	**PLL_EN**: In legacy SD mode, no separate PLL is used to generate SDCLK. SW should set this as a part of standard SDCLK generation process. PLL is used to generate RCLK in UHS-II mode

Bit	R/W	Reset	PROD	Description
2	RW	0x0	_NONE_	**SD_CLOCK_EN**: **0** = DISABLE **1** = ENABLE
1	RO	0x0	READY	**INTERNAL_CLOCK_STABLE**: **0** = NOT_READY **1** = READY
0	RW	0x0	_NONE_	**INTERNAL_CLOCK_EN**: when disabled turns off PLL in uhsII IOBRICK in uhsII mode **0** = STOP **1** = OSCILLATE

SDMMCAB_INTERRUPT_STATUS_0

Normal Interrupt Status Register

VEND_SPEC_ERR[1:0]
1:BOOT_ACK_ERR - Occurs When Boot Ack Status is not equal to '010'
0:BOOT_ACK_TIMEOUT_ERR - Occurs When Boot Ack is not received within the programmed number of cycles.
TARGET_RESP_ERROR - Not supported
SPI_ERR
Indicate when SPI Error has occured. The SPI Errors are registered in SPI_INTERRUPT_STATUS register.
Response Error
Host Controller Version 4.00 supports response error check function to avoid overhead of response error check by Host Driver during DMA execution. If Response Error Check Enable is set to1 in the Transfer Mode register, Host Controller Checks R1 or R5 response. If an error is detected in a response, this bit is set to 1.
TUNING_ERR
This bit is set when an unrecoverable error is detected in a tuning circuit except during tuning procedure (Occurrence of an error during tuning procedure is indicated by Sampling Select). By detecting Tuning Error, Host Driver needs to abort a command executing and perform tuning. To reset tuning circuit, Sampling Clock shall be set to 0 before executing tuning procedure (refer to Figure 2-29).
ADMA_ERR
This bit is set when the Host Controller detects errors during ADMA based data transfer. The state of the ADMA at an error occurrence is saved in the ADMA Error Status Register,
In addition, the Host Controller generates this Interrupt when it detects invalid descriptor data (Valid=0) at the ST_FDS state. ADMA Error State in the

ADMA Error Status indicates that an error occurs in ST_FDS state. The Host Driver may find that Valid bit is not set at the error descriptor.
AUTO_CMD12_ERR
Auto CMD12 and Auto CMD23 use this error status. This bit is set when detecting that any of the bits D00 to D05 in Auto CMD Error Status register has changed from 0 to 1. D07 is effective in case of Auto CMD12. Auto CMD Error Status register is valid while this bit is set to 1 and may be cleared with clearing of this bit (another implementation is also allowed).
CURRENT_LIMIT_ERR
By setting the SD Bus Power bit in the Power Control register, the Host Controller is requested to supply power for the SD Bus. If the Host Controller supports the Current Limit function, it can be protected from an illegal card by stopping power supply to the card in which case this bit indicates a failure status. Reading 1 means the Host Controller is not supplying power to SD card due to some failure. Reading 0 means that the Host Controller is supplying power and no error has occurred. The Host Controller may require some sampling time to detect the current limit. If the Host Controller does not support this function, this bit shall always be set to 0.
DATA_END_BIT_ERR
Occurs either when detecting 0 at the end bit position of read data which uses the DAT line or at the end bit position of the CRC Status.
DATA_CRC_ERR
Occurs when detecting CRC error when transferring read data which uses the DAT line or when detecting the Write CRC status having a value of other than "010".
DATA_TIMEOUT_ERR
Occurs when detecting one of following timeout conditions.
(1) Busy timeout for R1b,R5b type
(2) Busy timeout after Write CRC status
(3) Write CRC Status timeout
(4) Read Data timeout.
COMMAND_INDEX_ERR
Occurs if a Command Index error occurs in the command response.
COMMAND_END_BIT_ERR
Occurs when detecting that the end bit of a command response is 0.
COMMAND_CRC_ERR
Command CRC Error is generated in two cases.
(1) If a response is returned and the Command Timeout Error is set to 0 (indicating no timeout), this bit is set to 1 when detecting a CRC error in the command response.
(2) The Host Controller detects a CMD line conflict by monitoring the CMD line when a command is issued. If the Host Controller drives the CMD line to 1 level, but detects 0 level on the CMD line at the next SDCLK edge, then the Host Controller shall abort the command (Stop driving CMD line) and set this bit to 1. The Command Timeout Error shall also be set to 1 to distinguish CMD line conflict
COMMAND_TIMEOUT_ERR
Occurs only if no response is returned within 64 SDCLK cycles from the end bit of

the command. If the Host Controller detects a CMD line conflict, in which case Command CRC Error shall also be set as shown in Table 2-25, this bit shall be set without waiting for 64 SDCLK cycles because the command will be aborted by the Host Controller.
ERR_INTERRUPT
If any of the bits in the Error Interrupt Status register are set, then this bit is set. Therefore the Host Driver can efficiently test for an error by checking this bit first. This bit is read only.
CQE_INTR: Command Queuing Interrupt
This interrupt is asserted when at least one of the bits in CQIS register is set. This interrupt is cleared only by clearing the source interrupt in CQIS register.
FX_EVENT
This status is added from Version 4.10. Bit06 of response data will be stored in the R[14] of the Response register.
Basically, this interrupt is used with response check function. In this case, this status is set when R[14] of Response register is set to 1 and Response Type R1 / R5 is set to 0 in the Transfer Mode register or UHSII Transfer Mode register. If response check is disabled, this status is set when R[14] of Response register is set to 1. Host Driver needs to screen FX Event interrupt by checking response type is R1.

1 FX_EVENT is detected
0 No Event
RETUNING_EVENT - Re-Tuning Event
This status is set if Re-Tuning Request in the Present State register changes from 0 to 1.
Host Controller requests Host Driver to perform re-tuning for next data transfer. Current data transfer (not large block count) can be completed without re-tuning.
1 Re-Tuning should be performed
0 Re-Tuning is not required
CARD_INTERRUPT
Writing this bit to 1 does not clear this bit. It is cleared by resetting the SD card interrupt factor. In 1-bit mode, the Host Controller shall detect the Card Interrupt without SD Clock to support wakeup. In 4-bit mode, the card interrupt signal is sampled during the interrupt cycle, so there are some sample delays between the interrupt signal from the SD card and the interrupt to the Host System. It is necessary to define how to handle this delay.
CARD_REMOVAL
This status is set if the Card Inserted in the Present State register changes from 1 to 0. When the Host Driver writes this bit to 1 to clear this status, the status of the Card Inserted in the Present State register should be confirmed. Because the card detect state may possibly be changed when the Host Driver clear this bit and interrupt event may not be generated.
CARD_INSERTION
This status is set if the Card Inserted in the Present State register changes from 0 to 1. When the Host Driver writes this bit to 1 to clear this status, the status of the Card

Inserted in the Present State register should be confirmed. Because the card detect state may possibly be changed when the Host Driver clear this bit and interrupt event may not be generated.
BUFFER_READ_READY
This status is set if the Buffer Read Enable changes from 0 to 1. Refer to the Buffer Read Enable in the Present State register.
BUFFER_WRITE_READY
This status is set if the Buffer Write Enable changes from 0 to 1. Refer to the Buffer Write Enable in the Present State register.
DMA_INTERRUPT
This status is set if the Host Controller detects the Host DMA Buffer boundary during transfer. Refer to the Host DMA Buffer Boundary in the Block Size register. Other DMA interrupt factors may be added in the future. This interrupt shall not be generated after the Transfer Complete.
BLOCK_GAP_EVENT
If the Stop At Block Gap Request in the Block Gap Control register is set, this bit is set when both a read / write transaction is stopped at a block gap. If Stop At Block Gap Request is not set to 1, this bit is not set to 1.
XFER_COMPLETE
This bit indicates stop of transaction on three cases:
(1) Completion of a data transfer
(2) Completion of a command pairing with response-with-busy (R1b, R5b)
(3) Stop of data transfer by setting Stop At Block Gap Request in the Block Gap Control register
CMD_COMPLETE
This bit is set when get the end bit of the command response. (Except Auto CMD12)
Refer to Command Inhibit (CMD) in the Present State register.

Offset: 0x30
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,xxx0,0000,0000)

Bit	R/W	Reset	Description
31:30	RW	0x0	**VEND_SPEC_ERR**: **0** = DISABLE **3** = ENABLE
29	RW	0x0	**SPI_ERR**: **0** = NO_ERR **1** = ERR
28	RW	0x0	**TARGET_RESP_ERROR**: **0** = NO_ERROR **1** = ERROR
27	RW	0x0	**RESP_ERR**: **0** = NO_ERR **1** = ERR

Bit	R/W	Reset	Description
26	RW	0x0	**TUNING_ERR**: **0** = NO_ERR **1** = ERR
25	RW	0x0	**ADMA_ERR**: **0** = NO_ERR **1** = ERR
24	RW	0x0	**AUTO_CMD12_ERR**: **0** = NO_ERR **1** = ERR
23	RW	0x0	**CURRENT_LIMIT_ERR**: **0** = NO_ERR **1** = POWER_FAIL
22	RW	0x0	**DATA_END_BIT_ERR**: **0** = NO_ERR **1** = ERR
21	RW	0x0	**DATA_CRC_ERR**: **0** = NO_ERR **1** = ERR
20	RW	0x0	**DATA_TIMEOUT_ERR**: **0** = NO_ERR **1** = TIMEOUT
19	RW	0x0	**COMMAND_INDEX_ERR**: **0** = NO_ERR **1** = ERR
18	RW	0x0	**COMMAND_END_BIT_ERR**: **0** = NO_ERR **1** = END_BIT_ERR_GENERATED
17	RW	0x0	**COMMAND_CRC_ERR**: **0** = NO_ERR **1** = CRC_ERR_GENERATED
16	RW	0x0	**COMMAND_TIMEOUT_ERR**: **0** = NO_ERR **1** = TIMEOUT
15	RO	0x0	**ERR_INTERRUPT**: **0** = NO_ERR **1** = ERR
14	RW	0x0	**CQE_INTR**: **0** = NO_INT **1** = INT_GENERATED
13	RO	0x0	**FX_EVENT**: **0** = NO_EVENT **1** = FX_EVENT_DETECTED
12	RO	0x0	**RETUNING_EVENT**: **0** = NO_INT **1** = GEN_INT

Bit	R/W	Reset	Description
8	RO	0x0	**CARD_INTERRUPT**: **0** = NO_INT **1** = GEN_INT
7	RW	0x0	**CARD_REMOVAL**: **0** = NO_INT **1** = GEN_INT
6	RW	0x0	**CARD_INSERTION**: **0** = NO_INT **1** = GEN_INT
5	RW	0x0	**BUFFER_READ_READY**: **0** = NO_INT **1** = GEN_INT
4	RW	0x0	**BUFFER_WRITE_READY**: **0** = NO_INT **1** = GEN_INT
3	RW	0x0	**DMA_INTERRUPT**: **0** = NO_INT **1** = GEN_INT
2	RW	0x0	**BLOCK_GAP_EVENT**: **0** = NO_INT **1** = GEN_INT
1	RW	0x0	**XFER_COMPLETE**: **0** = NO_INT **1** = GEN_INT
0	RW	0x0	**CMD_COMPLETE**: **0** = NO_INT **1** = GEN_INT

SDMMCAB_INTERRUPT_STATUS_ENABLE_0

Normal Interrupt Status Enable Register

This register is used to select which interrupt status is indicated to the Host System as the interrupt. These
status bits all share the same 1-bit interrupt line. Setting any of these bits to 1 enables interrupt generation.

Offset: 0x34
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,x000,xxx0,0000,0000)

Bit	Reset	Description
31:30	0x0	**VENDOR_SPECIFIC_ERR**: **0** = DISABLE **3** = ENABLE
29	0x0	**SPI_ERR**: **0** = DISABLE **1** = ENABLE
28	0x0	**TARGET_RESP_ERROR**: **0** = NO_ERROR **1** = ERROR
27	0x0	**RESP_ERROR**: **0** = NO_ERROR **1** = ERROR
26	0x0	**TUNING_ERR**: **0** = DISABLE **1** = ENABLE
25	0x0	**ADMA_ERR**: **0** = DISABLE **1** = ENABLE
24	0x0	**AUTO_CMD12_ERR**: **0** = DISABLE **1** = ENABLE
23	0x0	**CURRENT_LIMIT_ERR**: **0** = DISABLE **1** = ENABLE
22	0x0	**DATA_END_BIT_ERR**: **0** = DISABLE **1** = ENABLE
21	0x0	**DATA_CRC_ERR**: **0** = DISABLE **1** = ENABLE
20	0x0	**DATA_TIMEOUT_ERR**: **0** = DISABLE **1** = ENABLE
19	0x0	**COMMAND_INDEX_ERR**: **0** = DISABLE **1** = ENABLE
18	0x0	**COMMAND_END_BIT_ERR**: **0** = DISABLE **1** = ENABLE
17	0x0	**COMMAND_CRC_ERR**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
16	0x0	**COMMAND_TIMEOUT_ERR**: **0** = DISABLE **1** = ENABLE
14	0x0	**CQE_INTR**: **0** = DISABLE **1** = ENABLE
13	0x0	**FX_EVENT**: **0** = DISABLE **1** = ENABLE
12	0x0	**RETUNING_EVENT**: **0** = DISABLE **1** = ENABLE
8	0x0	**CARD_INTERRUPT**: **0** = DISABLE **1** = ENABLE
7	0x0	**CARD_REMOVAL**: **0** = DISABLE **1** = ENABLE
6	0x0	**CARD_INSERTION**: **0** = DISABLE **1** = ENABLE
5	0x0	**BUFFER_READ_READY**: **0** = DISABLE **1** = ENABLE
4	0x0	**BUFFER_WRITE_READY**: **0** = DISABLE **1** = ENABLE
3	0x0	**DMA_INTERRUPT**: **0** = DISABLE **1** = ENABLE
2	0x0	**BLOCK_GAP_EVENT**: **0** = DISABLE **1** = ENABLE
1	0x0	**TRANSFER_COMPLETE**: **0** = DISABLE **1** = ENABLE
0	0x0	**COMMAND_COMPLETE**: **0** = DISABLE **1** = ENABLE

SDMMCAB_INTERRUPT_SIGNAL_ENABLE_0

Normal Interrupt Signal Enable Register
This register is used to select which interrupt status is notified to the Host System as the

interrupt. These
status bits all share the same 1-bit interrupt line. Setting any of these bits to 1 enables interrupt generation.

Offset: 0x38
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,x000,xxx0,0000,0000)

Bit	Reset	Description
31:30	0x0	**VENDOR_SPECIFIC_ERR**: **0** = DISABLE **3** = ENABLE
29	0x0	**SPI_ERR**: **0** = DISABLE **1** = ENABLE
28	0x0	**TARGET_RESP_ERROR**: **0** = NO_ERROR **1** = ERROR
27	0x0	**RESP_ERROR**: **0** = DISABLE **1** = ENABLE
26	0x0	**TUNING_ERR**: **0** = DISABLE **1** = ENABLE
25	0x0	**ADMA_ERR**: **0** = DISABLE **1** = ENABLE
24	0x0	**AUTO_CMD12_ERR**: **0** = DISABLE **1** = ENABLE
23	0x0	**CURRENT_LIMIT_ERR**: **0** = DISABLE **1** = ENABLE
22	0x0	**DATA_END_BIT_ERR**: **0** = DISABLE **1** = ENABLE
21	0x0	**DATA_CRC_ERR**: **0** = DISABLE **1** = ENABLE
20	0x0	**DATA_TIMEOUT_ERR**: **0** = DISABLE **1** = ENABLE
19	0x0	**COMMAND_INDEX_ERR**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
18	0x0	**COMMAND_END_BIT_ERR**: **0** = DISABLE **1** = ENABLE
17	0x0	**COMMAND_CRC_ERR**: **0** = DISABLE **1** = ENABLE
16	0x0	**COMMAND_TIMEOUT_ERR**: **0** = DISABLE **1** = ENABLE
14	0x0	**CQE_INTR**: **0** = DISABLE **1** = ENABLE
13	0x0	**FX_EVENT**: **0** = DISABLE **1** = ENABLE
12	0x0	**RETUNING_EVENT**: **0** = DISABLE **1** = ENABLE
8	0x0	**CARD_INTERRUPT**: **0** = DISABLE **1** = ENABLE
7	0x0	**CARD_REMOVAL**: **0** = DISABLE **1** = ENABLE
6	0x0	**CARD_INSERTION**: **0** = DISABLE **1** = ENABLE
5	0x0	**BUFFER_READ_READY**: **0** = DISABLE **1** = ENABLE
4	0x0	**BUFFER_WRITE_READY**: **0** = DISABLE **1** = ENABLE
3	0x0	**DMA_INTERRUPT**: **0** = DISABLE **1** = ENABLE
2	0x0	**BLOCK_GAP_EVENT**: **0** = DISABLE **1** = ENABLE
1	0x0	**TRANSFER_COMPLETE**: **0** = DISABLE **1** = ENABLE
0	0x0	**COMMAND_COMPLETE**: **0** = DISABLE **1** = ENABLE

SDMMCAB_AUTO_CMD12_ERR_STATUS_0

Host Control2 Register / Auto CMD Error Status Register

PRESET_VALUE_ENABLE - Preset Value Enable
Host Controller Version 3.00 supports this bit.
As the operating SDCLK frequency and I/O driver strength depend on the Host System implementation, it is difficult to determine these parameters in the Standard Host Driver. When Preset Value Enable is set, automatic SDCLK frequency generation and driver strength selection is performed without considering system specific conditions. This bit enables the functions defined in the Preset Value registers.
1 Automatic Selection by Preset Value are Enabled
0 SDCLK and Driver Strength are controlled by Host Driver
If this bit is set to 0, SDCLK Frequency Select, Clock Generator Select in the Clock Control register and Driver Strength Select in Host Control 2 register are set by Host Driver.
If this bit is set to 1, SDCLK Frequency Select, Clock Generator Select in the Clock Control register and Driver Strength Select in Host Control 2 register are set by Host Controller as specified in the Preset Value registers.
ASYNC_INTR_EN - Asynchronous Interrupt Enable
This bit can be set to 1 if a card support asynchronous interrupt and Asynchronous Interrupt Support is set to 1 in the Capabilities register. Asynchronous interrupt is effective when DAT[1] interrupt is used in 4-bit SD mode (and zero is set to Interrupt Pin Select in the Shared Bus Control register). If this bit is set to 1, the Host Driver can stop the SDCLK during asynchronous interrupt period to save power. During this period, the Host Controller continues to deliver Card Interrupt to the host when it is asserted by the Card.
1 Enabled
0 Disabled
ADDRESSING_64BIT_EN - 64bit addressing enable
Host Controller selects either of 32-bit or 64-bit addressing modes to access system memory. Whether 32-bit or 64-bit is determined by OS installed in a host system. Host Driver sets this bit depends on addressing mode of installed OS. Refer to 64-bit System Address Support in the Capabilities register.
1 64 bits addressing
0 32 bits addressing
HOST_VERSION_4_EN - Host Version 4.00 Enable
This bit selects either Version 3.00 compatible mode or Ver4.00 mode. In Version 4.00, support of 64-bit System Addressing is modified. All DMAs support 64-bit System Addressing. UHS-II supported Host Driver shall

enable this bit.
In Version 4.10, supported 32-bit Block Count for all operations.

Functions of following fields are modified.
(1) SDMA Address
SDMA uses ADMA System Address register (05Fh-058h) instead of
SDMA System Address register (Offset 003-000h)
(2) ADMA2 / ADMA3 Selection
ADMA3/CQE is selected by DMA Select in the Host Control 1 register.
This bit should be set to 1 to use ADMA3/CQE.
(3) 64bit ADMA Descriptor Size
128bit descriptor is used instead of 96-bit descriptor when 64-bit
Addressing is set to 1.
(4) Selection of 32-bit / 64-bit System Addressing
Either 32-bit or 64-bit system addressing is selected by 64-bit
Addressing bit in this register instead of DMA Select in the Host Control 1 register.
(5) 32-bit Block Count
SDMA System Address register (003h-000h) is modified to 32-bit Block Count register.

1 Version 4.00 Mode
0 Version 3.00 Compatible Mode
CMD23_EN
In memory card initialization, Host Driver Version 4.10 checks whether card
supports CMD23 by checking a bit SCR[33]. If the card supports CMD23
(SCR[33]=1), this bit is set to 1. This bit is used to select Auto CMD23 or Auto CMD12 for ADMA3
data transfer. Refer to Auto CMD Enable in the
Transfer Mode register.
ADMA2 Length Mode
This bit selects one of ADMA2 Length Modes either 16-bit or 26-bit.
1 26-bit Data Length Mode
0 16-bit Data Length Mode
UHS2_IF_EN - UHS-II Interface Enable
This bit is used to enable UHS-II Interface. Before trying to start UHS-II
initialization, this bit shall be set to 1. SD 4-bit Interface signals shall be
tri-state (input or bi-directional) or drive to low (output). Before trying to start
SD mode initialization, this bit shall be set to.
1 UHS-II Interface Enabled
0 4-bit SD Interface Enabled
SAMPLING_CLK_SEL - Sampling Clock Select
Host Controller uses this bit to select sampling clock to receive CMD and
DAT. This bit is set by tuning procedure and valid after the completion of
tuning (when Execute Tuning is cleared). Setting 1 means that tuning is
completed successfully and setting 0 means that tuning is failed. Writing 1 to
this bit is meaningless and ignored. A tuning circuit is reset by writing to 0.
This bit can be cleared with setting Execute Tuning. Once the tuning circuit
is reset, it will take time to complete tuning sequence. Therefore, Host Driver
should keep this bit to 1 to perform re-tuning sequence to compete re-tuning

sequence in a short time. Change of this bit is not allowed while the Host Controller is receiving response or a read data block. Refer to Figure 2-29.
1 Tuned clock is used to sample data
0 Fixed clock is used to sample data
EXECUTE_TUNING - Execute Tuning
This bit is set to 1 to start tuning procedure and automatically cleared when tuning procedure is completed. The result of tuning is indicated to Sampling Clock Select. Tuning procedure is aborted by writing 0. Refer to Figure 2-29 for more detail about tuning procedure.
1 Execute Tuning
0 Not Tuned or Tuning Completed
DRIVE_STRENGTH_SEL - Driver Strength Select
Host Controller output driver in 1.8V signaling is selected by this bit. In 3.3V signaling, this field is not effective. This field can be set depends on Driver Type A, C and D support bits in the Capabilities register.
This bit depends on setting of Preset Value Enable.
If Preset Value Enable = 0, this field is set by Host Driver.
If Preset Value Enable = 1, this field is automatically set by a value specified in the one of Preset Value registers.
00b Driver Type B is Selected (Default)
01b Driver Type A is Selected
10b Driver Type C is Selected
11b Driver Type D is Selected
VOLT_18_EN - 1.8V Signaling Enable
This bit controls voltage regulator for I/O cell. 3.3V is supplied to the card regardless of signaling voltage.
Setting this bit from 0 to 1 starts changing signal voltage from 3.3V to 1.8V. 1.8V regulator output shall be stable within 5ms. Host Controller clears this bit if switching to 1.8V signaling fails.
Clearing this bit from 1 to 0 starts changing signal voltage from 1.8V to 3.3V. 3.3V regulator output shall be stable within 5ms.
Host Driver can set this bit to 1 when Host Controller supports 1.8V signaling (One of support bits is set to 1: SDR50, SDR104 or DDR50 in the Capabilities register) and the card or device supports UHS-I (S18R=1. Refer to Bus Signal Voltage Switch Sequence in the Physical Layer Specification Version 3.00).
1 1.8V Signaling
0 3.3V Signaling
UHS_MODE_SEL - UHS Mode Select
This field is used to select one of UHS-I modes and effective when 1.8V Signaling Enable is set to 1.
If Preset Value Enable in the Host Control 2 register is set to 1, Host Controller sets SDCLK Frequency Select, Clock Generator Select in the Clock Control register and Driver Strength Select according to Preset Value registers. In this case, one of preset value registers is selected by this field. Host Driver needs to reset SD Clock Enable before changing this field

to avoid generating clock glitch. After setting this field, Host Driver sets SD Clock Enable again.
000b SDR12
001b SDR25
010b SDR50
011b SDR104(SD/SDIO)/HS200(eMMC)
100b DDR50(SD/SDIO)/DDR52(eMMC)
101b HS400(eMMC)
110b Reserved
111b UHS2
When SDR50, SDR104 or DDR50 is selected for SDIO card, interrupt detection at the block gap shall not be used. Read Wait timing is changed for these modes. Refer to the SDIO Specification Version 3.00 for more detail
COMMAND_NOT_ISSUED - Command Not Issued By Auto CMD12 Error
Setting this bit to 1 means CMD_wo_DAT is not executed due to an Auto CMD12 Error (D04-D01) in this register.
INDEX_ERR - Auto CMD12 Index Error
This bit is set if the Command Index error occurs in response to a command.
END_BIT_ERR - Auto CMD12 End Bit Error
This bit is set when detecting that the end bit of command response is 0.
CRC_ERR - Auto CMD12 CRC Error
This bit is set when detecting a CRC error in the command response.
TIMEOUT_ERR - Auto CMD12 Timeout Error
This bit is set if no response is returned within 64 SDCLK cycles from the end bit of command. If this bit is set to 1, the other error status bits (D04-D02) are meaningless.
NOT_EXECUTED - Auto CMD12 Not Executed
If memory multiple block data transfer is not started due to command error, this bit is not set because it is not necessary to issue Auto CMD12. Setting this bit to 1 means the Host Controller cannot issue Auto CMD12 to stop memory multiple block data transfer due to some error. If this bit is set to 1, other error status bits (D04-D01) are meaningless.
The relation between Auto CMD12 CRC Error and Auto CMD12 Timeout Error is shown below

Auto CMD12 CRC Error	Auto CMD12 Timeout Error	Kinds of error
0	0	No Error
0	1	Response Timeout Error
1	0	Response CRC Error
1	1	CMD line conflict

RESP_ERR - Auto CMD Response Error
This bit is set when Response Error Check Enable in the Transfer Mode register is set to 1 and an error is detected in either R1 response of either Auto CMD12 or Auto CMD23.

1 Error
0 No Error

Offset: 0x3c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,00x0,0000,0000,xxxx,xxxx,0x00,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PRESET_VALUE_ENABLE**: **0** = HW_SEL **1** = SW_SEL
30	RW	0x0	**ASYNC_INTR_EN**: **0** = DISABLE **1** = ENABLE
29	RW	0x0	**ADDRESSING_64BIT_EN**: **0** = DISABLE **1** = ENABLE
28	RW	0x0	**HOST_VERSION_4_EN**: **0** = DISABLE **1** = ENABLE
27	RW	0x0	**CMD23_EN**: **0** = DISABLE **1** = ENABLE
26	RW	0x0	**ADMA2_LEN_MODE**: **0** = LEN_16BIT **1** = LEN_26BIT
24	RW	0x0	**UHS2_IF_EN**: **0** = DISABLE **1** = ENABLE
23	RW	0x0	**SAMPLING_CLK_SEL**: **0** = FIXED **1** = TUNED
22	RW	0x0	**EXECUTE_TUNING**: **0** = NOT_TUNED **1** = EXECUTE
21:20	RW	0x0	**DRIVE_STRENGTH_SEL**: **0** = TYPE_B **1** = TYPE_A **2** = TYPE_C **3** = TYPE_D
19	RW	0x0	**VOLT_18_EN**: **0** = V33 **1** = V18

Bit	R/W	Reset	Description
18:16	RW	0x0	**UHS_MODE_SEL**: The following table shows the PROD mnemonic mapping to different SD and eMMC speed modes |-------------|--------------|--------------------| |PROD mnemonic | SD speed mode | eMMC speed mode | |-------------|--------------|--------------------| |prod_c_ds | DS | Legacy (SDR at 26M) | |prod_c_hs | HS | NA | |prod_c_sdr12 | SDR12 | NA | |prod_c_sdr25 | SDR25 | NA | |prod_c_sdr50 | SDR50 | SDR52 | |prod_c_sdr104 | SDR104 | NA | |prod_c_ddr52 | DDR50 | DDR52 | |prod_c_hs200 | NA | HS200 | |prod_c_hs400 | NA | HS400 | |---| Please note that tuning should be done in HS200(reg value=SDR104) mode before entering HS400 modes For HS400 mode: tuning should be in HS200 - SDR@200MHz **0** = SDR12 **1** = SDR25 **2** = SDR50 **3** = SDR104 **4** = DDR50 **5** = HS400 **6** = RSVD **7** = UHS2
7	RO	0x0	**COMMAND_NOT_ISSUED**: **0** = NO_ERR **1** = NOT_ISSUED
5	RO	0x0	**RESP_ERR**: **0** = NO_ERR **1** = ERR
4	RO	0x0	**INDEX_ERR**: **0** = NO_ERR **1** = ERR
3	RO	0x0	**END_BIT_ERR**: **0** = NO_ERR **1** = END_BIT_ERR_GENERATED
2	RO	0x0	**CRC_ERR**: **0** = NO_ERR **1** = CRC_ERR_GENERATED
1	RO	0x0	**TIMEOUT_ERR**: **0** = NO_ERR **1** = TIMEOUT
0	RO	0x0	**NOT_EXECUTED**: **0** = EXECUTED **1** = NOT_EXECUTED

SDMMCAB_CAPABILITIES_0

Lower Capabilities Register

SLOT_TYPE
00b Removable Card Slot
01b Embedded Slot for One Device
10b Shared Bus Slot
11b UHS-II Multiple Embedded Devices
ASYNC_INTR
1 Asynchronous Interrupt Supported
0 Asynchronous Interrupt Not Supported
SYSTEM_BUS_64BIT_SUPPORT - 64-bit System Bus Support for V3
Meaning of this bit is different depends on Versions
Host Controller Version 3.00 and Ver4.10 use this bit as 64-bit System Address support for V3 mode.
Host Controller Version 4.00 uses this bit as 64-bit System Address support for both V3 and V4 modes.
SDMA cannot be used in 64-bit Addressing in Version 3 mode.
If this bit is set to 1, 64-bit ADMA2 with using 96-bit Descriptor may be enabled as follows:
In case of Host Controller Version 3, 64-bit ADMA2 is enabled by DMA Select =11b in the Host Control 1 register.
In case of Host Controller Version 4, 64-bit ADMA2 for Version 3 is enabled by setting Host Version 4 Enable =0 and DMA Select = 11b.

1 64-bit System Address for V3 is Supported
0 64-bit System Address for V3 is not Supported

SYSTEM_BUS_64BIT_SUPPORT_V4 - 64-bit System Bus Support for V4
This bit is added from Version 4.10. Setting 1 to this bit indicates that the Host Controller supports 64-bit System Addressing of Version 4 mode
When this bit is set to 1, full or a part of 64-bit address should be used to decode Host Controller Registers so that Host Controller Registers can be placed above system memory area. 64-bit address decode of Host Controller Registers is effective regardless of setting to 64bit Addressing in Host Control 2.
If this bit is set to 1, 64-bit DMA Addressing for Version 4 is enabled by setting Host Version 4 Enable =1,
64-bit Addressing =1 in the Host Control 2 register. SDMA can be used and ADMA2 uses 128-bit Descriptor.

1 64-bit System Address for V4 is Supported
0 64-bit System Address for V4 is not Supported

VOLTAGE_SUPPORT_1_8_V - Voltage Support 1.8V,The Voltage Support to Card is dependent on System & Slot. The platform datasheet has this.
VOLTAGE_SUPPORT_3_0_V - Voltage Support 3.0V,The Voltage Support to Card is dependent on System & Slot. The platform datasheet has this.
VOLTAGE_SUPPORT_3_3_V - Voltage Support 3.3V,The Voltage Support to Card is dependent on System & Slot. The platform datasheet has this.

SUSPEND_RESUME_SUPPORT - Suspend/Resume Support
This bit indicates whether the Host Controller supports Suspend / Resume functionality. If this bit is 0, the Host Driver shall not issue either Suspend or Resume commands because the Suspend and Resume mechanism (Refer to 1.6) is not supported.
DMA_SUPPORT - SDMA Support
This bit indicates whether the Host Controller is capable of using SDMA to transfer data between system memory and the Host Controller directly. Version 4.10 Host Controller shall support SDMA if ADMA2 is supported.
HIGH_SPEED_SUPPORT - High Speed Support
This bit indicates whether the Host Controller and the Host System support High Speed mode and they can supply SD Clock frequency from 25MHz to 50MHz.
ADMA1_SUPPORT - ADMA1 Support
This bit indicates whether the Host Controller is capable of using ADMA1.
ADMA2_SUPPORT - ADMA2 Support
This bit indicates whether the Host Controller is capable of using ADMA2. Version 4.10 Host Controller shall support ADMA2 if ADMA3 is supported.
EXTENDED_MEDIA_BUS_SUPPORT
Setting to 1,indicates 8-bit data bus is supported.
MAX_BLOCK_LENGTH - Max Block Length
This value indicates the maximum block size that the Host Driver can read and write to the buffer in the Host Controller. The buffer shall transfer this block size without wait cycles. Three sizes can be defined as indicated below. It is noted that transfer block length shall be always 512 bytes for SD Memory Cards regardless this field.
BASE_CLOCK_FREQUENCY - Base Clock Frequency For SD Clock
This value indicates the base (maximum) clock frequency for the SD Clock. Unit values are 1MHz. If the real frequency is 16.5MHz, the lager value shall be set 01 0001b (17MHz) because the Host Driver use this value to calculate the clock divider value (Refer to the SDCLK Frequency Select in the Clock Control register.) and it shall not exceed upper limit of the SD Clock frequency. The supported clock range is 10MHz to 63MHz. If these bits are all 0, the Host System has to get information via another method.
Not 0 - 1MHz to 63MHz
000000b- Get information via another method
TIMEOUT_CLOCK_UNIT - Timeout Clock Unit
This bit shows the unit of base clock frequency used to detect Data Timeout Error.
0 KHz
1 MHz
TIMEOUT_CLOCK_FREQUENCY - Timeout Clock Frequency
This bit shows the base clock frequency used to detect Data Timeout Error. The Timeout Clock Unit defines the unit of this fields value.
Timeout Clock Unit =0 [KHz] unit: 1KHz to 63KHz
Timeout Clock Unit =1 [MHz] unit: 1MHz to 63MHz
Not 0 - 1KHz to 63KHz or 1MHz to 63MHz
000000b - Get information via another method

Offset: 0x40
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x3f6cd08c (0b0011,1111,011x,1100,1101,0000,1x00,1100)

Bit	Reset	Description
31:30	0x0	**SLOT_TYPE**: **0** = REMOVABLE **1** = EMBEDDED **2** = SHARED **3** = UHS2_MULTIPLE_EMBEDDED
29	0x1	**ASYNC_INTR**: **0** = NOT_SUPPORTED **1** = SUPPORTED
28	0x1	**SYSTEM_BUS_64BIT_SUPPORT**: **0** = NOT_SUPPORTED **1** = SUPPORTED
27	0x1	**SYSTEM_BUS_64BIT_SUPPORT_V4**: **0** = NOT_SUPPORTED **1** = SUPPORTED
26	0x1	**VOLTAGE_SUPPORT_1_8_V**: **0** = NOT_SUPPORTED **1** = SUPPORTED
25	0x1	**VOLTAGE_SUPPORT_3_0_V**: **0** = NOT_SUPPORTED **1** = SUPPORTED
24	0x1	**VOLTAGE_SUPPORT_3_3_V**: **0** = NOT_SUPPORTED **1** = SUPPORTED
23	0x0	**SUSPEND_RESUME_SUPPORT**: **0** = NOT_SUPPORTED **1** = SUPPORTED
22	0x1	**DMA_SUPPORT**: **0** = NOT_SUPPORTED **1** = SUPPORTED
21	0x1	**HIGH_SPEED_SUPPORT**: **0** = NOT_SUPPORTED **1** = SUPPORTED
19	0x1	**ADMA2_SUPPORT**: **0** = NOT_SUPPORTED **1** = SUPPORTED
18	0x1	**EXTENDED_MEDIA_BUS_SUPPORT**: **0** = NOT_SUPPORTED **1** = SUPPORTED

Bit	Reset	Description
17:16	0x0	**MAX_BLOCK_LENGTH**: **0** = BYTE512 **1** = BYTE1024 **2** = BYTE2048 **3** = RESERVED
15:8	0xd0	**BASE_CLOCK_FREQUENCY**
7	0x1	**TIMEOUT_CLOCK_UNIT**: MHz **0** = KHZ **1** = MHZ
5:0	0xc	**TIMEOUT_CLOCK_FREQUENCY**: 12MHz TMCLK is used in legacy SD/eMMC mode TMCLK freq value will be advertised based on UHS2_IF_EN and USE_TMCLK_FOR_DATA_TIMEOUT

SDMMCAB_CAPABILITIES_HIGHER_0

Higher Capabilities Register

1.8V VDD2 Support
This bit indicates that support of VDD2 on the Host System.
0b 1.8V VDD2 is not supported
1b 1.8V VDD2 is supported
ADMA3_SUPPORT - ADMA3 Support
This bit indicates that support of ADMA3 on Host Controller.
0b ADMA3 is not supported
1b ADMA3 is supported
Clock Multiplier
This field indicates clock multiplier value of programmable clock generator.
Refer to Clock Control register. Setting 00h means that Host Controller
does not support programmable clock generator.
00h Clock Multiplier is Not Supported
01h Clock Multiplier M = 2
02h Clock Multiplier M = 3
....
Timer Count for Re-Tuning
0h Re-Tuning timer disabled
1h 1 seconds
2h 2 seconds
3h 4 seconds
. .....
. .....
nh 2**(n-1) seconds

. .....
. .....
Fh Get information from other source
Driver Type D Support
This bit indicates support of Driver Type D for 1.8 Signaling.
1 Driver Type D is Supported
0 Driver Type D is Not Supported
Driver Type C Support
This bit indicates support of Driver Type C for 1.8 Signaling.
1 Driver Type C is Supported
0 Driver Type C is Not Supported
Driver Type A Support
This bit indicates support of Driver Type A for 1.8 Signaling.
1 Driver Type A is Supported
0 Driver Type A is Not Supported
UHS2_SUPPORT - UHS-II Support (UHS-II only)
This bit indicates whether Host Controller supports UHS-II. If this bit is set to 1, 1.8V VDD2 Support shall be set to 1 (Host System shall support VDD2 power supply).
1 UHS-II is supported
0 UHS-II is not supported
DDR50 Support
1 DDR50 is Supported
0 DDR50 is Not Supported
SDR104 Support
SDR104 requires tuning.
1 SDR104 is Supported
0 SDR104 is Not Supported
SDR50 Support
If SDR104 is supported, this bit shall be set to 1. Bit 40 indicates whether SDR50 requires tuning or not.
1 SDR50 is Supported
0 SDR50 is Not Supported

Offset: 0x44
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x18002f73 (0bxxx1,1xxx,0000,0000,001x,1111,x111,0011)
PROD: 0x00000004 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	PROD	Description
28	0x1	_NONE_	**VDD2_1_8V_SUPPORT**: **0** = NOT_SUPPORTED **1** = SUPPORTED
27	0x1	_NONE_	**ADMA3_SUPPORT**: **0** = NOT_SUPPORTED **1** = SUPPORTED
23:16	0x0	_NONE_	**CLOCK_MULTIPLIER**
15:14	0x0	_NONE_	**RETUNING_MODES**: **0** = MODE1 **1** = MODE2 **2** = MODE3 **3** = MODE4
13	0x1	_NONE_	**SDR50_TUNING**: **0** = NOT_REQUIRED **1** = REQUIRED
11:8	0xf	_NONE_	**RETUNING_TIMER_COUNT**
6	0x1	_NONE_	**TYPE_D_DRIVER**: **0** = NOT_SUPPORTED **1** = SUPPORTED
5	0x1	_NONE_	**TYPE_C_DRIVER**: **0** = NOT_SUPPORTED **1** = SUPPORTED
4	0x1	_NONE_	**TYPE_A_DRIVER**: **0** = NOT_SUPPORTED **1** = SUPPORTED
3	0x0	_NONE_	**UHS2_SUPPORT**: **0** = NOT_SUPPORTED **1** = SUPPORTED
2	0x0	SUPPORTED	**DDR50**: **0** = NOT_SUPPORTED **1** = SUPPORTED
1	0x1	_NONE_	**SDR104**: **0** = NOT_SUPPORTED **1** = SUPPORTED
0	0x1	_NONE_	**SDR50**: **0** = NOT_SUPPORTED **1** = SUPPORTED

SDMMCAB_MAXIMUM_CURRENT_0

Maximum Current Capabilities Register

Offset: 0x48
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:16	0x0	**MAXIMUM_CURRENT_FOR_1_8V**: Maximum Current for 1.8V VDD1
15:8	0x0	**MAXIMUM_CURRENT_FOR_3_0V**: Maximum Current for 3.0V VDD1
7:0	0x0	**MAXIMUM_CURRENT_FOR_3_3V**: Maximum Current for 3.3V VDD1

SDMMCAB_MAXIMUM_CURRENT_HI_0

Maximum Current Capabilities2 Register

Offset: 0x4c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**MAXIMUM_CURRENT_FOR_1_8V_VDD2**: Maximum Current for 1.V VDD2

SDMMCAB_FORCE_EVENT_0

Force Event for Auto CMD12 Error Status Register

The Force Event Register is not a physically implemented register. Rather, it is an address at which the
Auto CMD12 Error Status Register can be written.
Writing 1 : set each bit of the Auto CMD12 Error Status Register
Writing 0 : no effect

Offset: 0x50
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,xxxx,xxxx,0x00,0000)

Bit	Reset	Description
31:30	0x0	**VENDOR_SPECIFIC_ERR_STATUS**: **0** = DISABLE **3** = ENABLE
28	0x0	**TARGET_RESP_ERROR**: **0** = NO_ERROR **1** = ERROR
27	0x0	**RESP_ERROR**: **0** = NO_ERROR **1** = ERROR
26	0x0	**TUNING_ERR**: **0** = DISABLE **1** = ENABLE
25	0x0	**ADMA_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
24	0x0	**AUTOCMD12_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
23	0x0	**CURRENTLIMIT_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
22	0x0	**DATA_END_BIT_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
21	0x0	**DATACRC_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
20	0x0	**DATATIMEOUT_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
19	0x0	**COMMAND_INDEX_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
18	0x0	**COMMAND_END_BIT_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
17	0x0	**COMMAND_CRC_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT

Bit	Reset	Description
16	0x0	**COMMAND_TIMEOUT_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
7	0x0	**AUTO_CMD12_NOT_ISSUED**: **0** = NO_INTERRUPT **1** = INTERRUPT
5	0x0	**AUTO_CMD12_RESP_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
4	0x0	**AUTO_CMD12_INDEX_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
3	0x0	**AUTO_CMD12_END_BIT_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
2	0x0	**AUTO_CMD12_CRC_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
1	0x0	**AUTO_CMD12_TIMEOUT_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
0	0x0	**AUTO_CMD12_NOT_EXECUTED**: **0** = NO_INTERRUPT **1** = INTERRUPT

SDMMCAB_ADMA_ERR_STATUS_0

ADMA Error Status Register

ADMA_LENGTH_MISMATCH_ERR - ADMA Length Mismatch Error
This error occurs in the following 2 cases.
(1) While Block Count Enable being set, the total data length specified by the Descriptor table is different from that specified by the Block Count and Block Length.
(2) Total data length can not be divided by the block length.
ADMA_ERR_STATE - ADMA Error State
This field indicates the state of ADMA when error is occurred during ADMA data transfer. This field never indicates "10" because ADMA never stops in this state.

|D01 - D00 |ADMA Error State when | Contents of SYS_SDR register |
| | error is occurred | |
---|
| 00 | ST_STOP (Stop DMA) | Points next of the error descriptor|
| 01 | ST_FDS (Fetch Descriptor) | Points the error descriptor |

| 10 | Never set this state | (Not used) |
| 11 | ST_TFR (Transfer Data) | Points the next of the error |
| | | descriptor |
----------|

Offset: 0x54
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ADMA_LENGTH_MISMATCH_ERR**: **0** = NO_ERR **1** = ERR
1:0	0x0	**ADMA_ERR_STATE**

SDMMCAB_ADMA_SYSTEM_ADDRESS_0

ADMA System Address Register
The 32-bit addressing Host Driver uses lower 32-bit of this register (upper 32-bit should be set to 0) and shall program Descriptor Table on 32-bit boundary and set 32-bit boundary address to this register. DMA2/3 ignores lower 2-bit of this register and assumes it to be 00b.
DMA in 64-bit addressing. The 64-bit addressing Host Driver uses all bits of this register and shall program Descriptor Table on 64-bit boundary and set 64-bit boundary address to this register. DMA2/3 ignores lower 3-bit of this register and assumes it to be 000b.
(1) SDMA
If Host Version 4.00 Enable is set to 1, SDMA use this register to indicate System Address of data location instead of using SDMA System Address register (Offset 003-000h). SDMA can be used in 32-bit and 64-bit addressing in Version 4.00.
(2) ADMA2
This register holds byte address of executing command of the Descriptor table. At the start of
ADMA2, the Host Driver shall set start address of the Descriptor table. The ADMA increments this register address, which points to next line, when every fetching a Descriptor line. When the ADMA Error Interrupt is generated, this register shall hold the Descriptor address depending on the ADMA state.
(3) ADMA3
This register is set by ADMA3. Host Driver is not necessary to set this register. The ADMA3 increments address of this register, which points to next line, when

every time fetching a Descriptor line. When Error Interrupt is generated, this register shall hold the Descriptor address depending on the ADMA state.

Offset: 0x58
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ADMA_SYSTEM_ADDRESS**

SDMMCAB_UPPER_ADMA_SYSTEM_ADDRESS_0

Upper ADMA System Address Register

This register is used by 64-bit address descriptor.
Upper bits of 64 bit address - ADMA system address[63:32] is set in this register

Offset: 0x5c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UPPER_ADMA_SYSTEM_ADDRESS**

SDMMCAB_PRESET_DEFAULT_AND_INIT_0

Preset Value Register Indexes

15-14 HwInit Driver Strength Select Value
Driver Strength is supported by 1.8V signaling bus speed modes. This field is meaningless for 3.3V signaling.
11b Driver Type D is Selected
10b Driver Type C is Selected
01b Driver Type A is Selected
00b Driver Type B is Selected
13-11 Rsvd Reserved

Clock Generator Select Value
This bit is effective when Host Controller supports programmable clock generator.
1 Programmable Clock Generator
0 Host Controller Ver2.00 Compatible Clock Generator
SDCLK Frequency Select Value
10-bit preset value to set SDCLK Frequency Select in the Clock Control Register is described by a host system.
Preset Value for Default Speed and Initialization

Offset: 0x60
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00040000 (0b00xx,x000,0000,0100,00xx,x000,0000,0000)

Bit	Reset	Description
31:30	0x0	**DRIVE_STRENGTH_VAL_HIGH**
26	0x0	**CLK_GEN_VAL_HIGH**
25:16	0x4	**SDCLK_FREQ_VAL_HIGH**
15:14	0x0	**DRIVE_STRENGTH_VAL_LOW**
10	0x0	**CLK_GEN_VAL_LOW**
9:0	0x0	**SDCLK_FREQ_VAL_LOW**

SDMMCAB_PRESET_SDR12_AND_HIGH_0

Preset Value for SDR12 Speed and HIGHSPEED

Offset: 0x64
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00040002 (0b00xx,x000,0000,0100,00xx,x000,0000,0010)

Bit	Reset	Description
31:30	0x0	**DRIVE_STRENGTH_VAL_HIGH**
26	0x0	**CLK_GEN_VAL_HIGH**

Bit	Reset	Description
25:16	0x4	**SDCLK_FREQ_VAL_HIGH**
15:14	0x0	**DRIVE_STRENGTH_VAL_LOW**
10	0x0	**CLK_GEN_VAL_LOW**
9:0	0x2	**SDCLK_FREQ_VAL_LOW**

SDMMCAB_PRESET_SDR50_AND_SDR25_0

Preset Value for SDR50 and SDR25

Offset: 0x68
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00010002 (0b00xx,x000,0000,0001,00xx,x000,0000,0010)

Bit	Reset	Description
31:30	0x0	**DRIVE_STRENGTH_VAL_HIGH**
26	0x0	**CLK_GEN_VAL_HIGH**
25:16	0x1	**SDCLK_FREQ_VAL_HIGH**
15:14	0x0	**DRIVE_STRENGTH_VAL_LOW**
10	0x0	**CLK_GEN_VAL_LOW**
9:0	0x2	**SDCLK_FREQ_VAL_LOW**

SDMMCAB_PRESET_DDR50_AND_SDR104_0

Preset Value for DDR50 and SDR104

Offset: 0x6c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00020000 (0b00xx,x000,0000,0010,00xx,x000,0000,0000)

Bit	Reset	Description
31:30	0x0	**DRIVE_STRENGTH_VAL_HIGH**
26	0x0	**CLK_GEN_VAL_HIGH**
25:16	0x2	**SDCLK_FREQ_VAL_HIGH**
15:14	0x0	**DRIVE_STRENGTH_VAL_LOW**
10	0x0	**CLK_GEN_VAL_LOW**
9:0	0x0	**SDCLK_FREQ_VAL_LOW**

SDMMCAB_ADMA3_INT_DESC_LOWER_ADDRESS_0

ADMA3 Integrated Descriptor Address 31:0 Register

The start address of Integrated DMA Descriptor is set to this register. Writing to a specific address starts ADMA3 depends on 32-bit/64-bit addressing. The ADMA3 fetches one Descriptor Address and increments this field to indicate the next Descriptor address.
The 32-bit addressing Host Driver uses lower 32-bit of this register and shall program Descriptor Table on 32-bit boundary. ADMA3 ignores lower 2-bit of this register and assumes it to be 00b. Writing to 07Bh starts ADMA3 data transfer.
The 64-bit addressing Host Driver uses all 64-bit of this register and shall program Descriptor Table on 64-bit boundary. ADMA3 ignores lower 3-bit of this register and assumes it to be 000b. Writing to 07Fh starts ADMA3 data transfer.

Register Value Addressing Mode
00000000_xxxxxxxxh 32-bit System Address
xxxxxxxx_xxxxxxxxh 64-bit System Address

Offset: 0x78
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ADDRESS_31_0**

SDMMCAB_ADMA3_INT_DESC_UPPER_ADDRESS_0

ADMA3 Integrated Descriptor Address 63:32 Register

Offset: 0x7c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ADDRESS_63_32**

SDMMCAB_VENDOR_REGS_PTR_0

Pointer Registers to 1FFh-100h Area - vendor specific area
Area of offset mFFh-m00h is defined as re-locatable area. The locations of following register sets are pointed by offset address.
vendor registers start at 100h. => m=1
Vendor registers pointer

Offset: 0xe8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000100 (0bxxxx,xxxx,xxxx,xxxx,xxxx,0001,0000,0000)

Bit	Reset	Description
11:0	0x100	**VENDOR_PTR**: Vendor regs start at 0x100 - offset[11:0]=0x100

SDMMCAB_SLOT_INTERRUPT_STATUS_0

Slot Interrupt Status Register

VENDOR_VERSION_NUMBER - Vendor Version Number
This status is reserved for the vendor version number. The Host Driver

should not use this status.
SPECIFICATION_VERSION_NUMBER - Specification Version Number
This status indicates the Host Controller Spec. Version. The upper and lower 4-bits indicate the version.
00 SD Host Specification Version 1.00
01 SD Host Specification Version 2.00
Including the feature of the ADMA and Test Register,
02 SD Host Specification Version 3.00
03 SD Host Specification Version 4.00
04h SD Host Controller Specification Version 4.10
05h SD Host Controller Specification Version 4.20
others Reserved
INTERRUPT_SIGNAL_FOR_EACH_SLOT - Interrupt Signal For Each Slot
These status bits indicate the logical OR of Interrupt Signal and Wakeup Signal for each slot. A maximum of 8 slots can be defined. If one interrupt signal is associated with multiple slots, the Host Driver can know which interrupt is generated by reading these status bits. By a power on reset or by setting Software Reset For All, the interrupt signal shall be de-asserted and this status shall read 00h.
Bit 00 Slot 1
Bit 01 Slot 2
Bit 02 Slot 3
....
Bit 07 Slot 8

Offset: 0xfc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x05050000 (0b0000,0101,0000,0101,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31:24	0x5	**VENDOR_VERSION_NUMBER**
23:16	0x5	**SPECIFICATION_VERSION_NUMBER**
7:0	0x0	**INTERRUPT_SIGNAL_FOR_EACH_SLOT**

SDMMCAB_VENDOR_CLOCK_CNTRL_0

The following Registers are Vendor Specific Registers and are mapped to Vendor Specific Address Space(0x100 - 0x1FF)

Offset: 0x100
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0804d06d (0bxx00,1000,0000,0100,1101,0000,x110,1101)
PROD: 0x12070020 (0bxxx1,0010,0000,0111,xxxx,xxxx,x01x,0x0x)

Bit	Reset	PROD	Description
29	0x0	_NONE_	**DIFF_CLK_SEL**: If set, selects differential CLK and DQS; Used by eMMC IOBRICK only. SW should set this appropriately based on eMMC part used. E_INPUT_* (CMD/DAT/CLK) of emmc IOBRICK should be set to 0 when differential signaling is used. default is '0' - selects single ended signaling for clk/dqs
28:24	0x8	0x12	**TRIM_VAL**: Trimmer tap value for the output data path trimmer This determines the trimmer value needed to drive the output data correctly. The tap for outbound trimmer is single MUX. The trim settings required are within very small (0-3) with absolute delay requirement of ~400ps. Minimal change with PVT variations.
23:16	0x4	0x7	**TAP_VAL**: Tap value for input data path trimmer This determines the tap value needed to sample the input data correctly. Delay per each tap can range from 70ps (hv_ff) to 505ps (lv_ss).
15:8	0xd0	_NONE_	**BASE_CLK_FREQ**: SW driver should write core clock frequency value in MHz to this field to advertise base frequency in SDMMCA_CAPABILITIES_0_ BASE_CLOCK_FREQUENCY for standard SD driver usage.
6	0x1	NORMAL	**LEGACY_CLKEN_OVERRIDE**: Override for sdmmc_legacy_g_clk clken; Set this to 0 to save power **0** = NORMAL :0 -> sdmmc_legacy_g_clk is gated **1** = OVERRIDE :1 -> sdmmc_legacy_g_clk is not gated
5	0x1	OVERRIDE	**SDR50_TUNING_OVERRIDE**: override the SDR50_TUNING capabilities bit. Software should only set this bit if it is required to use Tuning for SDR50. (only supported for SDMMC1) **0** = NORMAL :0 -> No Tuning support advertised for SDR50 mode. **1** = OVERRIDE :1 -> Tuning support is enabled for SDR50 mode.

Bit	Reset	PROD	Description
4	0x0	_NONE_	**UHS2_CAPABILITY_OVERRIDE**: override the UHS-II capabilities bit.
3	0x1	0x0	**PADPIPE_CLKEN_OVERRIDE**: Override for padmacro and pipemacro clken. **0** = NORMAL :0 -> CLKEN is de-asserted when internal CLKEN is de-asserted. **1** = OVERRIDE :1 -> CLKEN is kept asserted even when internal CLKEN is de-asserted.
2	0x1	_NONE_	**SPI_MODE_CLKEN_OVERRIDE**: This mode is not supported.
1	0x0	FEEDBACK	**INPUT_IO_CLK**: Feedback clock is selected by default. Software should not change this. Disabling Feedback clock will select Internal Clock that requires different TAP Value Programming. **0** = FEEDBACK **1** = INTERNAL
0	0x1	_NONE_	**SDMMC_CLK**: This is set when sdmmc_clk is supplied by the CAR module. Prior to sdmmc_clk switch OFF, this bit should be written as '0'. By writing zero, the asynchronous card interrupt is routed to the Interrupt controller. **0** = DISABLE **1** = ENABLE

SDMMCAB_VENDOR_SYS_SW_CNTRL_0

Offset: 0x104
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x38600002 (0b0011,1000,011x,xxxx,x000,0000,x000,0010)

Bit	Reset	Description
31	0x0	**ENHANCED_STROBE_MODE**: Enables enhanced strobe mode in HS400 mode for eMMC5.x devices 0 - cmd_in(Resp) is sampled by loopback clock which requires tuning 1 - cmd_in(Resp) is sampled by DQS_in which requires no tuning SW has to set this bit appropriately based on device capability since this is an optional feature for eMMC5.x devices. If device supports this feature, SW should set this bit to avoid tuning.

Bit	Reset	Description
30	0x0	**USE_TMCLK_FOR_WR_CRC_STATUS_TIMEOUT**: When set, uses TMCLK data timeout counter for generating wr_crc_status data-timeout When cleared, uses sdmmc_clk for maintaining wr_crc_status data timeout counter
29	0x1	**USE_NCRC_FOR_WR_CRC_STATUS_TIMEOUT_VAL**: This field is valid only when USE_TMCLK_FOR_WR_CRC_STATUS_TIMEOUT is set to 0. When cleared, uses data timeout value as wr crc status timeout value (spec defined one) When set, uses Ncrc cycles as timeout value
28	0x1	**USE_TMCLK_FOR_DATA_TIMEOUT**: When set, uses TMCLK data timeout counter for generating legacy data timeout error (except wr_crc_status timeout) When cleared, uses sdmmc_clk for maintaining data timeout counter
27:24	0x8	**DEVICE_BUSY_WAIT_CYCLES**: This register field is used to load wait_cycles counter before device busy sampling in HS400 mode. Please note that this counter is used only in HS400 mode.
23	0x0	**ALLOW_CARD_CLK_STALLS_IN_WR**: When set, allows card clock stopping during transfer of data within a block in DDR52/HS400 writes.
22	0x1	**EMMC_IOBRICK_CLK_DATA**: Used to drive AP_CLK and AN_CLK input of iobrick. 0x1 - clk_out will be same as iobrick_clk_in 0x0 - clk_out will be inverted iobrick_clk_in
21	0x1	**QUALIFY_WITH_RD_DATA_VLD**: We have async FIFOs in both cmd_in and dat_in paths in padmacro which are used in tunable modes. When this bit set, rdata from FIFO is treated as valid data only when rd_req is high. This is needed to handle bubbles on 'rd_req' when MTBF is high.
14	0x0	**SD_BUS_POWER_ON_OFF_INT_STATUS**: SD_BUS_POWER was changed. System software can use this interrupt to implement power switch.
13	0x0	**VOLT_SWITCH_INT_STATUS**: VOLT_18_EN was changed. System software can use this interrupt to implement a UHS-I voltage switch procedure for a standard SD Host driver **0** = NO_INT **1** = GEN_INT
12	0x0	**TUNING_SYS_INT_STATUS**: CMD19 was issued while EXECUTE_TUNING was set. System software can use this interrupt to implement a UHS-I tuning procedure for a standard SD Host driver. **0** = NO_INT **1** = GEN_INT

Bit	Reset	Description
11:8	0x0	**TUNING_ASYNC_FIFO_ADDNL_DELAY**: This register field holds the additional delay in cycles which should be added to round trip delay Default value is - zero. SW should not update this field unless a new PROD setting is given.
6	0x0	**SD_BUS_POWER_ON_OFF_INT_ENABLE**: Enables a seperate system interrupt (i.e., not the standard SDHCI interrupt) when SD_BUS_POWER is changed. **0** = DISABLE **1** = ENABLE
5	0x0	**VOLT_SWITCH_INT_ENABLE**: Enables a seperate system interrupt (i.e., not the standard SDHCI interrupt) when VOLT_18_EN is changed. **0** = DISABLE **1** = ENABLE
4	0x0	**TUNING_SYS_INT_ENABLE**: Enables a seperate system interrupt (i.e., not the standard SDHCI interrupt) when CMD19 is issued while EXECUTE_TUNING is set. **0** = DISABLE **1** = ENABLE
3	0x0	**ASSERT_BUFF_RD_RDY_INT**: Write a 1 to this field to assert sdmmc_interrupt_status_0_buffer_read_ready. Used by the system software that implements the tuning procedure to signal to the standard SD driver that the tuning process has completed **0** = DISABLE **1** = ENABLE
2	0x0	**IO_TRIM_BYPASS**: Override bit for selecting between core trimmer (Vcore dependent) and io trimmer (custom trimmer) in IB clock path Default option is IO trimmer; SW should not set this field.
1	0x1	**INT_MASK_WHILE_TUNING**: As per spec, Host should not generate any interrupts (including cmd_complete and data_xfer_complete) except buffer_read_ready interrupt during tuning sequence is being performed SW can override this behavior by clearing this bit - but this leads to a spec violation **0** = DISABLE **1** = ENABLE
0	0x0	**SPI_MODE**: This mode is not supported.

SDMMCAB_VENDOR_ERR_INTR_STATUS_0

Legacy Interrupt Status Register

The fields are valid when a error interrupt has occurred.

Offset: 0x108
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000X0000 (0bxxxx,xxxx,xxxx,xx00,0000,xxx0,0000,0000)

Bit	Reset	Description
18	X	**SDMMC_LEGACY_CTLR_IDLE**: indicates legacy SD interface controller is idle - no active data transfers on legacy SD interface
17	0x0	**READ_DATA_TIMEOUT**: valid when a data timeout error occurs
16	0x0	**WRITE_CRC_STATUS_TIMEOUT**: valid when a data timeout error occurs
15	0x0	**WRITE_BUSY_TIMEOUT**: valid when a data timeout error occurs
14	0x0	**RESP_BUSY_TIMEOUT**: valid when a data timeout error occurs
13	0x0	**SPI_WRITE_BUSY_TIMEOUT**
12	0x0	**SPI_RX_START_TOKEN_TIMEOUT**
8:5	0x0	**SPI_DAT_ERR_TOKEN**: Data Error Token, while read from card.
4:0	0x0	**SPI_DAT_RESPONSE**: Data Response while write to card **5** = DATA_ACCEPTED **11** = CRC_ERR **13** = WRITE_ERR

SDMMCAB_VENDOR_CAP_OVERRIDES_0

Capabilities override bits

Offset: 0x10c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00001107 (0bxxxx,xxxx,xxxx,xxxx,xx01,0001,xxxx,0111)

Bit	Reset	Description
13:8	0x11	**DQS_TRIM_VAL**: Tap value for incoming DQS path trimmer - used in HS400 modes
3	0x0	**DRV_LPBK_CLK_ON_CMD_LINE**: Loopback trimmed clock will be driven onto cmd line, if this bit set to 1. Should be set to zero during normal data transfers. Useful in debug.
2	0x1	**VOLTAGE_3_3_V_SUPPORT_OVERRIDE**: Voltage support 3_3_V override
1	0x1	**VOLTAGE_3_0_V_SUPPORT_OVERRIDE**: Voltage support 3_0_V override
0	0x1	**VOLTAGE_1_8_V_SUPPORT_OVERRIDE**: Voltage support 1_8_V override

SDMMCAB_VENDOR_MISC_CNTRL_0

Misc Vendor Cntrl Register

SDMMC_SPARE0: Spare register bits with reset value of 0

- SDMMC_SPARE0[0] : SW_RESET_CLKEN_OVERRIDE, override the sdmmc_clken when doing SW_RESET if set to 1.
- SDMMC_SPARE0[1] : When set, allows SD clock to be stopped in the middle of a read data block while in SDR104/HS400 modes(allow_sdr104_intrablock_stalls).
- Unsafe for some SD/eMMC cards, but may improve SDR104 DMA read performance in some cases.
- SDMMC_SPARE0[2] : When set, SDR104 support is advertised in SDMMC_CAPABILITIES_HIGHER_0_SDR104
- SDMMC_SPARE0[3] : When set, SDR50 support is advertised in SDMMC_CAPABILITIES_HIGHER_0_SDR50
- SDMMC_SPARE0[5] : When 0, masks the pad macro's "high speed" enable to 0, causing the pad macro to always launch
- data on the falling edge of the clock. This prevents the SD Host driver's setting of
- SDMMC_POWER_CONTROL_HOST_x_HIGH_SPEED_EN from undesirably affecting the output timing.
- SDMMC_SPARE0[7:6] : Number of pipe stages
- SDMMC_SPARE0[8] : When set, DDR50 support is advertised in SDMMC_CAPABILITIES_HIGHER_0_DDR50.

SDMMC_SPARE1: Spare register bits with reset value of 1

- SDMMC_SPARE1[0] : Reserved

- SDMMC_SPARE1[1] : Reserved

Offset: 0x120
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffff0098 (0b1111,1111,1111,1111,0000,0000,1001,1000)
PROD: 0x00000298 (0bxxxx,xxxx,xxxx,xxxx,0000,0010,1001,100x)

Bit	Reset	PROD	Description
31:16	0xffff	_NONE_	**SDMMC_SPARE1**: Spare register bits with reset value of 1
15:1	0x4c	0x14c	**SDMMC_SPARE0**: Spare register bits with reset value of 0x4C
0	0x0	_NONE_	**ERASE_TIMEOUT_LIMIT**: Erase timeout value. **0** = FINITE :Finite, It is limited to the programmed value in the DATA_TIMEOUT_VALUE **1** = INFINITE :Infinite, Controller would be monitoring until the card is busy.

SDMMCAB_VENDOR_MISC_CNTRL1_0

Offset: 0x124
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:16	0x0	**OVERRIDE_FOR_1_8V**: Maximum override for 1.8V VDD1
15:8	0x0	**OVERRIDE_FOR_3_0V**: Maximum override for 3.0V VDD1
7:0	0x0	**OVERRIDE_FOR_3_3V**: Maximum override for 3.3V VDD1

SDMMCAB_VENDOR_MISC_CNTRL2_0

Offset: 0x128
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x63e00000 (0b0110,0011,111x,0000,0000,0000,0000,0000)
PROD: 0x00000000 (0bx0xx,xx00,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	PROD	Description
30	0x1	0x0	**SDMMC_CLK_OVR_ON**: Master clk en Override bit for all SLCGs
29	0x1	_NONE_	**SD_CARD_DETECT_STATUS_N**: SW should read CD_N status from external PMIC/VGPIO controller and update this field to get SDMMC present state register gets updated. Present state register is read by standard SDHC driver to know card status. 0 - card detected 1 - no card present in slot
28	0x0	_NONE_	**SD_CARD_WP_STATUS**: SW should read Write Protect status from external PMIC/VGPIO controller and update this field to get SDMMC present state register gets updated. Present state register is read by standard SDHC driver to know card status. 0 - card is not write protected 1 - card is write protected
27	0x0	_NONE_	**CMD_TFIFO_HOT_RESET**: SW can reset CMD tuning FIFO present in padmacro incase there is any error or hang condition. Reset duration should be at least 20 cycles. Set this bit to 1 and clear it after 20 cycles to reset FIFO.
26	0x0	_NONE_	**DAT_TFIFO_HOT_RESET**: SW can reset DAT tuning FIFO present in padmacro incase there is any error or hang condition. Reset duration should be at least 20 cycles. Set this bit to 1 and clear it after 20 cycles to reset FIFO.
25	0x1	NORMAL	**ADMA3_CLKEN_OVERRIDE**: Override for sdmmc_adma3_g_clk clken; **0** = NORMAL :0 -> sdmmc_adma3_g_clk is gated in nonADMA3 modes **1** = OVERRIDE :1 -> sdmmc_adma3_g_clk is not gated in nonADMA3 modes
24	0x1	NORMAL	**CQE_CLKEN_OVERRIDE**: Override for sdmmc_cqe_g_clk clken; **0** = NORMAL :0 -> sdmmc_cqe_g_clk is gated in nonCQE modes **1** = OVERRIDE :1 -> sdmmc_cqe_g_clk is not gated in nonCQE modes
23	0x1	_NONE_	**CQE_DESC_PREFETCH_EN**: When set to 1, enables CQE task descriptors pre-fetch feature. It helps improving perf when system memory is heavily fragmented (number of descriptors programmed per transfer will be more)

Bit	Reset	PROD	Description
22	0x1	_NONE_	**ADMA3_DESC_PREFETCH_EN**: When set to 1, enables ADMA3 descriptors pre-fetch feature. It helps improving perf when system memory is heavily fragmented (number of descriptors programmed per transfer will be more)
21	0x1	_NONE_	**ADMA2_DESC_PREFETCH_EN**: When set to 1, enables ADMA2 descriptors pre-fetch feature. It helps improving perf when system memory is heavily fragmented (number of descriptors programmed per transfer will be more)
19:16	0x0	_NONE_	**DATA_TIMEOUT_VAL_MULTIPLIER**: Used when SDMMC IO clock is used instead TMCLK for running data timeout counter (USE_TMCLK_FOR_DATA_TIMEOUT is set in VENDOR_SYS_SW_CNTRL register). Effective data timeout val = (multiplier+1) * data_timeout_val 0 - no multiplier
15:12	0x0	_NONE_	**DAT_TUNING_ASYNC_FIFO_ADDNL_DELAY**: This register field holds the additional delay in cycles which should be added to wdata/crc token round trip delay Default value is - zero. NOTE: SW should not update this field unless a new PROD setting is given.
11:8	0x0	_NONE_	**ADDITIONAL_NCR_CYCLES**: Additional Ncr wait time - useful for HW debug Default is 0 - SW should not modify this.
7:0	0x0	_NONE_	**OVERRIDE_FOR_1_8V_VDD2**: Maximum override for 1.8V VDD2

SDMMCAB_VENDOR_IO_TRIM_CNTRL_0

Vendor IO trimmer control register
Used to configure IO trimmer
Truth table
Input Pins Output Comments
E_DPD SEL_VREG SEL_VREF CLKOUT
1 x x 0 The cell is in deep power down mode
0 1 x based on ip_clk_select selected clock input Trimmer is powered by VAUXC
0 0 x based on ip_clk_select selected clock input Trimmer is powered by regulated voltage
* 'x' indicates don't care

Offset: 0x1ac
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000205 (0b0000,xxxx,xxxx,xxxx,xxxx,x010,000x,x101)

Bit	Reset	Description
31:28	0x0	**TRIM_PAD_RFU_IN**: Unused comp pad input pins. Reserved
10:8	VREF_750_MV	**SEL_VREF_LEVEL**: Selects Vref voltage level **0** = VREF_700_MV **1** = VREF_725_MV **2** = VREF_750_MV **3** = VREF_775_MV **4** = VREF_800_MV **5** = VREF_850_MV **6** = VREF_900_MV **7** = VREF_950_MV
7:6	0x0	**TRIM_SEL_ATEST**: For testing purpose only - should be used when sdmmc is in idle state Select analog test signals to send to comp pad. 0x0: Not used (float) 0x1: Regulator input voltage 0x2: Regulator output voltage before analog mux 0x3: Regulator output after analog mux
5	ENABLE	**TRIM_PWRSAVE**: Enables power saving mode by clock gating the unused taps in delay chain Active low signal, 0 - power saving mode enabled - clock gating is enabled for unused trimmer taps - may affect tap delay 1 - no power saving - all the trimmer taps are not clock gated **0** = ENABLE **1** = DISABLE

Bit	Reset	Description
2	0x1	**SEL_VREG**: By default, BG is disabled to save power if interface is not used. SW should select BG for error free SD/eMMC operation. For BG <-> VAUXC switching, SW should follow the switching sequence given in TRM/IAS. Select voltage supply for delay chain present in both Trimmer and DLLs PROD value: 0x0 ***SW should set this to 0x0 before accessing SD/eMMC. This setting makes IB trimmer delay independent of VDD_CORE*** 0 - selects regulated reference voltage for trimmer supply - default (recommended option for tunable SD/eMMC modes) 1 - selects VAUXC for trimmer supply and shut down BG+REG circuit (can be used in non-Tunable modes for power saving) Power up time for BG+REG is ~3us(worst case). Power down time for BG+REG is ~1us(worst case). If SW wants to turn on/off BG+REG when SDMMC is idle, it has to take hit of 3us power on time. When SEL_VREG is toggled, both DLL and rx clock trimmer output could glitch irrespective of input clock state which could cause corresponding rx CMD and DATA FIFOs to go into bad state. SW should issue SW_RESET_DAT and SW_RESET_CMD to reset host FIFOs after BG <-> VAUXC switching. This would ensure error free data transfers from there on. Powering down BG would need 3usec turn ON time which may cause IOPS reduction, if SW shut downs BG after every transfer and enables it on seeing new xfer req. Hence, it may not be possible to do dynamic shut down of BG without stalling new requests.
1	0x0	**SEL_VREF**: Select reference voltage for voltage regulator 0 - selects Bandgap Voltage Reference (recommended option for SD/eMMC tunable modes) 1 - selects resistor divider voltage reference and power down bandgap Switching time between the supplies is 1us. When switching from one supply to other supply, we need to wait for at least 1 us before doing any data transfers. Providing reference voltage from R divider network is just a backup plan, if A. Bandgap does not work or B. Bandgap works very well but we want to save bandgap power when Silicon Characterization results shows that the eMMC/ SDMMC interface perform well even by using R divider+REG+TRIMMER

Bit	Reset	Description
0	0x1	**PD_BGREG**: Not used - Dummy control Power down Band Gap voltage reference, voltage regulator and resistor chain voltage ref (BG+REG) present in custom IO trimmer used for SD/eMMC bus tuning. Active High signal, PD_BGREG=1 => Power down BG+REG; PD_BGREG=0 => power up Power down and up time for BG+REG is 1us. If SW wants to turn on/off BG+REG when SDMMC is idle, it has to follow 1us power on/off time. Back-up option for powering down BG. SW should clear this bit when it wants to turn ON BG by setting SEL_VREG=0. The original idea to have PD_BGREG pin is to provide power saving feature when the eMMC/SDMMC is in IDLE state, but not in DPD mode. This pin function is actually merged into SEL_VREG function -BG+REG circuit is shut-down when SEL_VREG=1 (trimmer powered by VAUXC)

SDMMCAB_VENDOR_DLLCAL_CFG_0

Vendor DLL calibration configuration register

This register is used to setup the DLL Calibration settings.
SW has to run DLL calibration first before initiating data transfers in HS400 mode and re-calibration is required when ever there is a change
in emmc device clock frequemcy in HS400 mode.

Offset: 0x1b0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x16083504 (0b0x01,0110,0000,1000,0011,0101,0000,0100)

Bit	Reset	Description
31	0x0	**CALIBRATE**: CALIBRATE is used to start a DLL calibration process. SW should set this bit to trigger calibration and should not clear this bit. This bit will be cleared once the calibration process is completed.
29:25	0xb	**END_COUNT**: END_COUNT determines the end condition of the calibration. If USE_STATIC_CYCLES is not set, a usec timer is loaded with (2^END_COUNT) and calibration will be stopped once the timer expires. Recommended timer value is 1msec. If USE_STATIC_CYCLES is set and if the trimmer tap value being calibrated has not changed for this number of loops, the calibration ends. The number iterations is (2 ^ END_COUNT), if END_COUNT != 0

Bit	Reset	Description
24	0x0	**USE_STATIC_CYCLES**: When set, calibration will be stopped when num static cycles reaches END_COUNT else u sec timer is used. Recommended option is use u sec timer.
23	0x0	**IGNORE_START_TRIM**: IGNORE_START is used to ignore the START_TRM value. Instead the calibration starts at the current programmed DDLLCAL value. This is intended to be used if we ever support periodic DDLL calibration.
22:16	0x8	**START_TRIM**: START_TRM specifies the starting trimmer value to calibrate the with.
15:12	0x3	**FILTER_BITS**: FILTER_BITS is the LSB of the counter to use for updating the trimmer value. The new trimmer value is trim_old + (count >> FILTER_BITS) - (trim_old >> FILTER_BITS) FILTER_BITS chosen < bits used for SAMPLE_COUNT
11:8	0x5	**SAMPLE_COUNT**: SAMPLE_COUNT is the number of times the phase detector is sampled before going onto the next set of trimmer values. The number of samples is (2 ^ SAMPLE_COUNT)
7:0	0x4	**SAMPLE_DELAY**: SAMPLE_DELAY is the number of sdmmc host clks from changing the trimmer value to when the phase detector is sampled.

SDMMCAB_VENDOR_DLL_CTRL0_0

Vendor DLL control register0

This register has SW overrides for controlling both master and slave DLL inputs.

Offset: 0x1b4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x60000000 (0b0110,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	**MST_DLL_CLK_EN_OVERRIDE**: Master DLL CLK_IN enable override **0** = NORMAL :0 -> Mst DLL clk_in is gated when dll cal is not running **1** = OVERRIDE :1 -> Mst DLL clk_in is not gated irrespective of dll cal status

Bit	Reset	Description
30	0x1	**TX_SLV_DLL_PWRSAVE**: PWRSAVE is active low signal 1: no clock gating enabled for TX slave DLL 0: clock gating is enabled for unused delay taps in TX slv DLL to save power This control does not affect the logic output of slave DLL ; controls slv dll enable in iobrick
29	0x1	**RX_SLV_DLL_PWRSAVE**: PWRSAVE is active low signal 1: no clock gating is enabled for RX slave DLL 0: clock gating is enabled for unused delay taps in RX slv DLL to save power This control does not affect the logic output of slave DLL ; controls slv dll enable in iobrick
28:22	0x0	**TX_SLV_DLL_DLY_CODE**: 7-bit delay code (128 taps) to be applied to TX slave DLL when DLLCAL_BYPASS is enabled
21:15	0x0	**RX_SLV_DLL_DLY_CODE**: 7-bit delay code (128 taps) to be applied to RX slave DLL when DLLCAL_BYPASS is enabled
14	DISABLED	**DLLCAL_BYPASS**: DLL calibration bypass enable **0** = DISABLED :1 : Programmed delay code will be applied to both master and slave DLLs; **1** = ENABLED :0 : Delay code determined by DLL calibration is applied to master and slave DLLS
13:7	0x0	**TX_DLY_CODE_OFFSET**: two-s complement offset will be added to delay code generated by calibration controller and sent to TX slv DLL
6:0	0x0	**RX_DLY_CODE_OFFSET**: two-s complement offset will be added to delay code generated by calibration controller and sent to RX slv DLL

SDMMCAB_VENDOR_DLL_CTRL1_0

Vendor DLL control register1

This register has SW overrides for controlling both master and slave DLL inputs.

Offset: 0x1b8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x20208780 (0bx010,0000,x010,0000,1000,0111,1000,0000)

Bit	Reset	Description
30:24	0x20	**REQD_DELAY_STEPS_TX**: Delay required in steps of 1/64UI - 0 to 63 steps MST DLL is half cycle locked - UI=0.5cycle=64steps Default value is quarter cycle. Calibrated code covers oneUI of mst DLL. This reg field value is used to scale the delay code before sending it to TX slv DLL.
22:16	0x20	**REQD_DELAY_STEPS_RX**: Delay required in steps of 1/64UI - 0 to 63 steps MST DLL is half cycle locked - UI=0.5cycle=64steps Default value is quarter cycle. Calibrated code covers oneUI of mst DLL. This reg field value is used to scale the delay code before sending it to RX slv DLL.
15:11	0x10	**MST_DLL_RESET_TIME**: Master DLL reset duration - 16cycles
10:7	0xf	**SLV_DLL_SETTLE_TIME**: Slave DLL requires 4 cycles settle time when dly code is changed for providing stable output
6:0	0x0	**MST_DLL_DLY_CODE**: 7-bit delay code (128 taps) to be applied to master DLL when DLLCAL_BYPASS is enabled

SDMMCAB_VENDOR_DLLCAL_CFG_STA_0

Vendor DLL calibration config and status register

Offset: 0x1bc
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x10000000 (0b0001,0000,x000,0000,x000,0000,x000,0000)

Bit	R/W	Reset	Description
31	RO	DONE	**DLL_CAL_ACTIVE**: Calibration process status SW has to wait for DONE once calibrate bit is cleared and before starting data transfers on eMMC interface. **0** = DONE :0 : DLL calibration is completed - calibration engine is idle and slave DLLs are ready for normal operation **1** = RUNNING :1 : DLL calibration is still running
30	RO	0x0	**DLL_PD**: Master DLL Phase detector output

Bit	R/W	Reset	Description
29	RW	0x0	**MST_DLL_RST_OVERRIDE_EN**: When set, master dll reset is controlled by programming MST_DLL_RST_ Else reset is generated by DLL controller
28	RW	0x1	**MST_DLL_RST_**: Master DLL reset - active low signal - used when DLL_RST_OVERRIDE_EN is set minimum reset duration is 16 sdhost clk cycles
27	RW	0x0	**SLV_DLL_CLK_OUT_DIS_OVERRIDE_EN**: When set, slave dll clk_out_dis is controlled by programming SLV_DLL_CLK_OUT_DIS Else clk_out_dis is controlled by DLL controller
26	RW	0x0	**SLV_DLL_CLK_OUT_DIS**: Slave DLL clock out disable - used when SLV_DLL_CLK_OUT_DIS_OVERRIDE_EN is set 0 : Delayline clock normal output 1 : DelayLIne clock output gated and grounded to 0. should be set to 1 when SW trys to update slv dll delay code should be cleared after slave dll settle time - 3cycles
25	RW	0x0	**MST_DLL_PWRDN_OVERRIDE_EN**: When set, master dll can be kept in power down mode by programming MST_DLL_PWRDN field. Else pwrdn is controlled by DLL controller
24	RW	0x0	**MST_DLL_PWRDN**: Master DLL power down enable - active high control - used when MST_DLL_PWRDN_OVERRIDE_EN is set 0: power up ; 1 - power down SW can keep MST dll in power down mode once calibration is done. MST DLL should be powered up before triggering calibration. Controls dll_en in iobrick
22:16	RO	0x0	**TX_DLY_CODE_ADJ**: delay code sent to TX slave DLL after applying offset; Valid only when dll auto-calibration is used. (calib_dly_code * reqd_dly_steps/64) + offset
14:8	RO	0x0	**RX_DLY_CODE_ADJ**: delay code sent to RX slave DLL after applying offset; Valid only when dll auto-calibration is used. (calib_dly_code * reqd_dly_steps/64) + offset
6:0	RO	0x0	**DLY_CODE**: Holds delay code determined by calibration process which is sent to MST dll during calibration; Valid only when DLL_CAL_ACTIVE is set

SDMMCAB_VENDOR_TUNING_CNTRL0_0

Vendor Tuning Control0 register

Offset: 0x1c0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x74020090 (0bx111,0100,0000,001x,0000,0000,1001,0000)
PROD: 0x00000040 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,01xx,xxxx)

Bit	Reset	PROD	Description
30	0x1	_NONE_	**RD_DATA_CRC_ERR_EN**: If cleared, Re-tuning request/tuning error is not generated on read data crc error
29	0x1	_NONE_	**WR_DATA_CRC_ERR_EN**: If cleared, Re-tuning request/tuning error is not generated on write crc error
28	0x1	_NONE_	**CMD_CRC_ERR_EN**: If cleared, Re-tuning request/tuning error is not generated on cmd crc error
27	0x0	_NONE_	**RETUNING_REQ_EN_ON_CRC_ERR_DETECTION**: Re-tuning request is generated when set to initiate re-tuning by SW to compensate for temperature drift when any data or cmd CRC errors are detected in SDR50/SDR104/HS200 modes
26	0x1	_NONE_	**TUNING_ERR_EN_ON_CRC_ERR_DETECTION**: Tuning error is generated to initiate re-tuning by SW to compensate for temperature drift when any data or cmd CRC errors are detected in SDR50/SDR104/HS200 modes
25:18	0x0	_NONE_	**START_TAP_VAL**: start tap value to be used by tuning; start_tap should be multiple of step_size chosen and its valid range is 0-255; Not valid when TAP_VAL_UPDATED_BY_HW is set to zero. Tuning algorithm uses this as the start tap value for scanning through trimmer taps.

Bit	Reset	PROD	Description
17	0x1	_NONE_	**TAP_VAL_UPDATED_BY_HW**: This bit is functional only in tunable modes (SDR50, SDR104 and HS200) SW can choose to update the tap val by itself by clearing this bit; Preferred value is 1 - tap val is updated by HW. If this bit is cleared, HW does not update tap_val per every tuning iteration. SW can update it as desired. Tuning pattern match is indicated by sampling_clock_select per every tuning iteration. SW has to maintain the status of each tuning iteration and determine the best PASS window to fix the final sampling point. And SW can program NUM_TUNING_ITERATIONS as desired. Once the number of tuning commands issued reaches number of tuning iterations programmmed, execute_tuning bit will be cleared to indicate the completion of tuning procedure. Please note that using this option violates Host Spec but provided for legacy reasons. It helps us in using legacy SW tuning solution incase HW solution does not work.
15:13	TRIES_40	_NONE_	**NUM_TUNING_ITERATIONS**: The number of tuning iterations to be used by tuning circuit. **0** = TRIES_40 **1** = TRIES_64 **2** = TRIES_128 **3** = TRIES_192 **4** = TRIES_256
12:6	0x2	0x1	**MUL_M**: implements a multiplier - M+1 Final tap value is derived from best passing window and calculated as follows. Final tap value = first_pass + ((last_pass - first_pass)*Q); where Q = percentage of pass window;default-75% Q = M+1/(2^N); N:1...7 M:should be in range [0:2^N-1];
5:3	0x2	_NONE_	**DIV_N**: implements a divider - 2^N; max div is 2^7 =>128
2:0	0x0	_NONE_	**TUNING_WORD_SEL**: Selects desired word from 256-bit tuning status bitmap status_word[31:0] = status[255:0] >> (tuning_word_sel * 32)

SDMMCAB_VENDOR_TUNING_CNTRL1_0

Vendor Tuning Control1 register
Different step size is required in SDR50 mode to cover two UI (100 MHz => 2*10 ns)
With 70ps/tap trimmer resolution, we can cover almost 2UI using step_size=8 in SDR50 and step_size=4 in SDR104.

Tuning will be done in HS200 - SDR mode only.
HS200 - tuning @200 MHz - UHS_SEL should be SDR104 for executing tuning
Before initiating data transfers in HS400 mode, tuning procedure should be executed in HS200 mode with IO clock running @200MHz
DQ_OFFSET: offset between even and odd bits. Td is per-tap delay in trimmer. When the DQ offset function is turned off (DQ_OFFSET[1:0]=00), there is no offset between ZI0~7. When the DQ offset function is turned on
(DQ_OFFSET[1:0]=01, 10, 11),
extra delays are added at the odd bits (DQ1, 3, 5,7). Thus, there is an offset between even bits and odd bits. DQ offset function is turned on during auto-tuning to avoid the window-merged issue and turned off during normal operation.
00 no offset
01 1*Td
10 2*Td
11 3*Td

Offset: 0x1c4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000400 (0b00xx,xxxx,xxxx,xx00,0000,0100,x000,x000)

Bit	Reset	Description
31:30	0x0	**DQ_OFFSET**: offset between even and odd bits
17	0x0	**FIRST_PASS_WINDOW_SEL**: This enables the selection of the first pass window by the HW tuning engine. First pass window select feature is enabled only when STEP_SIZE is set to 0.
16	0x0	**FALSE_PASS_MASK**: This enables masking of the false pass windows from the tap value selection. False pass mask feature is enabled only when STEP_SIZE is set to 0.
15:8	0x4	**MIN_PASS_WINDOW_WIDTH**: This is used to mask false passes. Tuning engine considers PASS windows of size > MIN_WIDTH for tap value calculation. Allowed range is 1<= MIN_PASS_WINDOW_WIDTH <=4.
6:4	0x0	**STEP_SIZE_SDR104_HS200**: tap_val is incremented by step_size for every tuning iteration - used in SDR104/HS200/HS400 mode increment = 2^step_size; step_size should be in range 0-4. Others are RSVD
2:0	0x0	**STEP_SIZE_SDR50**: tap_val is incremented by step_size for every tuning iteration - used in SDR50 mode increment = 2^step_size; step_size should be in range 0-4. Others are RSVD

SDMMCAB_VENDOR_TUNING_STATUS0_0

Vendor Tuning Status0 register

Offset: 0x1c8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**STATUS_WORD**: Each bit indicates the status of each tuning iteration, when tap value is updated by HW (TAP_VAL_UPDATED_BY_HW=1); 0-Tuning pattern not matched 1-tuning pattern matched We have a total of 256 tap values. SW can issue a max. of 256 tuning commands for debug. SW needs to read this register eight times to get status of all 256 iterations by changing tuning_word_sel status[255:0] is left shifted and loaded into this register every time when tuning_word_sel is changed status_word[31:0] = status[255:0] >> (tuning_word_sel * 32)

SDMMCAB_VENDOR_TUNING_STATUS1_0

Vendor Tuning Status1 register

Offset: 0x1cc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**PASS_WINDOW_END_BEFORE_FINE_TUNING**: End tap value of best PASS window found by scan FSM
23:16	0x0	**PASS_WINDOW_START_BEFORE_FINE_TUNING**: Start tap value of best PASS window found by scan FSM
15:8	0x0	**PASS_WINDOW_END_AFTER_FINE_TUNING**: End tap value of best PASS window after fine tuning
7:0	0x0	**PASS_WINDOW_START_AFTER_FINE_TUNING**: Start tap value of best PASS window after fine tuning

SDMMCAB_VENDOR_CLK_GATE_HYSTERESIS_COUNT_0

Vendor Clk gating Hysteresis Counter initial value

Offset: 0x1d0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,1111)

Bit	Reset	Description
5:0	0xf	**CLK_COUNT**: Before gating second level clocks controller will wait for these many cycles. we can recover if any idle windows are missed in clken equation

SDMMCAB_VENDOR_PRESET_VAL0_0

Vendor Preset Value Registers

SD host spec defines one preset value register for each bus speed mode which should be set by host by some unique method.
Preset values vary based on the base frequency used which is in SW (SoC system driver) control.
System driver supposed to set BASE_CLK_FREQ in VENDOR_CLOCK_CNTRL register before handling over the control to SD host standard driver.
In the similar way, system driver should set below vendor preset values based on the base clock frequency and the desired card clock frequency in each
bus speed mode
This should be done after every time SDMMC is reset and after every soft reset.
This is important as all SDMMC controllers follow the same register map, but could
be programmed with different frequencies depending on the use case.
Default values are set assuming base clock frequency=208 MHz.

Offset: 0x1d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00201000 (0bxx00,0000,0010,0000,0001,0000,0000,0000)

Bit	Reset	Description
29:20	0x2	**SDCLK_FREQ_SEL_HIGH_SPEED**: System software programs 10-bit divider value to generate SD clk in default speed mode (<50MHz,3.3Vsignaling) This value is readable in the standard via PRESET_SDR12_AND_HIGH_0_SDCLK_FREQ_VAL_LOW register field Default val is 0x2 assuming 208MHz base clock
19:10	0x4	**SDCLK_FREQ_SEL_DEFAULT**: System software programs 10-bit divider value to generate SD clk in default speed mode (<25MHz,3.3Vsignaling) This value is readable in the standard via PRESET_DEFAULT_AND_INIT_0_SDCLK_FREQ_VAL_HIGH register field Default val is 0x4 assuming 208MHz base clock
9:0	0x0	**SDCLK_FREQ_SEL_INIT**: System software programs 10-bit divider value to generate desired SD clk frequency during initialization This value is readable in the standard via PRESET_DEFAULT_AND_INIT_0_SDCLK_FREQ_VAL_LOW register field For Eg., if 400KHz SDCLK is desired @base clk freq=48MHz, this register should be programmed with 0x3C

SDMMCAB_VENDOR_PRESET_VAL1_0

Offset: 0x1d8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00100804 (0bxx00,0000,0001,0000,0000,1000,0000,0100)

Bit	Reset	Description
29:20	0x1	**SDCLK_FREQ_SEL_SDR50**: System software programs 10-bit divider value to generate SD clk in SDR50 mode (<100MHz,1.8Vsignaling) This value is readable in the standard via PRESET_SDR50_AND_SDR25_0_SDCLK_FREQ_VAL_HIGH register field Default val is 0x1 (gives 2N divider) assuming 208MHz base clock
19:10	0x2	**SDCLK_FREQ_SEL_SDR25**: System software programs 10-bit divider value to generate SD clk in SDR25 mode (<50MHz,1.8Vsignaling) This value is readable in the standard via PRESET_SDR50_AND_SDR25_0_SDCLK_FREQ_VAL_LOW register field Default val is 0x2 assuming 208MHz base clock
9:0	0x4	**SDCLK_FREQ_SEL_SDR12**: System software programs 10-bit divider value to generate SD clk in SDR12 mode (<25MHz,1.8Vsignaling) This value is readable in the standard via PRESET_SDR12_AND_HIGH_0_SDCLK_FREQ_VAL_HIGH register field

SDMMCAB_VENDOR_PRESET_VAL2_0

Offset: 0x1dc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000800 (0bxxxx,xxxx,xxxx,0000,0000,1000,0000,0000)

Bit	Reset	Description
19:10	0x2	**SDCLK_FREQ_SEL_DDR50**: System software programs 10-bit divider value to generate SD clk in DDR50 mode (<50MHz,1.8Vsignaling) This value is readable in the standard via PRESET_DDR50_AND_SDR104_0_SDCLK_FREQ_VAL_HIGH register field Default val is 0x2 assuming 208MHz base clock
9:0	0x0	**SDCLK_FREQ_SEL_SDR104**: System software programs 10-bit divider value to generate SD clk in SDR104 mode (<208MHz,1.8Vsignaling) This value is readable in the standard via PRESET_DDR50_AND_SDR104_0_SDCLK_FREQ_VAL_LOW register field

SDMMCAB_SDMEMCOMPPADCTRL_0

SDMEMCOMP Pad control register

This register is used to control COMP pad inputs.

Offset: 0x1e0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x08000000 (0b00x0,1xx0,0000,xxx0,0000,0000,0000,0000)
PROD: 0x00a0a000 (0bxxxx,xxx0,1010,xxx0,1010,xxxx,xxxx,xxxx)

Bit	Reset	PROD	Description
31	0x0	_NONE_	**PAD_E_INPUT_OR_E_PWRD**: used to control E_INPUT(for SDMMC1/3) and E_PWRD (for SDMMC4) input of pu/pd comp pad should be set at least 1usec before starting auto-cal and cleared once auto-calibration is done (for power saving) NOTE: E_PWRD = !PAD_E_INPUT_OR_E_PWRD and E_INPUT = PAD_E_INPUT_OR_E_PWRD

Bit	Reset	PROD	Description
30	0x0	_NONE_	**COMP_PAD_E_PBIAS_BUF**: Active high. Enables internally generated bias levels for driver PMOS. We dont use this feature for SDMMC pads. SW *should not* set this bit. Keeping reg for debug purpose.
28:27	0x1	_NONE_	**COMP_PAD_DRV_TYPE**: used to control drv_type input of BDSDMEMLVCOMP_C pad
24:20	0x0	0xa	**COMP_PAD_DRVUP_OVR**: used to drive DRVUP input of COMP pad if AUTO_CAL_ENABLE is disabled
16:12	0x0	0xa	**COMP_PAD_DRVDN_OVR**: used to drive DRVDN input of COMP pad if AUTO_CAL_ENABLE is disabled
11	0x0	_NONE_	**COMP_PAD_E_TEST_OUT**: used to control e_test_out input of COMP pad
10:7	0x0	_NONE_	**COMP_PAD_RFU_IN**: Unused comp pad input pins. Reserved
6:4	0x0	_NONE_	**COMP_PAD_TEST_SEL**: used to control test_sel input of COMP pad
3:0	0x0	_NONE_	**SDMMC2TMC_CFG_SDMEMCOMP_VREF_SEL**: Select different bias levels for driver PMOS when E_PBIAS_BUF=1. We dont use this feature for SDMMC pads. SW *should not* set this bit. Keeping reg for debug purpose.

SDMMCAB_AUTO_CAL_CONFIG_0

SDMEMCOMP pad auto-calibration settings
AUTO_CAL_SLW_OVERRIDE
0 (Normal operation) pad DRVDN/UP_SLWR/F tied to AUTO_CAL output
DRDVDN/UP_SLWR/F[1:0] = AUTO_CAL_PULLDOWN/UP[4:3]
1 (override) use CFG2TMC_SDIO[1|3]*_DRVDN/UP_SLWR/F pins to control pad slew inputs
AUTO_CAL_OVERRIDE
0 (normal operation): use AUTO_CAL_PU/PD_OFFSET as an offset to the calibration state machine setting
1 (override) : use AUTO_CAL_PU/PD_OFFSET register values directly

Offset: 0x1e4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0

Reset: 0x00010000 (0b0000,xxxx,xxxx,x001,xxx0,0000,xxx0,0000)
PROD: 0x20000000 (0bxx1x,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	PROD	Description
31	0x0	_NONE_	**AUTO_CAL_START**: Writing a one to this bit starts the calibration state machine. This bit must be set even if the override is set in order to latch in the override value.
30	0x0	_NONE_	**AUTO_CAL_OVERRIDE**: AUTOCAL override. **0** = NORMAL :0 (normal operation): use AUTO_CAL_PU/PD_OFFSET as an offset to the calibration state machine setting **1** = OVERRIDE :1 (override) : use AUTO_CAL_PU/PD_OFFSET register values directly
29	DISABLED	ENABLED	**AUTO_CAL_ENABLE**: AUTOCAL enable. **0** = DISABLED :0 (disabled): use sdmmc2tmc_cfg* register settings for pullup/dn **1** = ENABLED :1 (normal operation): use SDMMC generated pullup/dn (override or AUTOCAL)
28	0x0	_NONE_	**AUTO_CAL_SLW_OVERRIDE**: AUTOCAL slew rate override **0** = NORMAL :0 (Normal operation) pad DRVDN/UP_SLWR/F tied to AUTO_CAL output DRDVDN/UP_SLWR/F[1:0] = AUTO_CAL_PULLDOWN/UP[4:3] **1** = OVERRIDE :1 (override) use CFG2TMC_SDIO[1|3]*_DRVDN/UP_SLWR/F pins to control pad slew inputs
18:16	0x1	_NONE_	**AUTO_CAL_STEP**: calibration step interval (in microseconds)
12:8	0x0	_NONE_	**AUTO_CAL_PD_OFFSET**: 2's complement offset for pull-down value
4:0	0x0	_NONE_	**AUTO_CAL_PU_OFFSET**: 2's complement offset for pull-up value

SDMMCAB_AUTO_CAL_INTERVAL_0

SDMEMCOMP pad calibration interval

Offset: 0x1e8
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**AUTO_CAL_INTERVAL**: 0: do calibration once Otherwise, auto-calibration occurs at intervals equivalent to the programmed number of microseconds.

SDMMCAB_AUTO_CAL_STATUS_0

SDMEMCOMP pad calibration status

Offset: 0x1ec
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0xx0,0000,xxx0,0000,xxx0,0000,xxx0,0000)

Bit	Reset	Description
31	0x0	**AUTO_CAL_ACTIVE**: One when auto calibrate is active - valid only after auto calibrate sequence has completed (AUTO_CAL_ACTIVE == 0)
28:24	0x0	**AUTO_CAL_PULLDOWN_ADJ**: Pulldown code sent to pads
20:16	0x0	**AUTO_CAL_PULLUP_ADJ**: Pullup code sent to pads
12:8	0x0	**AUTO_CAL_PULLDOWN**: Pulldown code generated by auto-calibration
4:0	0x0	**AUTO_CAL_PULLUP**: Pullup code generated by auto-calibration

SDMMCAB_IO_SPARE_0

These SPARE_OUT bits go to pipe -> pad and then come back as SPARE_IN
SPARE_OUT[3] : IO_SPARE[19] - used as MUX select which selects between one cycle delay and two cycle delay versions of cmd_oen to mask wdata of IB
capture flop.
0x0 : selects zero cycle delayed version
0x1 : selects one cycle delayed version - recommended
SPARE_OUT[2] : IO_SPARE[18] - used as active low enable for gating both CMD_IN and DAT_IN

aysnc FIFOs wdata when we are driving CMD/DAT lines.
0x0 : write 1 when OEN is active and Zi value when OEN is not active into FIFO (default)
0x1 : write Zi value into FIFO irrespective of OEN state.

Offset: 0x1f0
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0008XXXX (0b0000,0000,0000,1000,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31:16	RW	0x8	**SPARE_OUT**
15:0	RO	X	**SPARE_IN**

SDMMCAB_CIF2AXI_CTRL_0

SDMMC CIF2AXI control register

DMA transaction (MC transaction) attributes

Offset: 0x1fc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:8	0x0	**MC_WRITE_REQ_STREAM_ID**: MC write transaction stream ID
7:0	0x0	**MC_READ_REQ_STREAM_ID**: MC read transaction stream ID

SDMMCAB_TZ_DMA_CTRL_0

SDMMC DMA requests security attribute control register for DMA transaction (MC transaction) security attributes.
This register is used to control write/read access to secure memory region.
This register can be accessed only by TZ. Non-secure writes to this register are dropped by

controller and reads return all ones.
Controller will assert PSLVERR when this register is accessed by a non-secure master.
Usage:
wsb_ns = AWPROT[1] = ~{SDMMCxx_TZ_DMA_CTRL_MC_WRITE_REQ_TZ_ACCESS_EN};
rsb_ns = ARPROT[1] = ~{SDMMCxx_TZ_DMA_CTRL_MC_READ_REQ_TZ_ACCESS_EN};

Offset: 0x200
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**MC_WRITE_REQ_TZ_ACCESS_EN**: TZ must set this bit to 1 to enable write access to secure memory region. TZ must clear this bit to disable write access to secure memory region.
0	0x0	**MC_READ_REQ_TZ_ACCESS_EN**: TZ must set this bit to 1 to enable read access to secure memory region. TZ must clear this bit to disable read access to secure memory region.

SDMMCAB_VENDOR_MISC_CNTRL3_0

Offset: 0x204
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x4f01000f (0b0100,1111,xxxx,xxx1,0000,0000,xxx0,1111)

Bit	Reset	Description
31	0x0	**ALLOW_INTRABLOCK_CLK_STALLING_IN_SDR50**: This disables the intrablock clock stopping in SDR50 mode
30	0x1	**STOP_TRIM_IN_CLK_DURING_TUNING**: When set to 1, rx clk trimmer and rx fifos input clock is stopped when tap value is changed during tuning process. When set to 0, rx clk trimmer and rx fifos input clock is not stopped during tap value change. Trimmer output is also clamped to zero during tuning tap val change.

Bit	Reset	Description
29:24	0xf	**TUNING_TRIMMER_RECOVERY_TIME**: Trimmer output may glitch for 2 or 3 cycles, when tap value is changed irrespective of its input clock state. We should not use trimmer output during this uncertainty window. If STOP_TRIM_IN_CLK_DURING_TUNING is set to 1, - trimmer input clock is stopped for TUNING_TRIMMER_RECOVERY_TIME cycles and - rx fifos input clock is stopped (trimmer output clock is clamped) for TUNING_TRIMMER_RECOVERY_TIME cycles. This is required not to propagate glitch into FIFOs and other downstream logic.
16	0x1	**EXTEND_SYNC_INTR_MASK_DURING_ABORT_OR_STOP_CMD**: When an async abort is issued during read operation, dat_fsm will move to idle state as soon as CMD12/CMD52 END bit is sent by core. This would start sync_intr_period. But due to the intermediate delay stages present in pipemacro and padmacro, END bit reaches device after some cycles. So, device would not stop data transmission till it sees END bit of ABORT CMD. During this time, core controller receives the data already driven by the device. If DAT[1] line has any zeroes during this period, SDIO sync interrupt detector will generate a spurious card interrupt. This register bit is used to mask sync intr detection period to avoid spurious interrupts. SW should not write into this field unless it is published in TRM/IAS.
15:8	0x0	**E_DIFF_DQ**: diff/Vref rx selection for DAT[7:0] If set to 1, enables differential amplitude receiver for DAT lines in EMMC IOBRICK. If set to 0, enables vref receiver for DAT lines in EMMC IOBRICK. (default)
4	0x0	**E_DIFF_CMD**: diff/Vref rx selection for CMD If set to 1, enables differential amplitude receiver for CMD in EMMC IOBRICK. If set to 0, enables vref receiver for CMD in EMMC IOBRICK. (default)
3	0x1	**DAT_OE_POSTAMBLE_EN**: If set, DAT pads output driver will be disabled one cycle after END bit of DATA pkt sent. If cleared, DAT pads output driver will be disabled in the same cycle END bit is sent.
2	0x1	**DAT_OE_PREAMBLE_EN**: If set, DAT pads output driver will be enabled one cycle before START bit of DATA pkt is transmitted on DAT lines. If cleared, DAT pads output driver will be enabled in the same cycle START bit is sent.
1	0x1	**CMD_OE_POSTAMBLE_EN**: If set, CMD pad output driver will be disabled one cycle after END bit of CMD pkt sent. If cleared, CMD pad output driver will be disabled in the same cycle END bit is sent.
0	0x1	**CMD_OE_PREAMBLE_EN**: If set, CMD pad output driver will be enabled one cycle before START bit of CMD pkt is transmitted on DAT lines. If cleared, CMD pad output driver will be enabled in the same cycle START bit is sent.

SDMMCAB_VENDOR_CQE_CNTRL0_0

Offset: 0x208
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xe0000c82 (0b1110,xxxx,xxxx,xxxx,xx00,1100,1000,0010)

Bit	Reset	Description
31	0x1	**TASK_SELECTION_PRIORITY_EN**: If set to 1, CQE host considers high priority attribute set for a queued task while selecting a task for execution.
30	0x1	**MASK_CMD_COMPLETE_INTR**: CMD complete interrupt wont be generated during CQ mode to reduce CPU overhead. But SW can still enable intr generation by clearing this reg field for debug purpose.
29	0x1	**MASK_DAT_XFER_COMPLETE_INTR**: DAT transfer complete intrerrupt wont be generated during CQ mode to reduce CPU overhead. But SW can still enable intr generation by clearing this reg field for debug purpose.
28	0x0	**CQE_SW_CLR_ALL**: Vendor SW reset for CQE. Can be used in any hang conditions. Resets CMD Queuing engine but not SD host. Active high reset SW should keep it asserted for 10cycles. No ack would be given by HW. Works as module reset - flushes internal data. Can cause data loss, if issued when bus is busy. Can be used along with SW_RESET_ALL of SD host
13:4	0xc8	**ITC_FREQ_VAL_OVR**: This reg should be programmed with sdmmc_clk frequency value.
3:0	0x2	**ITC_FREQ_MUL_OVR**: This reg is used to update ITC_FREQ_MUL_OVR field in CQE reg. Internal Timer Clock Frequency Multiplier (ITCFMUL) Field Value Description: 0h = 0.001 MHz 1h = 0.01 MHz 2h = 0.1 MHz 3h = 1 MHz Other values are reserved

SDMMCAB_VENDOR_MISC_CNTRL4_0

SPARE register1

Offset: 0x20c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffff0000 (0b1111,1111,1111,1111,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0xffff	**SDMMC_SPARE3**: Spare register bits with reset value of 1
15:0	0x0	**SDMMC_SPARE2**: Spare register bits with reset value of 0

SDMMCAB_IST_AXI_P2P_CNTRL_0

For IST AXI bridge control

Offset: 0x210
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**AXI_P2P_FIFO_HOT_RESET**: SW can reset the FIFO's in AXI P2P bridge on the functional AXI path incase there is any error or hang condition. Set this bit to 1 and clear it after 30 cycles to reset FIFO.
0	0x0	**IST_AXI_P2P_FIFO_HOT_RESET**: SW can reset the FIFO's in AXI P2P bridge on the IST AXI path incase there is any error or hang condition. Set this bit to 1 and clear it after 30 cycles to reset FIFO.

SDMMCAB_SDMEMCOMPPADCTRL_MISC_CTL_0

SDMEMCOMP Pad misc control register

This register is used to control COMP pad inputs.

Offset: 0x214
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:4	0x0	**COMP_PAD_SPARE_VDD**: Comp pad spare inputs
3:0	0x0	**COMP_PAD_SPARE_VAUXC**: Comp pad spare inputs

SDMMCAB_VENDOR_CQE_CNTRL1_0

Offset: 0x218
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x7fffffff (0b0111,1111,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
31	0x0	**VENDOR_INTR_COAL_TIMEOUT_EN**: CQE spec has defined 7-bit ICTOVAL which gives a max of 650ms timeout when 200MHzx0.001 timeout clock is used. Spec defined timeout may not be sufficient in some cases for SW. So, We have implemented a 31-bit vendor specific timer to provide huge timeout value. If set to 1, Intr coalescing timer is loaded with 31-bit timeout val as programmed in VENDOR_INTR_COAL_TIMEOUT_VAL (vendor defined) If set to 0, Intr coalescing timer is loaded with spec defined ICTOVAL - default option. SW should not set this field to 1 unless it is advertised in TRM.
30:0	0x7fffffff	**VENDOR_INTR_COAL_TIMEOUT_VAL**: This field is valid only when VENDOR_INTR_COAL_TIMEOUT_EN is set to 1. This 31-bit register holds the value of timeout in base clock cycles. For eg., if this is set to 1000 (Decimal), timer gives 1000 (Decimal) base clock cycles time.

SDMMCAB_CQE_CQVER_0

Command Queuing Version

This register provides information about the version of the eMMC CQ standard which is implemented by the CQE, in BCD format. The current version is rev 5.1

Offset: 0xf000
Read/Write: RO

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000510 (0bxxxx,xxxx,xxxx,xxxx,xxxx,0101,0001,0000)

Bit	Reset	Description
11:8	0x5	**EMMC_MAJOR_VER**: eMMC Major Version Number (digit left of decimal point), in BCD format
7:4	0x1	**EMMC_MINOR_VER**: eMMC Minor Version Number (digit right of decimal point), in BCD format
3:0	0x0	**EMMC_VER_SUFFIX**: eMMC Version Suffix (2 nd digit right of decimal point), in BCD format

SDMMCAB_CQE_CQCAP_0

Command Queuing Capabilities

This register is reserved

Offset: 0xf004
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000020c8 (0bxxxx,xxxx,xxxx,xxxx,0010,xx00,1100,1000)

Bit	Reset	Description
15:12	0x2	**ITC_FREQ_MUL**: Internal Timer Clock Frequency Multiplier (ITCFMUL) can be updated by writing into VENDOR_CQE_CNTRL0_0_CQE_ITC_FREQ_MUL_OVR. ITCFMUL and ITCFVAL indicate the frequency of the clock used for interrupt coalescing timer and for determining the SQS polling period. See ITCFVAL definition for details. Field Value Description: 0h = 0.001 MHz 1h = 0.01 MHz 2h = 0.1 MHz 3h = 1 MHz Other values are reserved

Bit	Reset	Description
9:0	0xc8	**ITC_FREQ_VAL**: Internal Timer Clock Frequency Value (ITCFVAL) CQE timer base clock is sdmmc_clk only. This register field is updated by VENDOR_CLOCK_CNTRL_BASE_CLK_FREQ. VENDOR_CLOCK_CNTRL_BASE_CLK_FREQ should be updated by SoC specific driver with sdmmc_clk frequency selected. ITCFMUL and ITCFVAL indicate the frequency of the clock used for interrupt coalescing timer and for determining the polling period when using periodic SEND_QUEUE_ STATUS (CMD13) polling. The clock frequency is calculated as ITCFVAL* ITCFMUL. For example, to encode 19.2 MHz, ITCFVAL shall be C0h (= 192 decimal) and ITCFMUL shall be 2h (0.1 MHz): 192 * 0.1 MHz = 19.2 MHz.

SDMMCAB_CQE_CQCFG_0

Command Queuing Configuration

This register controls CQE behavior affecting the general operation of command queueing module or operation of multiple tasks in the same time.

Offset: 0xf008
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxx0,xxx0,xxxx,xxx0)

Bit	Reset	Description
12	0x0	**DCMD_EN**: Direct Command (DCMD) Enable This bit indicates to the hardware whether the Task Descriptor in slot-31 of the TDL is a Data Transfer Task Descriptor, or a Direct Command Task Descriptor. CQE uses this bit when a task is issued in slot-31, to determine how to decode the Task Descriptor. Bit Value Description 1 = Task descriptor in slot-31 is a DCMD Task Descriptor 0 = Task descriptor in slot-31 is a Data Transfer Task Descriptor
8	0x0	**TASK_DESCR_SIZE**: Task Descriptor Size This bit indicates whether the task descriptor size is 128 bits or 64 bits as detailed in Data Structures section. This bit can only be configured when Command Queuing Enable bit is '0' (command queueing is disabled) Bit Value Description 1 = Task descriptor size is 128 bits 0 = Task descriptor size is 64 bits

Bit	Reset	Description
0	0x0	**CQ_EN**: Command Queuing Enable Software shall write '1' this bit when in order to enable command queueing mode (i.e. enable CQE). When this bit is 0, CQE is disabled and software controls the eMMC bus using the legacy eMMC host controller. Before software writes '1' to this bit, software shall verify that the eMMC host controller is in idle state and there are no commands or data transfers ongoing. When software wants to exit command queueing mode, it shall clear all previous tasks if such exist before setting this bit to 0.

SDMMCAB_CQE_CQCTL_0

Command Queuing Control

This register controls CQE behavior affecting the general operation of command queueing module or operation of multiple tasks in the same time.

Offset: 0xf00c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,xxxx,xxx0)

Bit	Reset	Description
8	0x0	**CLR_ALL_TASKS**: Clear All Tasks Software shall write '1' this bit when it wants to clear all the tasks sent to the device. This bit can only be written when CQE is in halt state (i.e. Halt bit is 1). When software writes 1, the value of the register is updated to '1', and CQE shall reset CQTDBR register and all other context information for all unfinished tasks. Then CQE will clear this bit. Software should poll on this bit until it is set to back 0 and may then resume normal operation, by clearing the Halt bit. CQE does not communicate to the device that the tasks were cleared. It is softwares responsibility to order the device to discard the tasks in its queue using CMDQ_TASK_MGMT command. Writing '0' to this register shall have no effect.

Bit	Reset	Description
0	0x0	**HALT**: Halt Host software shall write '1' to the bit when it wants to acquire software control over the eMMC bus and disable CQE from issuing commands on the bus. For example, issuing a Discard Task command (CMDQ_TASK_MGMT) When software writes '1', CQE shall complete the ongoing task if such a task is in progress. Once the task is completed and CQE is in idle state, CQE shall not issue new commands and shall indicate so to software by setting this bit to 1. Software may poll on this bit until it is set to 1, and may only then send commands on the eMMC bus. In order to exit halt state (i.e. resume CQE activity), software shall clear this bit (write '0'). Writing '0' when the value is already '0' shall have no effect.

SDMMCAB_CQE_CQIS_0

Command Queuing Interrupt Status

This register indicates pending interrupts that require service. Each bit in this registers is asserted in response a specific event, only if the respective bit is set in CQISTE register.

Offset: 0xf010
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	0x0	**TASK_CLR_INTR**: Task Cleared (TCL) This status bit is asserted (if CQISTE.TCL=1) when a task clear operation is completed by CQE. The completed task clear operation is either an individual task clear (CQTCLR) or clearing of all tasks (CQCTL).
2	0x0	**RESP_ERR_DETECT_INTR**: Response Error Detected Interrupt (RED) This status bit is asserted (if CQISTE.RED=1) when a response is received with an error bit set in the device status field. The contents of the device status field are listed in Section 6.13 of eMMC5.1 spec. Software uses CQRMEM register to configure which device status bit fields may trigger an interrupt, and which are masked.

Bit	Reset	Description
1	0x0	**TASK_COMPLETE_INTR**: Task Complete Interrupt (TCC) This status bit is asserted (if CQISTE.TCC=1) when at least one of the following two conditions are met: (1) A task is completed and the INT bit is set in its Task Descriptor (2) Interrupt caused by Interrupt Coalescing logic (see Section C.4.9)
0	0x0	**HALT_COMPLETE_INTR**: Halt Complete Interrupt (HAC) This status bit is asserted (if CQISTE.HAC=1) when halt bit in CQCTL register transitions from 0 to 1 indicating that host controller has completed its current ongoing task and has entered halt state.

SDMMCAB_CQE_CQISTE_0

Command Queuing Interrupt Status Enable

This register enables and disables the reporting of the corresponding interrupt to host software in CQIS register. When a bit is set ('1') and the corresponding interrupt condition is active, then the bit is CQIS is asserted. Interrupt sources that are disabled ('0') are not indicated in the CQIS register. This register is bit-index matched to CQIS register.

Offset: 0xf014
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	0x0	**TASK_CLR_INTR_STATUS_EN**: Task Cleared Status Enable (TCL) 1 = CQIS.TCL will be set when its interrupt condition is active 0 = CQIS.TCL is disabled
2	0x0	**RESP_ERR_DETECT_INTR_STATUS_EN**: Response Error Detected Status Enable (RED) 1 = CQIS.RED will be set when its interrupt condition is active 0 = CQIS.RED is disabled
1	0x0	**TASK_COMPLETE_INTR_STATUS_EN**: Task Complete Status Enable (TCC) 1 = CQIS.TCC will be set when its interrupt condition is active 0 = CQIS.TCC is disabled
0	0x0	**HALT_COMPLETE_INTR_STATUS_EN**: Halt Complete Status Enable (HAC) 1 = CQIS.HAC will be set when its interrupt condition is active 0 = CQIS.HAC is disabled

SDMMCAB_CQE_CQISGE_0

Command Queuing Interrupt Signal Enable

This register enables and disables the generation of interrupts to host software. When a bit is set ('1') and the corresponding bit in CQIS is set, then an interrupt is generated. Interrupt sources that are disabled ('0') are still indicated in the CQIS register. This register is bit-index matched to CQIS register.

Offset: 0xf018
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	0x0	**TASK_CLR_INTR_SIGNAL_EN**: Task Cleared Signal Enable (TCL) When set and CQIS.TCL is asserted, the CQE shall generate an interrupt
2	0x0	**RESP_ERR_DETECT_INTR_SIGNAL_EN**: Response Error Detected Signal Enable (TCC) When set and CQIS.RED is asserted, the CQE shall generate an interrupt
1	0x0	**TASK_COMPLETE_INTR_SIGNAL_EN**: Task Complete Signal Enable (TCC) When set and CQIS.TCC is asserted, the CQE shall generate an interrupt
0	0x0	**HALT_COMPLETE_INTR_SIGNAL_EN**: Halt Complete Signal Enable (HAC) When set and CQIS.HAC is asserted, the CQE shall generate an interrupt

SDMMCAB_CQE_CQIC_0

Interrupt Coalescing

This register controls the interrupt coalescing feature.

Offset: 0xf01c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,xxx0,xxx0,0xx0,0000,0000,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**INTR_COAL_EN**: Interrupt Coalescing Enable/Disable: When set to '0' by software, command responses are neither counted nor timed. Interrupts are still triggered by completion of tasks with INT=1 in the Task Descriptor. When set to '1', the interrupt coalescing mechanism is enabled and coalesced interrupts are generated
20	RO	0x0	**INTR_COAL_STATUS**: Interrupt Coalescing Status Bit (ICSB): This bit indicates to software whether any tasks (with INT=0) have completed and counted towards interrupt coalescing (i.e., ICSB is set if and only if IC counter > 0). Bit Value Description 1 = At least one task completion has been counted (IC counter >0) 0 = No task completions have occurred since last counter reset (IC counter =0)
16	RW	0x0	**INTR_COAL_CNTR_TIMER_RST**: Counter and Timer Reset(ICCTR): When host driver writes '1', the interrupt coalescing timer and counter are reset
15	RW	0x0	**INTR_COAL_CNTR_TH_WEN**: Interrupt Coalescing Counter Threshold Write Enable (ICCTHWEN): When software writes '1', the value ICCTH is updated with the contents written at the same cycle. When software writes '0', the value in ICCTH is not updated. NOTE: Write operations to ICCTH are only allowed when the task queue is empty.
12:8	RW	0x0	**INTR_COAL_CNTR_THRESHOLD**: Interrupt Coalescing Counter Threshold (ICCTH): Software uses this field to configure the number of task completions (only tasks with INT=0 in the Task Descriptor) which are required in order to generate an interrupt. Counter Operation: As data transfer tasks with INT=0 complete, they are counted by CQE. The counter is reset by software during the interrupt service routine. The counter stops counting when it reaches the value configured in ICCTH. The maximum allowed value is 31 NOTE: When ICCTH is 0, task completions are not counted, and counting-based interrupts are not generated. In order to write to this field, the ICCTHWEN bit must be set at the same write operation.
7	RW	0x0	**INTR_COAL_TIMEOUT_VAL_WEN**: Interrupt Coalescing Timeout Value Write Enable (ICTOVALWEN): When software writes '1', the value ICTOVAL is updated with the contents written at the same cycle. When software writes '0', the value in ICTOVAL is not updated. NOTE: Write operations to ICTOVAL are only allowed when the task queue is empty.

Bit	R/W	Reset	Description
6:0	RW	0x0	**INTR_COAL_TIMEOUT_VAL**: Interrupt Coalescing Timeout Value (ICTOVAL): Software uses this field to configure the maximum time allowed between the completion of a task on the bus and the generation of an interrupt. Timer Operation: The timer is reset by software during the interrupt service routine. It starts running when a data transfer task with INT=0 is completed, after the timer was reset. When the timer reaches the value configured in ICTOVAL field it generates an interrupt and stops. The timers unit is equal to 1024 clock periods of the clock whose frequency is specified in the Internal Timer Clock Frequency field CQCAP register. The minimum value is 01h (1024 clock periods) and the maximum value is 7Fh (127*1024 clock periods). For example, a CQCAP field value of 0 indicates a 19.2 MHz clock frequency (period = 52.08 ns). If the setting in ICTOVAL is 10h, the calculated polling period is 16*1024*52.08 ns= 853.33 us NOTE: When ICTOVAL is 0, the timer is not running, and timer-based interrupts are not generated. In order to write to this field, the ICTOVALWEN bit must be set at the same write operation.

SDMMCAB_CQE_CQTDLBA_0

Command Queuing Task Descriptor List Base Address

This register is used for configuring the lower 32 bits of the byte address of the head of the Task Descriptor List in the host memory.

Offset: 0xf020
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TASK_DESCR_LIST_BASE_ADDR**: Task Descriptor List Base Address (TDLBA) This register stores the LSB bits (bits 31:0) of the byte address of the head of the Task Descriptor List in system memory. The size of the task descriptor list is 32 * (Task Descriptor size + Transfer Descriptor size) as configured by Host driver. This address shall be set on 1 KByte boundary: The lower 10 bits of this register shall be set to 0 by software and shall be ignored by CQE.

SDMMCAB_CQE_CQTDLBAU_0

Command Queuing Task Descriptor List Base Address Upper 32 Bits

This register is used for configuring the upper 32 bits of the byte address of the head of the Task Descriptor List in the host memory.

Offset: 0xf024
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TASK_DESCR_LIST_BASE_UPPER_ADDR**: Task Descriptor List Base Upper Address (TDLBA) This register stores the MSB bits (bits 63:32) of the byte address of the head of the Task Descriptor List in system memory. The size of the task descriptor list is 32 * (Task Descriptor size + Transfer Descriptor size) as configured by Host driver. This register is reserved when using 32-bit addressing mode.

SDMMCAB_CQE_CQTDBR_0

Command Queuing Task Doorbell

Using this register, software triggers CQE to process a new task.

Offset: 0xf028
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TASK_DOORBELL**: Command Queuing Task Doorbell Software shall configure TDLBA and TDLBAU, and enable CQE in CQCFG before using this register. Writing 1 to bit n of this register triggers CQE to start processing the task encoded in slot n of the TDL. CQE always processes tasks in-order according to the order submitted to the list by CQTDBR write transactions. CQE processes Data Transfer tasks by reading the Task Descriptor and sending QUEUED_TASK_PARAMS (CMD44) and QUEUED_TASK_ADDRESS (CMD45) commands to the device. CQE processes DCMD tasks (in slot 31, when enabled) by reading the Task Descriptor, and generating the command encoded by its index and argument. The corresponding bit is cleared to '0' by CQE in one of the following events: (a) When a task execution is completed (with success or error) (b) The task is cleared using CQTCLR register (c) All tasks are cleared using CQCTL register (d) CQE is disabled using CQCFG register Software may initiate multiple tasks at the same time (batch submission) by writing 1 to multiple bits of this register in the same transaction. In the case of batch submission: CQE shall process the tasks in order of the task index, starting with the lowest index. If one or more tasks in the batch are marked with QBR, the ordering of execution will be based on said processing order. Writing 0 by software shall have no impact on the hardware, and will not change the value of the register bit.

SDMMCAB_CQE_CQTCN_0

Task Completion Notification

This register is used by CQE to notify software about completed tasks.

Offset: 0xf02c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TASK_COMPLETE_NOTIFICATION**: Task Complete Notification CQE shall set bit n of this register (at the same time it clears bit n of CQTDBR) when a task execution is completed (with success or error). When receiving interrupt for task completion, software may read this register to know which tasks have finished. After reading this register, software may clear the relevant bit fields by writing 1 to the corresponding bits.

SDMMCAB_CQE_CQDQS_0

Device Queue Status

This register stores the most recent value of the device queue status.

Offset: 0xf030
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DEVICE_QUEUE_STATUS**: Device Queue Status Every time the Host controller receives a queue status register (QSR) from the device, it updates this register with the response of status command, i.e. the device queue status.

SDMMCAB_CQE_CQDPT_0

Device Pending Tasks

This register indicates to software which tasks are queued in the device, awaiting execution.

Offset: 0xf034
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DEVICE_PENDING_TASKS**: Device Pending Tasks Bit n of this register is set if and only if QUEUED_TASK_PARAMS (CMD44) and QUEUED_TASK_ADDRESS (CMD45) were sent for this specific task and if this task has not been executed yet. CQE shall set this bit after receiving a successful response for CMD45. CQE shall clear this bit after the task has completed execution. Software needs to read this register in the task-discard procedure, when the controller is halted, to determine if the task is queued in the device. If the task is queued, the driver sends a CMDQ_TASK_MGMT (CMD48) to the device ordering it to discard the task. Then software clears the task in the CQE. Only then the software orders CQE to resume its operation using CQCTL register.

SDMMCAB_CQE_CQTCLR_0

Task Clear

This register is used for removing an outstanding task in the CQE. The register should be used only when CQE is in Halt state.

Offset: 0xf038
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TASK_CLEAR**: Command Queuing Task Clear Writing 1 to bit n of this register orders CQE to clear a task which software has previously issued. This bit can only be written when CQE is in Halt state as indicated in CQCFG register Halt bit. When software writes '1' to a bit in this register, CQE updates the value to '1', and starts clearing the data structures related to the task. CQE clears the bit fields (sets a value of 0) in CQTCLR and in CQTDBR once clear operation is complete. Software should poll on the CQTCLR until it is cleared to verify clear operation was complete. Writing to this register only clears the task in the CQE and does not have impact on the device. In order to discard the task in the device, host software shall send CMDQ_TASK _MGMT while CQE is still in Halt state. Host driver is not allowed to use this register to clear multiple tasks at the same time. Clearing multiple tasks can be done using CQCTL register. Writing 0 to a register bit shall have no impact.

SDMMCAB_CQE_CQSSC1_0

Send Status Configuration 1

The register controls the when SEND_QUEUE_STATUS commands are sent.

Offset: 0xf040
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00011000 (0bxxxx,xxxx,xxxx,0001,0001,0000,0000,0000)

Bit	Reset	Description
19:16	0x1	**SQS_CMD_BLK_CNTR**: Send Status Command Block Counter This field indicates to CQE when to send SEND_QUEUE_STATUS (CMD13) command to inquire the status of the device task queue. A value of n means CQE shall send status command on the CMD line, during the transfer of data block BLOCK_CNT-n, on the data lines, where BLOCK_CNT is the number of blocks in the current transaction. A value of 0 means that SEND_QUEUE_STATUS (CMD13) command shall not be sent during the transaction. Instead it will be sent only when the data lines are idle. A value of 1 means that STATUS command is to be sent during the last block of the transaction.

Bit	Reset	Description
15:0	0x1000	**SQS_CMD_IDLE_TIMER**: Send Status Command Idle Timer This field indicates to CQE the polling period to use when using periodic SEND_QUEUE_STATUS (CMD13) polling. Periodic polling is used when tasks are pending in the device, but no data transfer is in progress. When a SEND_QUEUE_STATUS response indicating that no task is ready for execution, CQE counts the configured time until it issues the next SEND_QUEUE_STATUS. Timer units are clock periods of the clock whose frequency is specified in the Internal Timer Clock Frequency field CQCAP register. The minimum value is 0001h (1 clock period) and the maximum value is FFFFh (65535 clock periods). Default interval is: 4096 clock periods. For example, a CQCAP field value of 0 indicates a 19.2 MHz clock frequency (period = 52.08 ns). If the setting in CQSST is 1000h, the calculated polling period is 4096*52.08 ns= 213.33 us

SDMMCAB_CQE_CQSSC2_0

Send Status Configuration 2

This register is used for configuring RCA field in SEND_QUEUE_STATUS command argument.

Offset: 0xf044
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**SQS_RCA**: Send Queue Status RCA This field provides CQE with the contents of the 16-bit RCA field in SEND_QUEUE_ STATUS (CMD13) command argument. SW should program this. CQE shall copy this field to bits 31:16 of the argument when transmitting SEND_ QUEUE_STATUS (CMD13) command.

SDMMCAB_CQE_CQCRDCT_0

Command Response for Direct-Command Task

This register is used for passing the response of a DCMD task to software.

Offset: 0xf048
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**DCMD_RESP**: Direct Command Last Response This register contains the response of the command generated by the last direct command (DCMD) task which was sent. CQE shall update this register when it receives the response for a DCMD task. This register is considered valid only after bit 31 of CQTDBR register is cleared by CQE.

SDMMCAB_CQE_CQRMEM_0

Response Mode Error Mask

This register controls the generation of Response Error Detection (RED) interrupt.

Offset: 0xf050
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xfdf9a080 (0b1111,1101,1111,1001,1010,0000,1000,0000)

Bit	Reset	Description
31:0	0xfdf9a080	**RESP_ERR_MASK**: Response Mode Error Mask This bit is used as in interrupt mask on the device status filed which is received in R1/R1b responses. Bit Value Description (for any bit i): 1 = When a R1/R1b response is received, with bit i in the device status set, a RED interrupt is generated 0 = When a R1/R1b response is received, bit i in the device status is ignored The reset value of this register is set to trigger an interrupt on all Error type bits in the device status, as defined in Section 6.13 of eMMC5.1 spec. Note: Responses to CMD13 (SQS) encode the QSR, so they are ignored by this logic.

SDMMCAB_CQE_CQTERRI_0

Task Error Information

This register is updated by CQE when an error occurs on data or command related to a task activity. When such error is detected by CQE or indicated by the eMMC controller CQE stores in CQTERRI the task IDs and the command indices of the commands which were executed on the command line and data lines when the error occurred. Software to use this information in the error recovery procedure.

Offset: 0xf054
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0xx0,0000,xx00,0000,0xx0,0000,xx00,0000)

Bit	Reset	Description
31	0x0	**DAT_XFER_ERR_FIELDS_VLD**: Data Transfer Error Fields Valid This bit is updated when an error is detected by CQE, or indicated by eMMC controller. If a data transfer is in progress when the error is detected/indicated, the bit is set to 1. If a no data transfer is in progress when the error is detected/ indicated, the bit is cleared to 0.
28:24	0x0	**DAT_XFER_ERR_TASK_ID**: Data Transfer Error Task ID This field indicates the ID of the task which was executed on the data lines when an error occurred. The field is updated if a data transfer is in progress when an error is detected by CQE, or indicated by eMMC controller.
21:16	0x0	**DAT_XFER_ERR_CMD_IDX**: Data Transfer Error Command Index This field indicates the index of the command which was executed on the data lines when an error occurred. The index shall be set to EXECUTE_READ_TASK (CMD46) or EXECUTE_WRITE_TASK (CMD47) according to the data direction. This field is updated if a data transfer is in progress when an error is detected by CQE, or indicated by eMMC controller.
15	0x0	**RESP_ERR_FIELDS_VLD**: Response Mode Error Fields Valid This bit is updated when an error is detected by CQE, or indicated by eMMC controller. If a command transaction is in progress when the error is detected/ indicated, the bit is set to 1. If a no command transaction is in progress when the error is detected/indicated, the bit is cleared to 0.
12:8	0x0	**RESP_ERR_TASK_ID**: Response Mode Error Task ID This field indicates the ID of the task which was executed on the command line when an error occurred. The field is updated if a command transaction is in progress when an error is detected by CQE, or indicated by eMMC controller.

Bit	Reset	Description
5:0	0x0	**RESP_ERR_CMD_IDX**: Response Mode Error Command Index This field indicates the index of the command which was executed on the command line when an error occurred. The field is updated if a command transaction is in progress when an error is detected by CQE, or indicated by eMMC controller.

SDMMCAB_CQE_CQCRI_0

Command Response Index

This register stores the index of the last received command response.

Offset: 0xf058
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**LAST_CMD_RESP_IDX**: Last Command Response index This field stores the index of the last received command response. CQE shall update the value every time a command response is received.

SDMMCAB_CQE_CQCRA_0

Command Response Argument

This register stores the index of the last received command response.

Offset: 0xf05c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**LAST_CMD_RESP_ARG**: Last Command Response Argument This field stores the argument of the last received command. CQE shall update the value every time a command response is received.

9.5.4.2 SDMMCA Registers

SDMMCA_SYSTEM_ADDRESS_0

32-bit Block Count (SDMA System Address) Register

When Host Version 4 Enable is set to 0 in the Host Control 2 register, SDMA
uses this register as system address in only 32-bit addressing mode. Auto CMD23 cannot be used with SDMA.
When Host Version 4 Enable is set to 1, SDMA uses ADMA System Address
register (05Fh-058h) instead of using this register to support both 32-bit and 64bit addressing. This register is re-assigned to 32-bit Block Count and then SDMA may use Auto CMD23.

(1) SDMA System Address (Host Version 4 Enable = 0)
This register contains the system memory address for a SDMA transfer in 32bit addressing mode. When the Host Controller stops a SDMA transfer, this register shall point to the system address of the next
contiguous data position. It can be accessed only if no transaction is executing (i.e., after a transaction has stopped).
Reading this register during SDMA transfers may return an invalid value. The Host Driver shall initialize this register before starting a SDMA
transaction.
After SDMA has stopped, the next system address of the next contiguous data position can be read from this register. The SDMA transfer waits at the
every boundary specified by the SDMA Buffer Boundary in the Block Size register.
The Host Controller generates DMA Interrupt to request the Host Driver to update this register. The Host Driver sets the next system address
of the next data position to this register. When the most upper byte of this register (003h) is written, the Host Controller
restarts the SDMA transfer. When restarting SDMA by setting Continue Request in the Block Gap Control register, the Host Controller
shall start at the next contiguous address stored here in the SDMA System Address register.
ADMA does not use this register.

(2) 32-bit Block Count (Host Version 4 Enable = 1)
Host Controller Version 4.10 re-defines this register as 32-bit Block Count
(Refer to Section 1.15 in SD Host spec4.1 for more details about block count extension). In version 4.00, this register may be used

as 32-bit block count only for Auto CMD23 to set the argument of the CMD23 while executing Auto CMD23.

FFFF_FFFFh 4G - 1 block
... ...
0000_0002h 2 blocks
0000_0001h 1 block
0000_0000h Stop Count

The Host Controller would decrement the block count of this register every block transfer and data transfer stops when the count reaches zero.
This register should be accessed only when no transaction is executing.
Reading this register during data transfers may return invalid value.

Offset: 0x0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ADDRESS**

SDMMCA_BLOCK_SIZE_BLOCK_COUNT_0

Block Size Register

HOST_DMA_BUFFER_SIZE
The large contiguous memory space may not be available in the virtual memory system. To perform long SDMA transfer, SDMA System Address register shall be updated at every system memory boundary during SDMA transfer.
These bits specify the size of contiguous buffer in the system memory. The SDMA transfer shall wait at the every boundary specified by these fields and the Host Controller generates the DMA Interrupt to request the Host Driver to update the SDMA System Address register. At the end of transfer, the Host Controller may issue or may not issue DMA Interrupt. In particular, DMA Interrupt shall not be issued after Transfer Complete Interrupt is issued.
In case of this register is set to 0 (buffer size = 4K bytes), lower 12-bit of byte address points data in the contiguous buffer and the upper 20-bit points the location of the buffer in the system memory. The SDMA transfer stops when the Host Controller detects carry out of the address from bit 11 to 12.
These bits shall be supported when the SDMA Support in the Capabilities

register is set to 1 and this function is active when the DMA Enable in the Transfer Mode register is set to 1.
ADMA does not use this register.
000b 4K bytes (Detects A11 carry out)
001b 8K bytes (Detects A12 carry out)
010b 16K Bytes (Detects A13 carry out)
011b 32K Bytes (Detects A14 carry out)
100b 64K bytes (Detects A15 carry out)
101b 128K Bytes (Detects A16 carry out)
110b 256K Bytes (Detects A17 carry out)
111b 512K Bytes (Detects A18 carry out)
XFER_BLOCK_SIZE_11_0
This register specifies the block size of data transfers for CMD17, CMD18, CMD24, CMD25, and CMD53. Values ranging from 1 up to the maximum buffer size can be set. In case of memory, it shall be set up to 512 bytes (Refer to Implementation Note in Section 1.7.2). It can be accessed only if no transaction is executing (i.e., after a transaction has stopped). Read operations during transfers may return an invalid value, and write operations shall be ignored.
0800h 2048 Bytes
.....
0200h 512 Bytes
01FFh 511 Bytes
.....
0004h 4 Bytes
0003h 3 Bytes
0002h 2 Bytes
0001h 1 Byte
0000h No data transfer
16-bit BLOCKS_COUNT
Host Controller Version 4.10 extends block count to 32-bit (Refer to Section 1.15 in SD host spec4.1).
Selection of either 16-bit Block Count register or 32-bit Block Count register is defined as follows:
(1) If Host Version 4 Enable in the Host Control 2 register is set to 0 or 16-bit Block Count register is set to non-zero, 16-bit Block Count register is selected.
(2) If Host Version 4 Enable is set to 1 and 16-bit Block Count register is set to zero, 32-bit Block Count register is selected.
Use of 16-bit/32-bit Block Count register is enabled when Block Count Enable in the Transfer Mode register is set to 1 and is valid only for multiple block transfers. The Host Driver shall set this
register to a value between 1 and the maximum block count. The Host Controller decrements the block count after each block transfer and stops when the count reaches zero. Setting the block count to 0 results in no data blocks is transferred. This register should be accessed only when no transaction is executing (i.e., after transactions are stopped). During data transfer, read operations on this register may return an invalid value and write operations are ignored.
When a suspend command is completed, the number of blocks yet to be transferred

can be determined by reading this register. Before issuing a resume command, the Host Driver shall restore the previously saved block count.
FFFFh 65535 blocks
.....
0002h 2 blocks
0001h 1 block
0000h Stop Count

Offset: 0x4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,x000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**BLOCKS_COUNT**
14:12	0x0	**HOST_DMA_BUFFER_SIZE**: **0** = DMA4K **1** = DMA8K **2** = DMA16K **3** = DMA32K **4** = DMA64K **5** = DMA128K **6** = DMA256K **7** = DMA512K
11:0	0x0	**XFER_BLOCK_SIZE_11_0**

SDMMCA_ARGUMENT_0

Argument 1 Register

COMMAND_ARGUMENT
The SD Command Argument is specified as bit39-8 of Command-Format in the Physical Layer Specification.

Offset: 0x8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**COMMAND_ARGUMENT**

SDMMCA_CMD_XFER_MODE_0

Command and Transfer Mode Register

COMMAND_INDEX
These bits shall be set to the command number (CMD0-63, ACMD0-63) that is specified in bits 45-40 of the Command-Format in the Physical Layer Specification and SDIO Card Specification.
COMMAND_TYPE
There are three types of special commands: Suspend, Resume and Abort. These bits shall be set to 00b for all other commands.
(1) Suspend Command
If the Suspend command succeeds, the Host Controller shall assume the SD Bus has been released and that it is possible to issue the next command, which uses the DAT line. The Host Controller shall de-assert Read Wait for read transactions and stop checking busy for write transactions. The interrupt cycle shall start, in 4-bit mode. If the Suspend command fails, the Host Controller shall maintain its current state, and the Host Driver shall restart the transfer by setting Continue Request in the Block Gap Control register. (Refer to 3.12.1 Suspend Sequence)
(2) Resume Command
The Host Driver re-starts the data transfer by restoring the registers in the range of 000-00Dh. (Refer to Figure 1-4 in section 1.6 for the register map.) The Host Controller shall check for busy before starting write transfers.
(3) Abort Command
If this command is set when executing a read transfer, the Host Controller shall stop reads to the buffer. If this command is set when executing a write transfer, the Host Controller shall stop driving the DAT line. After issuing the Abort command, the Host Driver should issue a software reset. (Refer to 3.8 Abort Transaction)
11b Abort CMD12, CMD52 for writing "I/O Abort" in CCCR
10b Resume CMD52 for writing "Function Select" in CCCR
01b Suspend CMD52 for writing "Bus Suspend" in CCCR
00b Normal Other commands
DATA_PRESENT_SELECT
This bit is set to 1 to indicate that data is present and shall be transferred using the DAT line. It is set to 0 for the following:
(1) Commands using only CMD line (ex. CMD52).
(2) Commands with no data transfer but using busy signal on DAT[0] line (R1b or R5b ex. CMD38)
(3) Resume command

CMD_INDEX_CHECK_EN
If this bit is set to 1, the Host Controller shall check the Index field in the response to see if it has the same value as the command index. If it is not, it is reported as a Command Index Error. If this bit is set to 0, the Index field is not checked.
CMD_CRC_CHECK_EN
If this bit is set to 1, the Host Controller shall check the CRC field in the response. If an error is detected, it is reported as a Command CRC Error. If this bit is set to 0, the CRC field is not checked. The position of CRC field is determined according to the length of the response. (Refer to definition in D01-00 and Table 2-10 below.)
Sub Command Flag
This bit is added from Version 4.10 to distinguish a main command or sub command (Refer to Section 1.17). When issuing a main command, this bit is set to 0 and when issuing a sub command, this bit is set to 1. Setting of this bit is checked by Sub Command Status in the Present State register.
Host Driver manages whether main or sub command. Host Controller does not refer to this bit to issue a command.
1 Sub Command
0 Main Command
RESP_TYPE_SELECT
Normal Mode:
00 No Response
01 Response Length 136
10 Response Length 48
11 Response Length 48 check Busy after response
Response Interrupt Disable
Host Controller Version 4.00 supports response error check function to avoid overhead of response error check by Host Driver. Only R1 or R5 can be checked.
If Host Driver checks response error, sets this bit to 0 and waits Command Complete Interrupt and then check the response register.
If Host Controller checks response error, sets this bit to 1 and sets Response Error Check Enable to 1. Command Complete Interrupt is disabled by this bit regardless of Command Complete Signal Enable.
0 Response Interrupt is enabled
1 Response Interrupt is disabled
Response Error Check Enable
Host Controller Version 4.00 supports response error check function to avoid overhead of response error check by Host Driver. Only R1 or R5 can be checked.
If Host Driver checks response error, this bit is set to 0 and Response Interrupt Disable is set to 0.
If Host Controller checks response error, sets this bit to 1 and sets Response Interrupt Disable to 1. Response Type R1 / R5 selects either R1 or R5 response type. If an error is detected, Response Error Interrupt is generated in the Response Error Interrupt Status register
0 Response Error Check is disabled

1 Response Error Check is enabled
Response Type R1 / R5
When response error check is enabled, this bit selects either R1 or R5 response types. Two types of response check is supported: R1 for memory and R5 for SDIO.
Error Statuses Checked in R1
Bit31 OUT_OF_RANGE
Bit30 ADDRESS_ERROR
Bit29 BLOCK_LEN_ERROR
Bit26 WP_VIOLATION
Bit25 CARD_IS_LOCKED
Bit23 COM_CRC_ERROR
Bit21 CARD_ECC_FAILED
Bit20 CC_ERROR
Bit19 ERROR
Response Flags Checked in R5
Bit07 COM_CRC_ERROR
Bit03 ERROR
Bit01 FUNCTION_NUMBER
Bit00 OUT_OF_RANGE
0 R1 (Memory)
1 R5 (SDIO)
MULTI_BLOCK_SELECT - Multi / Single Block Select
This bit is set when issuing multiple-block transfer commands using DAT line. For any other commands, this bit shall be set to 0. If this bit is 0, it is not necessary to set the Block Count register. (Refer to Table 2-8)
1 Multiple Block
0 Single Block
DATA_XFER_DIR_SEL - Data Transfer Direction Select
This bit defines the direction of DAT line data transfers. The bit is set to 1 by the Host Driver to transfer data from the SD card to the SD Host Controller and it is set to 0 for all other commands.
1 Read (Card to Host)
0 Write (Host to Card)
AUTO_CMD12_EN - Auto CMD Enable
This field determines use of auto command functions.
00b Auto Command Disabled
01b Auto CMD12 Enable
10b Auto CMD23 Enable
11b Auto CMD Auto Select
There are three methods to stop Multiple-block read/write operation by CMD23 or CMD12. In the other operations (ex. single read/write operation), this field is set to 00b.

(1) Auto CMD12 Enable
When this field is set to 01b, the Host Controller issues CMD12 automatically when last block transfer is completed. Auto CMD12 error is indicated to the Auto CMD Error Status register.

The Host Driver shall not set this bit if the command does not require CMD12. In particular, secure commands
defined in the Part 3 File Security specification do not require CMD12.
When Host Version 4 Enable =0, CMD12 is issued when 16-bit Block Count is expired.
When Host Version 4 Enable =1, CMD12 is issued when 16-bit Block Count or 32-bit Block Count is expired.

(2) Auto CMD23 Enable
When this bit field is set to 10b, the Host Controller issues a CMD23 automatically before issuing a command specified in the Command Register. The Host Controller Version 3.00 and later shall support this function. The following conditions are required to use the Auto CMD23.
Auto CMD23 Supported (Host Controller Version is 3.00 or later)
A memory card that supports CMD23 (SCR[33]=1)
If DMA is used, it shall be ADMA.
Only when CMD18 or CMD25 is issued
(Note, the Host Controller doesnt check command index.)
Auto CMD23 can be used with or without ADMA. By writing the Command register, the Host Controller issues a CMD23 first and then issues a command specified by the Command Index in Command register. If response errors of CMD23 are detected, the second command is not issued. A CMD23 error is indicated in the Auto CMD Error Status register.
32-bit block count value for CMD23 is set to 32-bit Block Count (SDMA System Address) register.

(3) Auto CMD Auto Select (Version 4.10)
As CMD23 is optional for SD Memory Card except UHS104 Card, If card supports CMD23, Auto CMD23 should be used instead of Auto CMD12. Host Controller Version 4.10 defines this "Auto CMD Auto Select" mode.
Selection of Auto CMD depends on setting of CMD23 Enable in the Host Control 2 register which indicates whether card supports CMD23. If CMD23 Enable =1, Auto CMD23 is used and if CMD23 Enable =0, Auto CMD12 is used. In case of Version 4.10 or later, use of Auto CMD Auto Select is recommended rather than use of Auto CMD12 Enable or Auto CMD23 Enable.
BLOCK_COUNT_EN - Block Count Enable
This bit is used to enable the Block Count register, which is only relevant for multiple block transfers. When this bit is 0, the Block Count register is disabled, which is useful in executing an infinite transfer. (Refer to Table 2-8)
Host Driver should set this bit to 0 when ADMA is used.
DMA_EN - DMA Enable
This bit enables DMA functionality as described in section 1.4. DMA can be enabled only if it is supported as indicated in the Capabilities register. One of the DMA modes can be selected by DMA Select in the Host Control register. If DMA is not supported, this bit is meaningless and shall always read 0. If this bit is set to 1, a DMA operation shall begin when the Host Driver writes to the upper byte of Command register (00Fh).

Offset: 0xc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0000,0000,xxxx,xxx0,0000,0000)

Bit	Reset	Description
29:24	0x0	**COMMAND_INDEX**
23:22	0x0	**COMMAND_TYPE**: **0** = NORMAL **1** = SUSPEND **2** = RESUME **3** = ABORT
21	0x0	**DATA_PRESENT_SELECT**: **0** = NO_DATA_TRANSFER **1** = DATA_TRANSFER
20	0x0	**CMD_INDEX_CHECK_EN**: **0** = DISABLE **1** = ENABLE
19	0x0	**CMD_CRC_CHECK_EN**: **0** = DISABLE **1** = ENABLE
18	0x0	**SUB_CMD_FLAG**: **0** = MAIN_CMD **1** = SUB_CMD
17:16	0x0	**RESP_TYPE_SELECT**: **0** = NO_RESPONSE **1** = RESP_LENGTH_136 **2** = RESP_LENGTH_48 **3** = RESP_LENGTH_48BUSY
8	0x0	**RESP_INT_DIS**: **0** = ENABLE **1** = DISABLE
7	0x0	**RESP_ERR_CHK_EN**: **0** = DISABLE **1** = ENABLE
6	0x0	**RESP_TYPE**: **0** = R1 **1** = R5
5	0x0	**MULTI_BLOCK_SELECT**: **0** = DISABLE **1** = ENABLE
4	0x0	**DATA_XFER_DIR_SEL**: **0** = WRITE **1** = READ

Bit	Reset	Description
3:2	0x0	**AUTO_CMD12_EN**: **0** = DISABLE **1** = CMD12 **2** = CMD23 **3** = AUTO_CMD_AUTO_SEL
1	0x0	**BLOCK_COUNT_EN**: **0** = DISABLE **1** = ENABLE
0	0x0	**DMA_EN**: **0** = DISABLE **1** = ENABLE

SDMMCA_RESPONSE_R0_R1_0

Command Response Registers
The Table below describes the mapping of command responses from the SD Bus to this register for each response type. In the table, R[] refers to a bit range within the response data as transmitted on the SD Bus, REP[] refers to a bit range within the Response register.

Kind of Response	Meaning of Response	Response Field	Response Register
R1, R1b (normal response)	Card Status	R [39:8]	REP [31:0]
R1b (Auto CMD12 response)	Card Status for Auto CMD12	R [39:8]	REP [127:96]
R1 (Auto CMD23 response)	Card Status for Auto CMD23	R [39:8]	REP [127:96]
R2 (CID, CSD register)	CID or CSD reg. incl.	R [127:8]	REP [119:0]
R3 (OCR register)	OCR register for memory	R [39:8]	REP [31:0]
R4 (OCR register)	OCR register for I/O etc	R [39:8]	REP [31:0]
R5,R5b	SDIO response	R [39:8]	REP [31:0]
R6 (Published RCA response)	New published RCA[31:16] etc	R [39:8]	REP [31:0]

Response Bit Definition for Each Response Type
In UHS-II mode, the response of CM-TRAN abort CCMD (4-byte) is stored in offset 13h-10h and the response of SD-TRAN abort CCMD (8-byte) is stored in offset 1Fh-18h
Command Response [31:0] (R0) Register

Offset: 0x10
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CMD_RESP_31_16**
15:0	0x0	**CMD_RESP_15_0**

SDMMCA_RESPONSE_R2_R3_0

Command Response [63:32] (R2) Register

Offset: 0x14
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CMD_RESP_63_48**
15:0	0x0	**CMD_RESP_47_32**

SDMMCA_RESPONSE_R4_R5_0

Command Response [95:64] (R4) Register

Offset: 0x18
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CMD_RESP_95_80**
15:0	0x0	**CMD_RESP_79_64**

SDMMCA_RESPONSE_R6_R7_0

Command Response [127:96] (R6) Register

Offset: 0x1c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0x0	**CMD_RESP_127_112**
15:0	0x0	**CMD_RESP_111_96**

SDMMCA_BUFFER_DATA_PORT_0

Buffer Data Port Register

The Host Controller Buffer can be accessed through this 32-bit Data Port Register.

Offset: 0x20
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**BUFFER_DATA**

SDMMCA_PRESENT_STATE_0

Present State Register

CMD_LINE_LEVEL - CMD Line Signal Level
This status is used to check the CMD line level to recover from errors, and
for debugging.
DAT_3_0_LINE_LEVEL - DAT[3:0] Line Signal Level
This status is used to check the DAT line level to recover from errors, and for
debugging. This is especially useful in detecting the busy signal level from
DAT[0].
Sub Command Status
The Command register and Response register are commonly used for

main command and sub command. This status is used to distinguish which response error statuses, main command or sub command, indicated in the Error Interrupt Status register or in the UHS-II Error Interrupt Status register. Refer to Section 1.17 about details of response error statuses. Just before reading of this register, the Sub Command Flag of the Command register or the UHS-II Command register is copied to this status. This status is effective not only when Response Error interrupt is generated but also when data error interrupt is generated with Command Not Issued by Error (D27 of this register) or Auto CMD Error interrupt is generated with Command Not Issued by Error by Auto CMD12 in the Auto CMD Error Status register.

1 Sub Command Status
0 Main Command Status
Command Not Issued by Error
Setting of this status indicates that a command cannot be issued due to an error except Auto CMD12 error. (Equivalent error status by Auto CMD12 error is defined as Command Not Issued By Auto CMD12 Error in the Auto CMD Error Status register.) This status is set to 1 when Host Controller cannot issue a command after setting Command register or
UHS-II Command register. Refer to Section 3.10 about 2L-HD error case in UHS-II mode. Sub Command Status (D28) indicates which command is not issued (main or sub).

1 Command cannot be issued
0 No error for issuing a command

Host Regulator Voltage Stable
This status is added from Version 4.10 and is used to check whether host regulator voltage is stable for switching signal voltage of UHS-I mode.

1 Host Regulator Voltage is stable
0 Host Regulator Voltage is not stable

Support of this function is checked by reading this status after that Software Reset For All in the Software Reset register is cleared by the Host Controller in initialization. Setting this status to 1 means that this function is supported by the Host Controller.

This status may be related to 1.8V Signaling Enable in the Host Control 2 register. Changing 1.8V Signaling Enable causes unstable of host regulator voltage for I/O cell. Then once this status is set to 0 and retrieved to 1 when host regulator voltage is stable again. When executing power cycle, Host Driver also executes Software Reset For All and it clears 1.8V Signaling Enable to go back signal voltage to 3.3V.

If this status is not supported, Host Driver should take more than 5ms for stable time of host voltage regulator from changing 1.8V Signaling

Enable. Specific Host Driver may use a specific time, which is provided by Host System, instead of using 5ms.
WRITE_PROTECT_LEVEL - Write Protect Switch Pin Level
CARD_DETECT_PIN_LEVEL - Card Detect Pin Level
CARD_STATE_STABLE - Card State Stable
CARD_INSERTED - Card Inserted
BUFFER_READ_EN - Buffer Read Enable
This status is used for non-DMA read transfers.
BUFFER_WRITE_EN - Buffer Write Enable
This status is used for non-DMA write transfers.
READ_XFER_ACTIVE - Read Transfer Active
WRITE_XFER_ACTIVE - Write Transfer Active
RETUNING_REQUEST - Re-Tuning Request
DAT_LINE_ACTIVE - DAT Line Active
CMD_INHIBIT_DAT - Command Inhibit (DAT)
CMD_INHIBIT_CMD - Command Inhibit (CMD)

Offset: 0x24
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxx0,0x00,0000,0000,xxxx,0000,0000,0000)
PROD: 0x000b0000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	PROD	Description
28	0x0	_NONE_	**SUB_CMD_STATUS**
27	0x0	_NONE_	**CMD_NOT_ISSUED_BY_ERROR**
25	0x0	_NONE_	**HOST_REG_VOLTAGE_STABLE**
24	0x0	_NONE_	**CMD_LINE_LEVEL**: **0** = LOW **1** = HIGH
23:20	0x0	_NONE_	**DAT_3_0_LINE_LEVEL**
19	0x0	ENABLED	**WRITE_PROTECT_LEVEL**: **0** = PROTECTED **1** = ENABLED
18	0x0	_NONE_	**CARD_DETECT_PIN_LEVEL**: **0** = NO_CARD **1** = CARD
17	0x0	INSERTED	**CARD_STATE_STABLE**: **0** = DEBOUNCE **1** = INSERTED

Bit	Reset	PROD	Description
16	0x0	INSERTED	**CARD_INSERTED**: **0** = DEBOUNCE **1** = INSERTED
11	0x0	_NONE_	**BUFFER_READ_EN**: **0** = DISABLE **1** = ENABLE
10	0x0	_NONE_	**BUFFER_WRITE_EN**: **0** = DISABLE **1** = ENABLE
9	0x0	_NONE_	**READ_XFER_ACTIVE**: **0** = NO_DATA **1** = TRANSFERING
8	0x0	_NONE_	**WRITE_XFER_ACTIVE**: **0** = NO_DATA **1** = TRANSFERING
7:4	0x0	_NONE_	**DAT_7_4_LINE_LEVEL**
3	0x0	_NONE_	**RETUNING_REQUEST**: **0** = NOT_REQUIRED **1** = REQUIRED
2	0x0	_NONE_	**DAT_LINE_ACTIVE**: **0** = INACTIVE **1** = ACTIVE
1	0x0	_NONE_	**CMD_INHIBIT_DAT**: **0** = INACTIVE **1** = ACTIVE
0	0x0	_NONE_	**CMD_INHIBIT_CMD**: **0** = INACTIVE **1** = ACTIVE

SDMMCA_POWER_CONTROL_HOST_0

Power Control / Host Control Register

WAKEUP_ON_CARD_REMOVAL - Wakeup Event Enable On SD Card Removal
This bit enables wakeup event via Card Removal assertion in the Normal Interrupt Status register. FN_WUS (Wake Up Support) in CIS does not affect this bit.
WAKEUP_ON_CARD_INSERTION - Wakeup Event Enable On SD Card Insertion
This bit enables wakeup event via Card Insertion assertion in the Normal Interrupt Status register. FN_WUS (Wake Up Support) in CIS does not affect this bit.
WAKEUP_ON_CARD_INTERRUPT - Wakeup Event Enable On Card Interrupt
This bit enables wakeup event via Card Interrupt assertion in the Normal Interrupt Status register. This bit can be set to 1 if FN_WUS (Wake Up Support) in CIS is set to 1.
INTERRUPT_AT_BLOCK_GAP - Interrupt At Block Gap

This bit is valid only in 4-bit mode of the SDIO card and selects a sample point in the interrupt cycle. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer. Setting to 0 disables interrupt detection during a multiple block transfer. If the SD card cannot signal an interrupt during a multiple block transfer, this bit should be set to 0. When the Host Driver detects an SD card insertion, it shall set this bit according to the CCCR of the SDIO card.
READ_WAIT_CONTROL - Read Wait Control
The read wait function is optional for SDIO cards. If the card supports read wait, set this bit to enable use of the read wait protocol to stop read data using the DAT[2] line. Otherwise, the Host Controller has to stop the SD Clock to hold read data, which restricts commands generation. When the Host Driver detects an SD card insertion, it shall set this bit according to the CCCR of the SDIO card. If the card does not support read wait, this bit shall never be set to 1 otherwise DAT line conflict may occur. If this bit is set to 0, Suspend/Resume cannot be supported. In UHS-II mode, Read Wait is disabled and DAT[2] line is used for Interrupt Signal from UHS-II Card.
CONTINUE_REQUEST - Continue Request
This bit is used to restart a transaction, which was stopped using the Stop At Block Gap Request. To cancel stop at the block gap, set Stop At Block Gap Request to 0 and set this bit 1 to restart the transfer.
The Host Controller automatically clears this bit when the transaction re-starts. If Stop At Block Gap Request is set to 1, any write to this bit is ignored.
In SD mode, this bit is cleared in either of the following cases:
(1) In the case of a read transaction, the DAT Line Active changes from 0 to 1 as a read transaction restarts.
(2) In the case of a write transaction, the Write Transfer Active changes from 0 to 1 as the write transaction restarts.
Therefore, it is not necessary for Host Driver to set this bit to 0. If Stop At Block Gap Request is set to 1, any write to this bit is ignored.
STOP_AT_BLOCK_GAP_REQUEST - Stop At Block Gap Request
This bit is used to stop executing read and write transaction at the next block gap for non-DMA, SDMA and ADMA transfers. The Host Driver shall leave this bit set to 1 until the Transfer Complete is set to 1. Clearing both Stop At Block Gap Request and Continue Request shall not cause the transaction to restart. When Host Controller version is 1.00, the Host Driver can set this bit if the card supports Read Wait Control. When Host Controller version is 2.00 or higher, the Host Driver can set this bit regardless of the card supports Read Wait Control. The Host Controller shall stop read transfer by using Read Wait or stopping SD clock. In case of write transfers in which the Host Driver writes data to the Buffer Data Port register, the Host Driver shall set this bit after all block data is written. If this bit is set to 1, the Host Driver shall not write data to Buffer Data Port register. This bit affects Read Transfer Active, Write Transfer Active, DAT Line Active and Command Inhibit (DAT) in the Present State register.
Regarding detailed control of bits D01 and D00, refer to Section 3.8 and 3.12.
SD_BUS_VOLTAGE_SELECT - The Host doesnt support the bus voltage selections.For SDMMC1 Interfaces,

voltage switching between 3.3V to 1.8V is done by programming the PMU through I2C Interface.
SD_BUS_POWER - SD Bus Power
Before setting this bit, the SD Host Driver shall set SD Bus Voltage Select. If the Host Controller detects the No Card state, this bit shall be cleared.
If this bit is cleared, the Host Controller shall immediately stop driving CMD and DAT[3:0] (tri-state) and drive SDCLK to low level (refer to Section 2.2.14).
CARD_DETECT_SIGNAL_DETECT - Card Detect Signal Selection
This bit selects source for the card detection.
When the source for the card detection is switched, the interrupt should be disabled during the switching period by clearing the Interrupt Status/Signal Enable register in order to mask unexpected interrupt being caused by the glitch.
The Interrupt Status/Signal Enable should be disabled during over the period of debouncing.
CARD_DETECT_TEST_LVL - Card Detect Test Level
This bit is enabled while the Card Detect Signal Selection is set to 1 and it indicates card inserted or not.
EXTENDED_DATA_TRANSFER_WIDTH (VENDOR Bit)
1:8-bit Mode,DATA_XFER_WIDTH is ignored.
0:Card bus width is as per DATA_XFER_WIDTH value

DMA Select
This field is used to select DMA type. The Host Driver shall check support of DMA modes by referring the Capabilities register. Selected DMA is enabled by DMA Enable of the Transfer Mode register in SD mode and DMA Enable of UHS-II Transfer Mode register in UHS-II mode.

(1) Up to Version 3.00
When Host Version 4 Enable is set to 0, setting of this field is compatible to Host Controller Version 3.00.
SDMA is initiated by writing to the Command register when this field is set to 00b and the SDMA System Address register (32-bit) is used.
SDMA does not support 64-bit addressing.
ADMA2 is initiated by writing to the Command register when this field is set to 10b or 11b. Lower 32-bit of the ADMA System Address register is used when this field is set to 10b and 64-bit of the ADMA System Address register is used when this field is set to 11b. Support of 64-bit System Addressing is indicated by 64-bit System Address Support for V3 in the Capabilities register. 64-bit AMDA2 uses 96-bit Descriptor.

00 SDMA is selected
01 Reserved (New assignment is not allowed)
10 32-bit Address ADMA2 is selected
11 64-bit Address ADMA2 is selected (Optional)

(2) Version 4.00 or later
When Host Version 4 Enable is set to 1, setting of this field is changed as follows. SDMA is initiated by Host Driver writes to the Command register when this

field is set to 00b.
ADMA2 is initiated by Host Driver writes to the Command register when this field is set to 10b or 11b and by ADMA3 sets to the ADMA System Address register when this field is set to 11b.
ADMA3 is initiated by Host Driver writes to the ADMA3 ID Address register when this field is set to 11b.

00 SDMA is selected
01 Not Used (New assignment is not allowed)
10 ADMA2 is selected (AMDA3 is not supported or disabled)
11 ADMA2 or ADMA3/CQE is selected

Support of 64-bit DMA and 128-bit Descriptor is indicated by 64-bit System Address Support for V4 in the Capabilities register. If the support bit is set to 1, all supported DMAs (depends on Support, ADMA2 Support and ADMA3 Support) shall support 64-bit addressing. 64-bit Addressing in the Host Controller 2 register selects either 32-bit or 64-bit system addressing of DMAs.

HIGH_SPEED_EN - High Speed Enable
This bit is optional. Before setting this bit, the Host Driver shall check the High Speed Support in the Capabilities register. If this bit is set to 0 (default), the Host Controller outputs CMD line and DAT lines at the falling edge of the SD Clock (up to 25MHz). If this bit is set to 1, the Host Controller outputs CMD line and DAT lines at the rising edge of the SD Clock (up to 50MHz).

If Preset Value Enable in the Host Control 2 register is set to 1, Host Driver needs to reset SD Clock Enable before changing this field to avoid generating clock glitches. After setting this field, the Host Driver sets SD Clock Enable again.
DATA_XFER_WIDTH - Data Transfer Width
This bit selects the data width of the Host Controller. The Host Driver shall set it to match the data width of the SD card.
LED_CONTROL - LED Control
This bit is used to caution the user not to remove the card while the SD card is being accessed. If the software is going to issue multiple SD commands, this bit can be set during all these transactions. It is not necessary to change for each transaction.

Offset: 0x28
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,x000,xxxx,0000,xxxx,0000,0000,0000)
PROD: 0x00000002 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx0x,xx1x)

Bit	Reset	PROD	Description
26	0x0	_NONE_	**WAKEUP_ON_CARD_REMOVAL**: **0** = DISABLE **1** = ENABLE
25	0x0	_NONE_	**WAKEUP_ON_CARD_INSERTION**: **0** = DISABLE **1** = ENABLE
24	0x0	_NONE_	**WAKEUP_ON_CARD_INTERRUPT**: **0** = DISABLE **1** = ENABLE
19	0x0	_NONE_	**INTERRUPT_AT_BLOCK_GAP**: **0** = DISABLE **1** = ENABLE
18	0x0	_NONE_	**READ_WAIT_CONTROL**: **0** = DISABLE **1** = ENABLE
17	0x0	_NONE_	**CONTINUE_REQUEST**: **0** = IGNORED **1** = RESTART
16	0x0	_NONE_	**STOP_AT_BLOCK_GAP_REQUEST**: **0** = TRANSFER **1** = STOP
11:9	0x0	_NONE_	**SD_BUS_VOLTAGE_SELECT**: **5** = V1_8 **6** = V3_0 **7** = V3_3
8	0x0	_NONE_	**SD_BUS_POWER**: **0** = POWER_OFF **1** = POWER_ON
7	0x0	_NONE_	**CARD_DETECT_SIGNAL_DETECT**: **0** = SDCD **1** = CARD_DTECT_TST_LVL
6	0x0	_NONE_	**CARD_DETECT_TEST_LVL**: **0** = NO_CARD **1** = CARD_INSERTED
5	0x0	NOBIT_8	**EXTENDED_DATA_TRANSFER_WIDTH**: **0** = NOBIT_8 **1** = BIT_8
4:3	0x0	_NONE_	**DMA_SELECT**: SW should select ADMA3_CQE in eMMC CMD queuing mode. **0** = SDMA **1** = RSVD **2** = ADMA2 **3** = ADMA3_CQE
2	0x0	_NONE_	**HIGH_SPEED_EN**: **0** = NORMAL_SPEED **1** = HIGH_SPEED

Bit	Reset	PROD	Description
1	0x0	BIT_4	**DATA_XFER_WIDTH**: **0** = BIT_1 **1** = BIT_4
0	0x0	_NONE_	**LED_CONTROL**: **0** = OFF **1** = ON

SDMMCA_SW_RESET_TIMEOUT_CTRL_CLOCK_CONTROL_0

Clock Control Register

SW_RESET_FOR_DAT_LINE - Software Reset For DAT Line
Only part of data circuit is reset. DMA circuit is also reset.
The following registers and bits are cleared by this bit:
Buffer Data Port register
Buffer is cleared and initialized.
Present State register
Buffer Read Enable
Buffer Write Enable
Read Transfer Active
Write Transfer Active
DAT Line Active
Command Inhibit (DAT)
Block Gap Control register
Continue Request
Stop At Block Gap Request
Normal Interrupt Status register
Buffer Read Ready
Buffer Write Ready
DMA Interrupt
Block Gap Event
Transfer Complete
SW_RESET_FOR_CMD_LINE - Software Reset For CMD Line
Only part of command circuit is reset to be able to issue a command. From Version 4.10, this bit is also used to initialize UHS-II command circuit. This reset is effective only command issuing circuit (including response error statuses related to Command Inhibit (CMD) control) and does not affect data transfer circuit. Host Controller can continue data transfer even this reset is executed during handling of sub command response errors.

The following registers and bits are cleared by this bit:
Present State register
Command Inhibit (CMD)

Normal Interrupt Status register
Command Complete
Error Interrupt Status (from Version 4.10)
Response error statuses related to Command Inhibit (CMD)
SW_RESET_FOR_ALL - Software Reset For All
This reset affects the entire Host Controller except for the card detection circuit. Register bits of type ROC, RW, RW1C, RWAC are cleared to 0. During its initialization, the Host Driver shall set this bit to 1 to reset the Host Controller. The Host Controller shall reset this bit to 0 when Capabilities registers are valid and the Host Driver can read them. Additional use of Software Reset For All may not affect the value of the Capabilities registers. If this bit is set to 1, the host driver should issue reset command and reinitialize the SD card.
DATA_TIMEOUT_COUNTER_VALUE - Data Timeout Counter Value
This value determines the interval by which DAT line timeouts are detected. For more information about timeout generation, refer to the Data Timeout Error in the Error Interrupt Status register. Timeout clock frequency will be generated by dividing the base clock TMCLK value by this value. When setting this register, prevent inadvertent timeout events by clearing the Data Timeout Error Status Enable (in the Error Interrupt Status Enable register)
1111b Reserved
1110b TMCLK x pow(2,27)
.....
0001b TMCLK x pow(2,14)
0000b TMCLK x pow(2,13)

There are two types of busy periods in a multiple block write operation.
(1) Write busy at block gap (without CMD12) is maximum 250ms
(2) Write busy after CMD12 is maximum 250ms (500ms for SDXC)

If CMD12 is issued during a multiple block write operation busy period, the host timeout counter is reset and the 250ms (500ms for SDXC) timeout period is measured from the response of CMD12.

The duration of an erase command can be estimated by the number of write blocks (WRITE_BL) to be
erased multiplied by 250 ms.

SDCLK_FREQUENCYSELECT - SDCLK Frequency Select
This register is used to select the frequency of SDCLK pin. The frequency is not programmed directly; rather this register holds the divisor of the Base Clock Frequency For SD Clock in the Capabilities register. Only the following settings are allowed.
UPPER_SDCLK_FREQUENCYSELECT - Upper Bits of SDCLK Frequency Select
Host Controller Version 1.00 and 2.00 do not support these bits and they are treated as 00b fixed value (ROC).
Host Controller Version 3.00 shall support these bits to expand SDCLK Frequency Select to 10-bit. Bit 07-06 is assigned to bit 09-08 of clock divider in SDCLK Frequency Select.

CLOCK_GENERATOR_SELECT - Clock Generator Select
Host Controller Version 3.00 supports this bit. This bit is used to select the clock generator mode in SDCLK Frequency Select.
If the Programmable Clock Mode is supported (non-zero value is set to Clock Multiplier in the Capabilities register), this bit attribute is RW, and if not supported, this bit attribute is RO and zero is read.
This bit depends on the setting of Preset Value Enable in the Host Control 2 register.
If the Preset Value Enable = 0, this bit is set by Host Driver.
If the Preset Value Enable = 1, this bit is automatically set to a value specified in one of Preset Value registers.
PLL Enable
This bit is added from Version 4.10 for Host Controller using PLL. This feature allows Host Controller to initialize clock generator in two steps: by Internal Clock Enable and PLL Enable and to minimize output latency (ex. SDCLK/RCLK, D0 lane) from SD Clock Enable. There are two modes to keep Host Drivers compatibility. In both modes, PL L Locked timing is indicated by Internal Clock Stable.

(1) When Host Version 4 Enable =0 (Host Driver Version 3, which does not support this bit) or this bit is not implemented, Internal Clock Enable (or SD Clock Enable) may activate PLL (exit low power mode and start locking clock).

(2) When Host Version 4 Enable =1 (Host Driver Version 4), Internal Clock Enable is set before setting this bit and then setting this bit may activate PLL (exit low power mode and start locking clock).
1 PLL is enabled
0 PLL is in low power mode
SD_CLK_EN
The Host Controller shall stop providing SDCLK or RCLK when writing this bit to 0. SDCLK/RCLK Frequency Select can be changed when this bit is 0. Then, the Host Controller shall maintain the same clock frequency until SDCLK is stopped (Stop at SDCLK=0). If the Card Inserted in the Present State register is cleared, this bit shall be cleared.

1 Enable providing SDCLK or RCLK
0 Disable providing SDCLK or RCLK

(1) SD Mode
This is the case when UHS-II Interface Enable is set to 0 in the Host Control 2 register. By setting this bit to 1, SDCLK is provided on pin number 5 (CLK). Refer to Section 1.12 Controlling SDCLK.
When PLL is used to generate clock, PLL is enabled by PLL Enable (if supported) or by SD Clock Enable (if PLL Enable is not supported).
When PLL is enabled by PLL Enable, the clock synchronization is checked by Internal Clock Stable.

INTERNAL_CLOCK_STABLE - Internal Clock Stable
As PLL Enable is added from Version 4.10, this status is expanded to check two cases. Host Driver Version 4.10 checks clock stability by this status twice after Internal Clock Enable is set and after PLL Enable is set.
Refer to Figure 3-3 in SD host spec4.1.

(1) Internal Clock Stable (when PLL Enable = 0 or not supported)
This bit is set to 1 when internal clock is stable after writing to Internal Clock Enable in this register to 1.

(2) PLL Clock Stable (when PLL Enable = 1)
Host Controller which supports PLL Enable sets this status to 0 once when PLL Enable is changed 0 to 1 and then this status is set to 1 when PLL is locked. (PLL uses an internal clock in stable as a reference clock which is enabled by Internal Clock Enable). After this bit is set to 1, Host Driver may set SD Clock Enable.

1 Ready
0 Not Ready

INTERNAL_CLOCK_EN - Internal Clock Enable
This bit is set to 0 when the Host Driver is not using the Host Controller or the Host Controller awaits a wakeup interrupt. The Host Controller should stop its internal clock to go very low power state. Still, registers shall be able to be read and written. Clock starts to oscillate when this bit is set to 1. When clock oscillation is stable, the Host Controller shall set Internal Clock Stable in this register to 1. This bit shall not affect card detection.

Offset: 0x2c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,x000,xxxx,0000,0000,0000,000x,0000)
PROD: 0x00000002 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	PROD	Description
26	RW	0x0	_NONE_	**SW_RESET_FOR_DAT_LINE**: **0** = WORK **1** = RESETED
25	RW	0x0	_NONE_	**SW_RESET_FOR_CMD_LINE**: **0** = WORK **1** = RESETED
24	RW	0x0	_NONE_	**SW_RESET_FOR_ALL**: **0** = WORK **1** = RESETED

Bit	R/W	Reset	PROD	Description
19:16	RW	0x0	_NONE_	**DATA_TIMEOUT_COUNTER_VALUE**
15:8	RW	0x0	_NONE_	**SDCLK_FREQUENCYSELECT**: **128** = DIV256 **64** = DIV128 **32** = DIV64 **16** = DIV32 **8** = DIV16 **4** = DIV8 **2** = DIV4 **1** = DIV2 **0** = BASE
7:6	RW	0x0	_NONE_	**UPPER_SDCLK_FREQUENCYSELECT**
5	RW	0x0	_NONE_	**CLOCK_GENERATOR_SELECT**
3	RW	0x0	_NONE_	**PLL_EN**: In legacy SD mode, no separate PLL is used to generate SDCLK. SW should set this as a part of standard SDCLK generation process. PLL is used to generate RCLK in UHS-II mode
2	RW	0x0	_NONE_	**SD_CLOCK_EN**: **0** = DISABLE **1** = ENABLE
1	RO	0x0	READY	**INTERNAL_CLOCK_STABLE**: **0** = NOT_READY **1** = READY
0	RW	0x0	_NONE_	**INTERNAL_CLOCK_EN**: when disabled turns off PLL in uhsII IOBRICK in uhsII mode **0** = STOP **1** = OSCILLATE

SDMMCA_INTERRUPT_STATUS_0

Normal Interrupt Status Register

VEND_SPEC_ERR[1:0]
1:BOOT_ACK_ERR - Occurs When Boot Ack Status is not equal to '010'
0:BOOT_ACK_TIMEOUT_ERR - Occurs When Boot Ack is not recieved within the programmed number of cycles.
TARGET_RESP_ERROR - Not supported for Tegra
SPI_ERR
Indicate when SPI Error has occured.The SPI Errors are registerd in SPI_INTERRUPT_STATUS register.
Response Error
Host Controller Version 4.00 supports response error check function to avoid overhead of response error check by Host Driver during DMA execution. If

Response Error Check Enable is set to1 in the Transfer Mode register, Host Controller Checks R1 or R5 response. If an error is detected in a response, this bit is set to 1.
TUNING_ERR
This bit is set when an unrecoverable error is detected in a tuning circuit except during tuning procedure (Occurrence of an error during tuning procedure is indicated by Sampling Select). By detecting Tuning Error, Host Driver needs to abort a command executing and perform tuning. To reset tuning circuit, Sampling Clock shall be set to 0 before executing tuning procedure (refer to Figure 2-29).
ADMA_ERR
This bit is set when the Host Controller detects errors during ADMA based data transfer. The state of the ADMA at an error occurrence is saved in the ADMA Error Status Register,
In addition, the Host Controller generates this Interrupt when it detects invalid descriptor data (Valid=0) at the ST_FDS state. ADMA Error State in the ADMA Error Status indicates that an error occurs in ST_FDS state. The Host Driver may find that Valid bit is not set at the error descriptor.
AUTO_CMD12_ERR
Auto CMD12 and Auto CMD23 use this error status. This bit is set when detecting that any of the bits D00 to D05 in Auto CMD Error Status register has changed from 0 to 1. D07 is effective in case of Auto CMD12. Auto CMD Error Status register is valid while this bit is set to 1 and may be cleared with clearing of this bit (another implementation is also allowed).
CURRENT_LIMIT_ERR
By setting the SD Bus Power bit in the Power Control register, the Host Controller is requested to supply power for the SD Bus. If the Host Controller supports the Current Limit function, it can be protected from an illegal card by stopping power supply to the card in which case this bit indicates a failure status. Reading 1 means the Host Controller is not supplying power to SD card due to some failure. Reading 0 means that the Host Controller is supplying power and no error has occurred. The Host Controller may require some sampling time to detect the current limit. If the Host Controller does not support this function, this bit shall always be set to 0.
DATA_END_BIT_ERR
Occurs either when detecting 0 at the end bit position of read data which uses the DAT line or at the end bit position of the CRC Status.
DATA_CRC_ERR
Occurs when detecting CRC error when transferring read data which uses the DAT line or when detecting the Write CRC status having a value of other than "010".
DATA_TIMEOUT_ERR
Occurs when detecting one of following timeout conditions.
(1) Busy timeout for R1b,R5b type
(2) Busy timeout after Write CRC status
(3) Write CRC Status timeout
(4) Read Data timeout.
COMMAND_INDEX_ERR

Occurs if a Command Index error occurs in the command response.
COMMAND_END_BIT_ERR
Occurs when detecting that the end bit of a command response is 0.
COMMAND_CRC_ERR
Command CRC Error is generated in two cases.
(1) If a response is returned and the Command Timeout Error is set to 0 (indicating no timeout), this bit is set to 1 when detecting a CRC error in the command response.
(2) The Host Controller detects a CMD line conflict by monitoring the CMD line when a command is issued. If the Host Controller drives the CMD line to 1 level, but detects 0 level on the CMD line at the next SDCLK edge, then the Host Controller shall abort the command (Stop driving CMD line) and set this bit to 1. The Command Timeout Error shall also be set to 1 to distinguish CMD line conflict
COMMAND_TIMEOUT_ERR
Occurs only if no response is returned within 64 SDCLK cycles from the end bit of the command. If the Host Controller detects a CMD line conflict, in which case Command CRC Error shall also be set as shown in Table 2-25, this bit shall be set without waiting for 64 SDCLK cycles because the command will be aborted by the Host Controller.
ERR_INTERRUPT
If any of the bits in the Error Interrupt Status register are set, then this bit is set. Therefore the Host Driver can efficiently test for an error by checking this bit first. This bit is read only.
FX_EVENT
This status is added from Version 4.10. Bit06 of response data will be stored in the R[14] of the Response register.
Basically, this interrupt is used with response check function. In this case, this status is set when R[14] of Response register is set to 1 and Response Type R1 / R5 is set to 0 in the Transfer Mode register or UHSII Transfer Mode register. If response check is disabled, this status is set when R[14] of Response register is set to 1. Host Driver needs to screen FX Event interrupt by checking response type is R1.

1 FX_EVENT is detected
0 No Event
RETUNING_EVENT - Re-Tuning Event
This status is set if Re-Tuning Request in the Present State register changes from 0 to 1.
Host Controller requests Host Driver to perform re-tuning for next data transfer. Current data transfer (not large block count) can be completed without re-tuning.
1 Re-Tuning should be performed
0 Re-Tuning is not required
CARD_INTERRUPT
Writing this bit to 1 does not clear this bit. It is cleared by resetting the SD card interrupt factor. In 1-bit mode, the Host Controller shall detect the Card Interrupt

without SD Clock to support wakeup. In 4-bit mode, the card interrupt signal is sampled during the interrupt cycle, so there are some sample delays between the interrupt signal from the SD card and the interrupt to the Host System. It is necessary to define how to handle this delay.
CARD_REMOVAL
This status is set if the Card Inserted in the Present State register changes from 1 to 0. When the Host Driver writes this bit to 1 to clear this status, the status of the Card Inserted in the Present State register should be confirmed. Because the card detect state may possibly be changed when the Host Driver clear this bit and interrupt event may not be generated.
CARD_INSERTION
This status is set if the Card Inserted in the Present State register changes from 0 to 1. When the Host Driver writes this bit to 1 to clear this status, the status of the Card Inserted in the Present State register should be confirmed. Because the card detect state may possibly be changed when the Host Driver clear this bit and interrupt event may not be generated.
BUFFER_READ_READY
This status is set if the Buffer Read Enable changes from 0 to 1. Refer to the Buffer Read Enable in the Present State register.
BUFFER_WRITE_READY
This status is set if the Buffer Write Enable changes from 0 to 1. Refer to the Buffer Write Enable in the Present State register.
DMA_INTERRUPT
This status is set if the Host Controller detects the Host DMA Buffer boundary during transfer. Refer to the Host DMA Buffer Boundary in the Block Size register. Other DMA interrupt factors may be added in the future. This interrupt shall not be generated after the Transfer Complete.
BLOCK_GAP_EVENT
If the Stop At Block Gap Request in the Block Gap Control register is set, this bit is set when both a read / write transaction is stopped at a block gap. If Stop At Block Gap Request is not set to 1, this bit is not set to 1.
XFER_COMPLETE
This bit indicates stop of transaction on three cases:
(1) Completion of a data transfer
(2) Completion of a command pairing with response-with-busy (R1b, R5b)
(3) Stop of data transfer by setting Stop At Block Gap Request in the Block Gap Control register
CMD_COMPLETE
This bit is set when get the end bit of the command response. (Except Auto CMD12)
Refer to Command Inhibit (CMD) in the Present State register.

Offset: 0x30
Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0x00,xxx0,0000,0000)

Bit	R/W	Reset	Description
31:30	RW	0x0	**VEND_SPEC_ERR**: **0** = DISABLE **3** = ENABLE
29	RW	0x0	**SPI_ERR**: **0** = NO_ERR **1** = ERR
28	RW	0x0	**TARGET_RESP_ERROR**: **0** = NO_ERROR **1** = ERROR
27	RW	0x0	**RESP_ERR**: **0** = NO_ERR **1** = ERR
26	RW	0x0	**TUNING_ERR**: **0** = NO_ERR **1** = ERR
25	RW	0x0	**ADMA_ERR**: **0** = NO_ERR **1** = ERR
24	RW	0x0	**AUTO_CMD12_ERR**: **0** = NO_ERR **1** = ERR
23	RW	0x0	**CURRENT_LIMIT_ERR**: **0** = NO_ERR **1** = POWER_FAIL
22	RW	0x0	**DATA_END_BIT_ERR**: **0** = NO_ERR **1** = ERR
21	RW	0x0	**DATA_CRC_ERR**: **0** = NO_ERR **1** = ERR
20	RW	0x0	**DATA_TIMEOUT_ERR**: **0** = NO_ERR **1** = TIMEOUT
19	RW	0x0	**COMMAND_INDEX_ERR**: **0** = NO_ERR **1** = ERR
18	RW	0x0	**COMMAND_END_BIT_ERR**: **0** = NO_ERR **1** = END_BIT_ERR_GENERATED
17	RW	0x0	**COMMAND_CRC_ERR**: **0** = NO_ERR **1** = CRC_ERR_GENERATED

Bit	R/W	Reset	Description
16	RW	0x0	**COMMAND_TIMEOUT_ERR**: **0** = NO_ERR **1** = TIMEOUT
15	RO	0x0	**ERR_INTERRUPT**: **0** = NO_ERR **1** = ERR
13	RO	0x0	**FX_EVENT**: **0** = NO_EVENT **1** = FX_EVENT_DETECTED
12	RO	0x0	**RETUNING_EVENT**: **0** = NO_INT **1** = GEN_INT
8	RO	0x0	**CARD_INTERRUPT**: **0** = NO_INT **1** = GEN_INT
7	RW	0x0	**CARD_REMOVAL**: **0** = NO_INT **1** = GEN_INT
6	RW	0x0	**CARD_INSERTION**: **0** = NO_INT **1** = GEN_INT
5	RW	0x0	**BUFFER_READ_READY**: **0** = NO_INT **1** = GEN_INT
4	RW	0x0	**BUFFER_WRITE_READY**: **0** = NO_INT **1** = GEN_INT
3	RW	0x0	**DMA_INTERRUPT**: **0** = NO_INT **1** = GEN_INT
2	RW	0x0	**BLOCK_GAP_EVENT**: **0** = NO_INT **1** = GEN_INT
1	RW	0x0	**XFER_COMPLETE**: **0** = NO_INT **1** = GEN_INT
0	RW	0x0	**CMD_COMPLETE**: **0** = NO_INT **1** = GEN_INT

SDMMCA_INTERRUPT_STATUS_ENABLE_0

Normal Interrupt Status Enable Register

This register is used to select which interrupt status is indicated to the Host System as the interrupt. These
status bits all share the same 1-bit interrupt line. Setting any of these bits to 1 enables interrupt generation.

Offset: 0x34
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,xx00,xxx0,0000,0000)

Bit	Reset	Description
31:30	0x0	**VENDOR_SPECIFIC_ERR**: **0** = DISABLE **3** = ENABLE
29	0x0	**SPI_ERR**: **0** = DISABLE **1** = ENABLE
28	0x0	**TARGET_RESP_ERROR**: **0** = NO_ERROR **1** = ERROR
27	0x0	**RESP_ERROR**: **0** = NO_ERROR **1** = ERROR
26	0x0	**TUNING_ERR**: **0** = DISABLE **1** = ENABLE
25	0x0	**ADMA_ERR**: **0** = DISABLE **1** = ENABLE
24	0x0	**AUTO_CMD12_ERR**: **0** = DISABLE **1** = ENABLE
23	0x0	**CURRENT_LIMIT_ERR**: **0** = DISABLE **1** = ENABLE
22	0x0	**DATA_END_BIT_ERR**: **0** = DISABLE **1** = ENABLE
21	0x0	**DATA_CRC_ERR**: **0** = DISABLE **1** = ENABLE
20	0x0	**DATA_TIMEOUT_ERR**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
19	0x0	**COMMAND_INDEX_ERR**: **0** = DISABLE **1** = ENABLE
18	0x0	**COMMAND_END_BIT_ERR**: **0** = DISABLE **1** = ENABLE
17	0x0	**COMMAND_CRC_ERR**: **0** = DISABLE **1** = ENABLE
16	0x0	**COMMAND_TIMEOUT_ERR**: **0** = DISABLE **1** = ENABLE
13	0x0	**FX_EVENT**: **0** = DISABLE **1** = ENABLE
12	0x0	**RETUNING_EVENT**: **0** = DISABLE **1** = ENABLE
8	0x0	**CARD_INTERRUPT**: **0** = DISABLE **1** = ENABLE
7	0x0	**CARD_REMOVAL**: **0** = DISABLE **1** = ENABLE
6	0x0	**CARD_INSERTION**: **0** = DISABLE **1** = ENABLE
5	0x0	**BUFFER_READ_READY**: **0** = DISABLE **1** = ENABLE
4	0x0	**BUFFER_WRITE_READY**: **0** = DISABLE **1** = ENABLE
3	0x0	**DMA_INTERRUPT**: **0** = DISABLE **1** = ENABLE
2	0x0	**BLOCK_GAP_EVENT**: **0** = DISABLE **1** = ENABLE
1	0x0	**TRANSFER_COMPLETE**: **0** = DISABLE **1** = ENABLE
0	0x0	**COMMAND_COMPLETE**: **0** = DISABLE **1** = ENABLE

SDMMCA_INTERRUPT_SIGNAL_ENABLE_0

Normal Interrupt Signal Enable Register

This register is used to select which interrupt status is notified to the Host System as the interrupt. These
status bits all share the same 1-bit interrupt line. Setting any of these bits to 1 enables interrupt generation.

Offset: 0x38
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,xx00,xxx0,0000,0000)

Bit	Reset	Description
31:30	0x0	**VENDOR_SPECIFIC_ERR**: **0** = DISABLE **3** = ENABLE
29	0x0	**SPI_ERR**: **0** = DISABLE **1** = ENABLE
28	0x0	**TARGET_RESP_ERROR**: **0** = NO_ERROR **1** = ERROR
27	0x0	**RESP_ERROR**: **0** = DISABLE **1** = ENABLE
26	0x0	**TUNING_ERR**: **0** = DISABLE **1** = ENABLE
25	0x0	**ADMA_ERR**: **0** = DISABLE **1** = ENABLE
24	0x0	**AUTO_CMD12_ERR**: **0** = DISABLE **1** = ENABLE
23	0x0	**CURRENT_LIMIT_ERR**: **0** = DISABLE **1** = ENABLE
22	0x0	**DATA_END_BIT_ERR**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
21	0x0	**DATA_CRC_ERR**: **0** = DISABLE **1** = ENABLE
20	0x0	**DATA_TIMEOUT_ERR**: **0** = DISABLE **1** = ENABLE
19	0x0	**COMMAND_INDEX_ERR**: **0** = DISABLE **1** = ENABLE
18	0x0	**COMMAND_END_BIT_ERR**: **0** = DISABLE **1** = ENABLE
17	0x0	**COMMAND_CRC_ERR**: **0** = DISABLE **1** = ENABLE
16	0x0	**COMMAND_TIMEOUT_ERR**: **0** = DISABLE **1** = ENABLE
13	0x0	**FX_EVENT**: **0** = DISABLE **1** = ENABLE
12	0x0	**RETUNING_EVENT**: **0** = DISABLE **1** = ENABLE
8	0x0	**CARD_INTERRUPT**: **0** = DISABLE **1** = ENABLE
7	0x0	**CARD_REMOVAL**: **0** = DISABLE **1** = ENABLE
6	0x0	**CARD_INSERTION**: **0** = DISABLE **1** = ENABLE
5	0x0	**BUFFER_READ_READY**: **0** = DISABLE **1** = ENABLE
4	0x0	**BUFFER_WRITE_READY**: **0** = DISABLE **1** = ENABLE
3	0x0	**DMA_INTERRUPT**: **0** = DISABLE **1** = ENABLE
2	0x0	**BLOCK_GAP_EVENT**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
1	0x0	**TRANSFER_COMPLETE**: **0** = DISABLE **1** = ENABLE
0	0x0	**COMMAND_COMPLETE**: **0** = DISABLE **1** = ENABLE

SDMMCA_AUTO_CMD12_ERR_STATUS_0

Host Control2 Register / Auto CMD Error Status Register

PRESET_VALUE_ENABLE - Preset Value Enable
Host Controller Version 3.00 supports this bit.
As the operating SDCLK frequency and I/O driver strength depend on the Host System implementation, it is difficult to determine these parameters in the Standard Host Driver. When Preset Value Enable is set, automatic SDCLK frequency generation and driver strength selection is performed without considering system specific conditions. This bit enables the functions defined in the Preset Value registers.
1 Automatic Selection by Preset Value are Enabled
0 SDCLK and Driver Strength are controlled by Host Driver
If this bit is set to 0, SDCLK Frequency Select, Clock Generator Select in the Clock Control register and Driver Strength Select in Host Control 2 register are set by Host Driver.
If this bit is set to 1, SDCLK Frequency Select, Clock Generator Select in the Clock Control register and Driver Strength Select in Host Control 2 register are set by Host Controller as specified in the Preset Value registers.
ASYNC_INTR_EN - Asynchronous Interrupt Enable
This bit can be set to 1 if a card support asynchronous interrupt and Asynchronous Interrupt Support is set to 1 in the Capabilities register. Asynchronous interrupt is effective when DAT[1] interrupt is used in 4-bit SD mode (and zero is set to Interrupt Pin Select in the Shared Bus Control register). If this bit is set to 1, the Host Driver can stop the SDCLK during asynchronous interrupt period to save power. During this period, the Host Controller continues to deliver Card Interrupt to the host when it is asserted by the Card.
1 Enabled
0 Disabled
ADDRESSING_64BIT_EN - 64bit addressing enable
Host Controller selects either of 32-bit or 64-bit addressing modes to access system memory. Whether 32-bit or 64-bit is determined by OS installed in a host system. Host Driver sets this bit depends on addressing mode of installed OS. Refer to 64-bit System Address Support in the Capabilities

register.
1 64 bits addressing
0 32 bits addressing
HOST_VERSION_4_EN - Host Version 4.00 Enable
This bit selects either Version 3.00 compatible mode or Ver4.00 mode. In Version 4.00, support of 64-bit System Addressing is modified. All DMAs support 64-bit System Addressing. UHS-II supported Host Driver shall enable this bit.
In Version 4.10, supported 32-bit Block Count for all operations.

Functions of following fields are modified.
(1) SDMA Address
SDMA uses ADMA System Address register (05Fh-058h) instead of SDMA System Address register (Offset 003-000h)
(2) ADMA2 / ADMA3 Selection
ADMA3/CQE is selected by DMA Select in the Host Control 1 register. This bit should be set to 1 to use ADMA3/CQE.
(3) 64bit ADMA Descriptor Size
128bit descriptor is used instead of 96-bit descriptor when 64-bit Addressing is set to 1.
(4) Selection of 32-bit / 64-bit System Addressing
Either 32-bit or 64-bit system addressing is selected by 64-bit Addressing bit in this register instead of DMA Select in the Host Control 1 register.
(5) 32-bit Block Count
SDMA System Address register (003h-000h) is modified to 32-bit Block Count register.

1 Version 4.00 Mode
0 Version 3.00 Compatible Mode
CMD23_EN
In memory card initialization, Host Driver Version 4.10 checks whether card supports CMD23 by checking a bit SCR[33]. If the card supports CMD23 (SCR[33]=1), this bit is set to 1. This bit is used to select Auto CMD23 orAuto CMD12 for ADMA3 data transfer. Refer to Auto CMD Enable in the Transfer Mode register.
ADMA2 Length Mode
This bit selects one of ADMA2 Length Modes either 16-bit or 26-bit.
1 26-bit Data Length Mode
0 16-bit Data Length Mode
UHS2_IF_EN - UHS-II Interface Enable
This bit is used to enable UHS-II Interface. Before trying to start UHS-II initialization, this bit shall be set to 1. SD 4-bit Interface signals shall be tri-state (input or bi-directional) or drive to low (output). Before trying to start SD mode initialization, this bit shall be set to.
1 UHS-II Interface Enabled
0 4-bit SD Interface Enabled
SAMPLING_CLK_SEL - Sampling Clock Select
Host Controller uses this bit to select sampling clock to receive CMD and

DAT. This bit is set by tuning procedure and valid after the completion of tuning (when Execute Tuning is cleared). Setting 1 means that tuning is completed successfully and setting 0 means that tuning is failed. Writing 1 to this bit is meaningless and ignored. A tuning circuit is reset by writing to 0. This bit can be cleared with setting Execute Tuning. Once the tuning circuit is reset, it will take time to complete tuning sequence. Therefore, Host Driver should keep this bit to 1 to perform re-tuning sequence to compete re-tuning sequence in a short time. Change of this bit is not allowed while the Host Controller is receiving response or a read data block. Refer to Figure 2-29.
1 Tuned clock is used to sample data
0 Fixed clock is used to sample data
EXECUTE_TUNING - Execute Tuning
This bit is set to 1 to start tuning procedure and automatically cleared when tuning procedure is completed. The result of tuning is indicated to Sampling Clock Select. Tuning procedure is aborted by writing 0. Refer to Figure 2-29 for more detail about tuning procedure.
1 Execute Tuning
0 Not Tuned or Tuning Completed
DRIVE_STRENGTH_SEL - Driver Strength Select
Host Controller output driver in 1.8V signaling is selected by this bit. In 3.3V signaling, this field is not effective. This field can be set depends on Driver Type A, C and D support bits in the Capabilities register.
This bit depends on setting of Preset Value Enable.
If Preset Value Enable = 0, this field is set by Host Driver.
If Preset Value Enable = 1, this field is automatically set by a value specified in the one of Preset Value registers.
00b Driver Type B is Selected (Default)
01b Driver Type A is Selected
10b Driver Type C is Selected
11b Driver Type D is Selected
VOLT_18_EN - 1.8V Signaling Enable
This bit controls voltage regulator for I/O cell. 3.3V is supplied to the card regardless of signaling voltage.
Setting this bit from 0 to 1 starts changing signal voltage from 3.3V to 1.8V. 1.8V regulator output shall be stable within 5ms. Host Controller clears this bit if switching to 1.8V signaling fails.
Clearing this bit from 1 to 0 starts changing signal voltage from 1.8V to 3.3V. 3.3V regulator output shall be stable within 5ms.
Host Driver can set this bit to 1 when Host Controller supports 1.8V signaling (One of support bits is set to 1: SDR50, SDR104 or DDR50 in the Capabilities register) and the card or device supports UHS-I (S18R=1. Refer to Bus Signal Voltage Switch Sequence in the Physical Layer Specification Version 3.00).
1 1.8V Signaling
0 3.3V Signaling
UHS_MODE_SEL - UHS Mode Select

This field is used to select one of UHS-I modes and effective when 1.8V Signaling Enable is set to 1.
If Preset Value Enable in the Host Control 2 register is set to 1, Host Controller sets SDCLK Frequency Select, Clock Generator Select in the Clock Control register and Driver Strength Select according to Preset Value registers. In this case, one of preset value registers is selected by this field. Host Driver needs to reset SD Clock Enable before changing this field to avoid generating clock glitch. After setting this field, Host Driver sets SD Clock Enable again.
000b SDR12
001b SDR25
010b SDR50
011b SDR104(SD/SDIO)/HS200(eMMC)
100b DDR50(SD/SDIO)/DDR52(eMMC)
101b HS400(eMMC)
110b Reserved
111b UHS2
When SDR50, SDR104 or DDR50 is selected for SDIO card, interrupt detection at the block gap shall not be used. Read Wait timing is changed for these modes. Refer to the SDIO Specification Version 3.00 for more detail
COMMAND_NOT_ISSUED - Command Not Issued By Auto CMD12 Error
Setting this bit to 1 means CMD_wo_DAT is not executed due to an Auto CMD12 Error (D04-D01) in this register.
INDEX_ERR - Auto CMD12 Index Error
This bit is set if the Command Index error occurs in response to a command.
END_BIT_ERR - Auto CMD12 End Bit Error
This bit is set when detecting that the end bit of command response is 0.
CRC_ERR - Auto CMD12 CRC Error
This bit is set when detecting a CRC error in the command response.
TIMEOUT_ERR - Auto CMD12 Timeout Error
This bit is set if no response is returned within 64 SDCLK cycles from the end bit of command. If this bit is set to 1, the other error status bits (D04-D02) are meaningless.
NOT_EXECUTED - Auto CMD12 Not Executed
If memory multiple block data transfer is not started due to command error, this bit is not set because it is not necessary to issue Auto CMD12. Setting this bit to 1 means the Host Controller cannot issue Auto CMD12 to stop memory multiple block data transfer due to some error. If this bit is set to 1, other error status bits (D04-D01) are meaningless.
The relation between Auto CMD12 CRC Error and Auto CMD12 Timeout Error is shown below

Auto CMD12 CRC Error	Auto CMD12 Timeout Error	Kinds of error
0	0	No Error
0	1	Response Timeout Error
1	0	Response CRC Error

| 1 | 1 | CMD line conflict |

RESP_ERR - Auto CMD Response Error
This bit is set when Response Error Check Enable in the Transfer Mode register is set to 1 and an error is detected in either R1 response of either Auto CMD12 or Auto CMD23.

1 Error
0 No Error

Offset: 0x3c
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,00x0,0000,0000,xxxx,xxxx,0x00,0000)

Bit	R/W	Reset	Description
31	RW	0x0	**PRESET_VALUE_ENABLE**: **0** = HW_SEL **1** = SW_SEL
30	RW	0x0	**ASYNC_INTR_EN**: **0** = DISABLE **1** = ENABLE
29	RW	0x0	**ADDRESSING_64BIT_EN**: **0** = DISABLE **1** = ENABLE
28	RW	0x0	**HOST_VERSION_4_EN**: **0** = DISABLE **1** = ENABLE
27	RW	0x0	**CMD23_EN**: **0** = DISABLE **1** = ENABLE
26	RW	0x0	**ADMA2_LEN_MODE**: **0** = LEN_16BIT **1** = LEN_26BIT
24	RW	0x0	**UHS2_IF_EN**: **0** = DISABLE **1** = ENABLE
23	RW	0x0	**SAMPLING_CLK_SEL**: **0** = FIXED **1** = TUNED
22	RW	0x0	**EXECUTE_TUNING**: **0** = NOT_TUNED **1** = EXECUTE

Bit	R/W	Reset	Description
21:20	RW	0x0	**DRIVE_STRENGTH_SEL**: **0** = TYPE_B **1** = TYPE_A **2** = TYPE_C **3** = TYPE_D
19	RW	0x0	**VOLT_18_EN**: **0** = V33 **1** = V18
18:16	RW	0x0	**UHS_MODE_SEL**: The following table shows the PROD mnemonic mapping to different SD and eMMC speed modes |-------------|---------------|---------------------| |PROD mnemonic | SD speed mode | eMMC speed mode | |-------------|---------------|---------------------| |prod_c_ds | DS | Legacy (SDR at 26M) | |prod_c_hs | HS | NA | |prod_c_sdr12 | SDR12 | NA | |prod_c_sdr25 | SDR25 | NA | |prod_c_sdr50 | SDR50 | SDR52 | |prod_c_sdr104 | SDR104 | NA | |prod_c_ddr52 | DDR50 | DDR52 | |prod_c_hs200 | NA | HS200 | |prod_c_hs400 | NA | HS400 | |---| **0** = SDR12 **1** = SDR25 **2** = SDR50 **3** = SDR104 **4** = DDR50 **5** = HS400 **6** = RSVD **7** = UHS2
7	RO	0x0	**COMMAND_NOT_ISSUED**: **0** = NO_ERR **1** = NOT_ISSUED
5	RO	0x0	**RESP_ERR**: **0** = NO_ERR **1** = ERR
4	RO	0x0	**INDEX_ERR**: **0** = NO_ERR **1** = ERR
3	RO	0x0	**END_BIT_ERR**: **0** = NO_ERR **1** = END_BIT_ERR_GENERATED
2	RO	0x0	**CRC_ERR**: **0** = NO_ERR **1** = CRC_ERR_GENERATED
1	RO	0x0	**TIMEOUT_ERR**: **0** = NO_ERR **1** = TIMEOUT
0	RO	0x0	**NOT_EXECUTED**: **0** = EXECUTED **1** = NOT_EXECUTED

SDMMCA_CAPABILITIES_0

Lower Capabilities Register

SLOT_TYPE
00b Removable Card Slot
01b Embedded Slot for One Device
10b Shared Bus Slot
11b UHS-II Multiple Embedded Devices
ASYNC_INTR
1 Asynchronous Interrupt Supported
0 Asynchronous Interrupt Not Supported
SYSTEM_BUS_64BIT_SUPPORT - 64-bit System Bus Support for V3
Meaning of this bit is different depends on Versions
Host Controller Version 3.00 and Ver4.10 use this bit as 64-bit System Address support for V3 mode.
Host Controller Version 4.00 uses this bit as 64-bit System Address support for both V3 and V4 modes.
SDMA cannot be used in 64-bit Addressing in Version 3 mode.
If this bit is set to 1, 64-bit ADMA2 with using 96-bit Descriptor may be enabled as follows:
In case of Host Controller Version 3, 64-bit ADMA2 is enabled by DMA Select =11b in the Host Control 1 register.
In case of Host Controller Version 4, 64-bit ADMA2 for Version 3 is enabled by setting Host Version 4 Enable =0 and DMA Select = 11b.

1 64-bit System Address for V3 is Supported
0 64-bit System Address for V3 is not Supported

SYSTEM_BUS_64BIT_SUPPORT_V4 - 64-bit System Bus Support for V4
This bit is added from Version 4.10. Setting 1 to this bit indicates that the Host Controller supports 64-bit System Addressing of Version 4 mode
When this bit is set to 1, full or a part of 64-bit address should be used to decode Host Controller Registers so that Host Controller Registers can be placed above system memory area. 64-bit address decode of Host Controller Registers is effective regardless of setting to 64bit Addressing in Host Control 2.
If this bit is set to 1, 64-bit DMA Addressing for Version 4 is enabled by setting Host Version 4 Enable =1,
64-bit Addressing =1 in the Host Control 2 register. SDMA can be used and ADMA2 uses 128-bit Descriptor.

1 64-bit System Address for V4 is Supported
0 64-bit System Address for V4 is not Supported

VOLTAGE_SUPPORT_1_8_V - Voltage Support 1.8V,The Voltage Support to Card is dependent on System & Slot.The platfrom datasheet has this.
VOLTAGE_SUPPORT_3_0_V - Voltage Support 3.0V,The Voltage Support to Card is dependent on System & Slot.The platfrom datasheet has this.
VOLTAGE_SUPPORT_3_3_V - Voltage Support 3.3V,The Voltage Support to Card is dependent on System & Slot.The platfrom datasheet has this.
SUSPEND_RESUME_SUPPORT - Suspend/Resume Support
This bit indicates whether the Host Controller supports Suspend / Resume functionality. If this bit is 0, the Host Driver shall not issue either Suspend or Resume commands because the Suspend and Resume mechanism (Refer to 1.6) is not supported.
DMA_SUPPORT - SDMA Support
This bit indicates whether the Host Controller is capable of using SDMA to transfer data between system memory and the Host Controller directly. Version 4.10 Host Controller shall support SDMA if ADMA2 is supported.
HIGH_SPEED_SUPPORT - High Speed Support
This bit indicates whether the Host Controller and the Host System support High Speed mode and they can supply SD Clock frequency from 25MHz to 50MHz.
ADMA1_SUPPORT - ADMA1 Support
This bit indicates whether the Host Controller is capable of using ADMA1.
ADMA2_SUPPORT - ADMA2 Support
This bit indicates whether the Host Controller is capable of using ADMA2. Version 4.10 Host Controller shall support ADMA2 if ADMA3 is supported.
EXTENDED_MEDIA_BUS_SUPPORT
Setting to 1,indicates 8-bit data bus is supported.
MAX_BLOCK_LENGTH - Max Block Length
This value indicates the maximum block size that the Host Driver can read and write to the buffer in the Host Controller. The buffer shall transfer this block size without wait cycles. Three sizes can be defined as indicated below. It is noted that transfer block length shall be always 512 bytes for SD Memory Cards regardless this field.
BASE_CLOCK_FREQUENCY - Base Clock Frequency For SD Clock
This value indicates the base (maximum) clock frequency for the SD Clock. Unit values are 1MHz. If the real frequency is 16.5MHz, the lager value shall be set 01 0001b (17MHz) because the Host Driver use this value to calculate the clock divider value (Refer to the SDCLK Frequency Select in the Clock Control register.) and it shall not exceed upper limit of the SD Clock frequency. The supported clock range is 10MHz to 63MHz. If these bits are all 0, the Host System has to get information via another method.
Not 0 - 1MHz to 63MHz
000000b- Get information via another method
TIMEOUT_CLOCK_UNIT - Timeout Clock Unit
This bit shows the unit of base clock frequency used to detect Data Timeout Error.
0 KHz
1 MHz
TIMEOUT_CLOCK_FREQUENCY - Timeout Clock Frequency

This bit shows the base clock frequency used to detect Data Timeout Error.
The Timeout Clock Unit defines the unit of this fields value.
Timeout Clock Unit =0 [KHz] unit: 1KHz to 63KHz
Timeout Clock Unit =1 [MHz] unit: 1MHz to 63MHz
Not 0 - 1KHz to 63KHz or 1MHz to 63MHz
000000b - Get information via another method

Offset: 0x40
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x3f6cd08c (0b0011,1111,011x,1100,1101,0000,1x00,1100)

Bit	Reset	Description
31:30	0x0	**SLOT_TYPE**: **0** = REMOVABLE **1** = EMBEDDED **2** = SHARED **3** = UHS2_MULTIPLE_EMBEDDED
29	0x1	**ASYNC_INTR**: **0** = NOT_SUPPORTED **1** = SUPPORTED
28	0x1	**SYSTEM_BUS_64BIT_SUPPORT**: **0** = NOT_SUPPORTED **1** = SUPPORTED
27	0x1	**SYSTEM_BUS_64BIT_SUPPORT_V4**: **0** = NOT_SUPPORTED **1** = SUPPORTED
26	0x1	**VOLTAGE_SUPPORT_1_8_V**: **0** = NOT_SUPPORTED **1** = SUPPORTED
25	0x1	**VOLTAGE_SUPPORT_3_0_V**: **0** = NOT_SUPPORTED **1** = SUPPORTED
24	0x1	**VOLTAGE_SUPPORT_3_3_V**: **0** = NOT_SUPPORTED **1** = SUPPORTED
23	0x0	**SUSPEND_RESUME_SUPPORT**: **0** = NOT_SUPPORTED **1** = SUPPORTED
22	0x1	**DMA_SUPPORT**: **0** = NOT_SUPPORTED **1** = SUPPORTED
21	0x1	**HIGH_SPEED_SUPPORT**: **0** = NOT_SUPPORTED **1** = SUPPORTED

Bit	Reset	Description
19	0x1	**ADMA2_SUPPORT**: **0** = NOT_SUPPORTED **1** = SUPPORTED
18	0x1	**EXTENDED_MEDIA_BUS_SUPPORT**: **0** = NOT_SUPPORTED **1** = SUPPORTED
17:16	0x0	**MAX_BLOCK_LENGTH**: **0** = BYTE512 **1** = BYTE1024 **2** = BYTE2048 **3** = RESERVED
15:8	0xd0	**BASE_CLOCK_FREQUENCY**
7	0x1	**TIMEOUT_CLOCK_UNIT**: MHz **0** = KHZ **1** = MHZ
5:0	0xc	**TIMEOUT_CLOCK_FREQUENCY**: 12MHz TMCLK is used in legacy SD/eMMC mode TMCLK freq value will be advertised based on UHS2_IF_EN and USE_TMCLK_FOR_DATA_TIMEOUT

SDMMCA_CAPABILITIES_HIGHER_0

Higher Capabilities Register

1.8V VDD2 Support
This bit indicates that support of VDD2 on the Host System.
0b 1.8V VDD2 is not supported
1b 1.8V VDD2 is supported
ADMA3_SUPPORT - ADMA3 Support
This bit indicates that support of ADMA3 on Host Controller.
0b ADMA3 is not supported
1b ADMA3 is supported
Clock Multiplier
This field indicates clock multiplier value of programmable clock generator.
Refer to Clock Control register. Setting 00h means that Host Controller
does not support programmable clock generator.
00h Clock Multiplier is Not Supported
01h Clock Multiplier M = 2
02h Clock Multiplier M = 3
....
Timer Count for Re-Tuning
0h Re-Tuning timer disabled
1h 1 seconds

2h 2 seconds
3h 4 seconds
. .....
. .....
nh 2**(n-1) seconds
. .....
. .....
Fh Get information from other source
Driver Type D Support
This bit indicates support of Driver Type D for 1.8 Signaling.
1 Driver Type D is Supported
0 Driver Type D is Not Supported
Driver Type C Support
This bit indicates support of Driver Type C for 1.8 Signaling.
1 Driver Type C is Supported
0 Driver Type C is Not Supported
Driver Type A Support
This bit indicates support of Driver Type A for 1.8 Signaling.
1 Driver Type A is Supported
0 Driver Type A is Not Supported
UHS2_SUPPORT - UHS-II Support (UHS-II only)
This bit indicates whether Host Controller supports UHS-II. If this bit is set to 1, 1.8V VDD2 Support shall be set to 1 (Host System shall support VDD2 power supply).
1 UHS-II is supported
0 UHS-II is not supported
DDR50 Support
1 DDR50 is Supported
0 DDR50 is Not Supported
SDR104 Support
SDR104 requires tuning.
1 SDR104 is Supported
0 SDR104 is Not Supported
SDR50 Support
If SDR104 is supported, this bit shall be set to 1. Bit 40 indicates whether SDR50 requires tuning or not.
1 SDR50 is Supported
0 SDR50 is Not Supported

Offset: 0x44
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x18002f73 (0bxxx1,1xxx,0000,0000,001x,1111,x111,0011)

Bit	Reset	Description
28	0x1	**VDD2_1_8V_SUPPORT**: **0** = NOT_SUPPORTED **1** = SUPPORTED
27	0x1	**ADMA3_SUPPORT**: **0** = NOT_SUPPORTED **1** = SUPPORTED
23:16	0x0	**CLOCK_MULTIPLIER**
15:14	0x0	**RETUNING_MODES**: **0** = MODE1 **1** = MODE2 **2** = MODE3 **3** = MODE4
13	0x1	**SDR50_TUNING**: **0** = NOT_REQUIRED **1** = REQUIRED
11:8	0xf	**RETUNING_TIMER_COUNT**
6	0x1	**TYPE_D_DRIVER**: **0** = NOT_SUPPORTED **1** = SUPPORTED
5	0x1	**TYPE_C_DRIVER**: **0** = NOT_SUPPORTED **1** = SUPPORTED
4	0x1	**TYPE_A_DRIVER**: **0** = NOT_SUPPORTED **1** = SUPPORTED
3	0x0	**UHS2_SUPPORT**: **0** = NOT_SUPPORTED **1** = SUPPORTED
2	0x0	**DDR50**: **0** = NOT_SUPPORTED **1** = SUPPORTED
1	0x1	**SDR104**: **0** = NOT_SUPPORTED **1** = SUPPORTED
0	0x1	**SDR50**: **0** = NOT_SUPPORTED **1** = SUPPORTED

SDMMCA_MAXIMUM_CURRENT_0

Maximum Current Capabilities Register

Offset: 0x48
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:16	0x0	**MAXIMUM_CURRENT_FOR_1_8V**: Maximum Current for 1.8V VDD1
15:8	0x0	**MAXIMUM_CURRENT_FOR_3_0V**: Maximum Current for 3.0V VDD1
7:0	0x0	**MAXIMUM_CURRENT_FOR_3_3V**: Maximum Current for 3.3V VDD1

SDMMCA_MAXIMUM_CURRENT_HI_0

Maximum Current Capabilities2 Register

Offset: 0x4c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**MAXIMUM_CURRENT_FOR_1_8V_VDD2**: Maximum Current for 1.V VDD2

SDMMCA_FORCE_EVENT_0

Force Event for Auto CMD12 Error Status Register

The Force Event Register is not a physically implemented register. Rather, it is an address at which the
Auto CMD12 Error Status Register can be written.
Writing 1 : set each bit of the Auto CMD12 Error Status Register
Writing 0 : no effect

Offset: 0x50
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,xxxx,xxxx,0x00,0000)

Bit	Reset	Description
31:30	0x0	**VENDOR_SPECIFIC_ERR_STATUS**: **0** = DISABLE **3** = ENABLE
28	0x0	**TARGET_RESP_ERROR**: **0** = NO_ERROR **1** = ERROR
27	0x0	**RESP_ERROR**: **0** = NO_ERROR **1** = ERROR
26	0x0	**TUNING_ERR**: **0** = DISABLE **1** = ENABLE
25	0x0	**ADMA_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
24	0x0	**AUTOCMD12_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
23	0x0	**CURRENTLIMIT_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
22	0x0	**DATA_END_BIT_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
21	0x0	**DATACRC_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
20	0x0	**DATATIMEOUT_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
19	0x0	**COMMAND_INDEX_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
18	0x0	**COMMAND_END_BIT_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
17	0x0	**COMMAND_CRC_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT

Bit	Reset	Description
16	0x0	**COMMAND_TIMEOUT_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
7	0x0	**AUTO_CMD12_NOT_ISSUED**: **0** = NO_INTERRUPT **1** = INTERRUPT
5	0x0	**AUTO_CMD12_RESP_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
4	0x0	**AUTO_CMD12_INDEX_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
3	0x0	**AUTO_CMD12_END_BIT_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
2	0x0	**AUTO_CMD12_CRC_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
1	0x0	**AUTO_CMD12_TIMEOUT_ERR**: **0** = NO_INTERRUPT **1** = INTERRUPT
0	0x0	**AUTO_CMD12_NOT_EXECUTED**: **0** = NO_INTERRUPT **1** = INTERRUPT

SDMMCA_ADMA_ERR_STATUS_0

ADMA Error Status Register

ADMA_LENGTH_MISMATCH_ERR - ADMA Length Mismatch Error
This error occurs in the following 2 cases.
(1) While Block Count Enable being set, the total data length specified by the Descriptor table is different from that specified by the Block Count and Block Length.
(2) Total data length can not be divided by the block length.
ADMA_ERR_STATE - ADMA Error State
This field indicates the state of ADMA when error is occurred during ADMA data transfer. This field never indicates "10" because ADMA never stops in this state.

|D01 - D00 |ADMA Error State when | Contents of SYS_SDR register |
| | error is occurred | |
---|
| 00 | ST_STOP (Stop DMA) | Points next of the error descriptor|
| 01 | ST_FDS (Fetch Descriptor) | Points the error descriptor |

| 10 | Never set this state | (Not used) |
| 11 | ST_TFR (Transfer Data) | Points the next of the error |
| | | descriptor |
----------|

Offset: 0x54
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**ADMA_LENGTH_MISMATCH_ERR**: **0** = NO_ERR **1** = ERR
1:0	0x0	**ADMA_ERR_STATE**

SDMMCA_ADMA_SYSTEM_ADDRESS_0

ADMA System Address Register
The 32-bit addressing Host Driver uses lower 32-bit of this register (upper 32-bit should be set to 0) and shall program Descriptor Table on 32-bit boundary and set 32-bit boundary address to this register. DMA2/3 ignores lower 2-bit of this register and assumes it to be 00b.
DMA in 64-bit addressing. The 64-bit addressing Host Driver uses all bits of this register and shall program Descriptor Table on 64-bit boundary and set 64-bit boundary address to this register. DMA2/3 ignores lower 3-bit of this register and assumes it to be 000b.
(1) SDMA
If Host Version 4.00 Enable is set to 1, SDMA use this register to indicate System Address of data location instead of using SDMA System Address register (Offset 003-000h). SDMA can be used in 32-bit and 64-bit addressing in Version 4.00.
(2) ADMA2
This register holds byte address of executing command of the Descriptor table. At the start of
ADMA2, the Host Driver shall set start address of the Descriptor table. The ADMA increments this register address, which points to next line, when every fetching a Descriptor line. When the ADMA Error Interrupt is generated, this register shall hold the Descriptor address depending on the ADMA state.
(3) ADMA3
This register is set by ADMA3. Host Driver is not necessary to set this register.

The ADMA3 increments address of this register, which points to next line, when every time fetching a Descriptor line. When Error Interrupt is generated, this register shall hold the Descriptor address depending on the ADMA state.

Offset: 0x58
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ADMA_SYSTEM_ADDRESS**

SDMMCA_UPPER_ADMA_SYSTEM_ADDRESS_0

Upper ADMA System Address Register

This register is used by 64-bit address descriptor. Upper bits of 64bit address - ADMA system address[63:32] is set in this register

Offset: 0x5c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**UPPER_ADMA_SYSTEM_ADDRESS**

SDMMCA_PRESET_DEFAULT_AND_INIT_0

Preset Value Register Indexs

15-14 HwInit Driver Strength Select Value
Driver Strength is supported by 1.8V signaling bus speed modes. This field is meaningless for 3.3V signaling.
11b Driver Type D is Selected
10b Driver Type C is Selected
01b Driver Type A is Selected

00b Driver Type B is Selected
13-11 Rsvd Reserved
Clock Generator Select Value
This bit is effective when Host Controller supports programmable clock generator.
1 Programmable Clock Generator
0 Host Controller Ver2.00 Compatible Clock Generator
SDCLK Frequency Select Value
10-bit preset value to set SDCLK Frequency Select in the Clock Control Register is described by a host system.
Preset Value for Default Speed and Initialization

Offset: 0x60
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00040000 (0b00xx,x000,0000,0100,00xx,x000,0000,0000)

Bit	Reset	Description
31:30	0x0	**DRIVE_STRENGTH_VAL_HIGH**
26	0x0	**CLK_GEN_VAL_HIGH**
25:16	0x4	**SDCLK_FREQ_VAL_HIGH**
15:14	0x0	**DRIVE_STRENGTH_VAL_LOW**
10	0x0	**CLK_GEN_VAL_LOW**
9:0	0x0	**SDCLK_FREQ_VAL_LOW**

SDMMCA_PRESET_SDR12_AND_HIGH_0

Preset Value for SDR12 Speed and HIGHSPEED

Offset: 0x64
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00040002 (0b00xx,x000,0000,0100,00xx,x000,0000,0010)

Bit	Reset	Description
31:30	0x0	**DRIVE_STRENGTH_VAL_HIGH**

Bit	Reset	Description
26	0x0	**CLK_GEN_VAL_HIGH**
25:16	0x4	**SDCLK_FREQ_VAL_HIGH**
15:14	0x0	**DRIVE_STRENGTH_VAL_LOW**
10	0x0	**CLK_GEN_VAL_LOW**
9:0	0x2	**SDCLK_FREQ_VAL_LOW**

SDMMCA_PRESET_SDR50_AND_SDR25_0

Preset Value for SDR50 and SDR25

Offset: 0x68
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00010002 (0b00xx,x000,0000,0001,00xx,x000,0000,0010)

Bit	Reset	Description
31:30	0x0	**DRIVE_STRENGTH_VAL_HIGH**
26	0x0	**CLK_GEN_VAL_HIGH**
25:16	0x1	**SDCLK_FREQ_VAL_HIGH**
15:14	0x0	**DRIVE_STRENGTH_VAL_LOW**
10	0x0	**CLK_GEN_VAL_LOW**
9:0	0x2	**SDCLK_FREQ_VAL_LOW**

SDMMCA_PRESET_DDR50_AND_SDR104_0

Preset Value for DDR50 and SDR104

Offset: 0x6c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00020000 (0b00xx,x000,0000,0010,00xx,x000,0000,0000)

Bit	Reset	Description
31:30	0x0	**DRIVE_STRENGTH_VAL_HIGH**
26	0x0	**CLK_GEN_VAL_HIGH**
25:16	0x2	**SDCLK_FREQ_VAL_HIGH**
15:14	0x0	**DRIVE_STRENGTH_VAL_LOW**
10	0x0	**CLK_GEN_VAL_LOW**
9:0	0x0	**SDCLK_FREQ_VAL_LOW**

SDMMCA_ADMA3_INT_DESC_LOWER_ADDRESS_0

ADMA3 Integrated Descriptor Address 31:0 Register .

The start address of Integrated DMA Descriptor is set to this register. Writing to a specific address starts ADMA3 depends on 32-bit/64-bit addressing. The ADMA3 fetches one Descriptor Address and increments this field to indicate the next Descriptor address.
The 32-bit addressing Host Driver uses lower 32-bit of this register and shall program Descriptor Table on 32-bit boundary. ADMA3 ignores lower 2-bit of this register and assumes it to be 00b. Writing to 07Bh starts ADMA3 data transfer.
The 64-bit addressing Host Driver uses all 64-bit of this register and shall program Descriptor Table on 64-bit boundary. ADMA3 ignores lower 3-bit of this register and assumes it to be 000b. Writing to 07Fh starts ADMA3 data transfer.

Register Value Addressing Mode
00000000_xxxxxxxxh 32-bit System Address
xxxxxxxx_xxxxxxxxh 64-bit System Address

Offset: 0x78
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ADDRESS_31_0**

SDMMCA_ADMA3_INT_DESC_UPPER_ADDRESS_0

ADMA3 Integrated Descriptor Address 63:32 Register

Offset: 0x7c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ADDRESS_63_32**

SDMMCA_VENDOR_REGS_PTR_0

Pointer Registers to 1FFh-100h Area - vendor specific area
Area of offset mFFh-m00h is defined as re-locatable area. The locations of following register sets are pointed by offset address.
vendor registers start at 100h. => m=1
Vendor registers pointer

Offset: 0xe8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000100 (0bxxxx,xxxx,xxxx,xxxx,xxxx,0001,0000,0000)

Bit	Reset	Description
11:0	0x100	**VENDOR_PTR**: Vendor regs start at 0x100 - offset[11:0]=0x100

SDMMCA_SLOT_INTERRUPT_STATUS_0

Slot Interrupt Status Register

VENDOR_VERSION_NUMBER - Vendor Version Number
This status is reserved for the vendor version number. The Host Driver

should not use this status.
SPECIFICATION_VERSION_NUMBER - Specification Version Number
This status indicates the Host Controller Spec. Version. The upper and lower 4-bits indicate the version.
00 SD Host Specification Version 1.00
01 SD Host Specification Version 2.00
Including the feature of the ADMA and Test Register,
02 SD Host Specification Version 3.00
03 SD Host Specification Version 4.00
04h SD Host Controller Specification Version 4.10
05h SD Host Controller Specification Version 4.20
others Reserved
INTERRUPT_SIGNAL_FOR_EACH_SLOT - Interrupt Signal For Each Slot
These status bits indicate the logical OR of Interrupt Signal and Wakeup Signal for each slot. A maximum of 8 slots can be defined. If one interrupt signal is associated with multiple slots, the Host Driver can know which interrupt is generated by reading these status bits. By a power on reset or by setting Software Reset For All, the interrupt signal shall be de-asserted and this status shall read 00h.
Bit 00 Slot 1
Bit 01 Slot 2
Bit 02 Slot 3
....
Bit 07 Slot 8

Offset: 0xfc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x05050000 (0b0000,0101,0000,0101,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31:24	0x5	**VENDOR_VERSION_NUMBER**
23:16	0x5	**SPECIFICATION_VERSION_NUMBER**
7:0	0x0	**INTERRUPT_SIGNAL_FOR_EACH_SLOT**

SDMMCA_VENDOR_CLOCK_CNTRL_0

The following Registers are Vendor Specific Registers and are mapped to Vendor Specific Address Space(0x100 - 0x1FF)

Offset: 0x100
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0804d06d (0bxx00,1000,0000,0100,1101,0000,x110,1101)
PROD: 0x0e080020 (0bxxx0,1110,0000,1000,xxxx,xxxx,x01x,0x0x)

Bit	Reset	PROD	Description
29	0x0	_NONE_	**DIFF_CLK_SEL**: If set, selects differential CLK and DQS; Used by eMMC IOBRICK only. SW should set this appropriately based on eMMC part used. E_INPUT_* (CMD/DAT/CLK) of emmc IOBRICK should be set to 0 when differential signalling is used. default is '0' - selects single ended signalling for clk/dqs
28:24	0x8	0xe	**TRIM_VAL**: Trimmer tap value for the output data path trimmer This determines the trimmer value needed to drive the output data correctly. The tap for outbound trimmer is single MUX. The trim settings required are within very small (0-3) with absolute delay requirement of ~400ps. Minimal change with PVT variations.
23:16	0x4	0x8	**TAP_VAL**: Tap value for input data path trimmer This determines the tap value needed to sample the input data correctly. Delay per each tap can range from 70ps (hv_ff) to 505ps (lv_ss).
15:8	0xd0	_NONE_	**BASE_CLK_FREQ**: SW driver should write core clock frequency value in MHz to this field to advertise base frequency in SDMMCA_CAPABILITIES_0_ BASE_CLOCK_FREQUENCY for standard SD driver usage.
6	0x1	NORMAL	**LEGACY_CLKEN_OVERRIDE**: Override for sdmmc_legacy_g_clk clken; Set this to 0 to save power **0** = NORMAL :0 -> sdmmc_legacy_g_clk is gated **1** = OVERRIDE :1 -> sdmmc_legacy_g_clk is not gated
5	0x1	OVERRIDE	**SDR50_TUNING_OVERRIDE**: override the SDR50_TUNING capabilities bit. Software should only set this bit if it is required to use Tuning for SDR50. (only supported for SDMMC1) **0** = NORMAL :0 -> No Tuning support advertised for SDR50 mode. **1** = OVERRIDE :1 -> Tuning support is enabled for SDR50 mode.

Bit	Reset	PROD	Description
4	0x0	_NONE_	**UHS2_CAPABILITY_OVERRIDE**: override the UHS-II capabilities bit.
3	0x1	0x0	**PADPIPE_CLKEN_OVERRIDE**: Override for padmacro and pipemacro clken. **0** = NORMAL :0 -> CLKEN is de-asserted when internal CLKEN is de-asserted. **1** = OVERRIDE :1 -> CLKEN is kept asserted even when internal CLKEN is de-asserted.
2	0x1	_NONE_	**SPI_MODE_CLKEN_OVERRIDE**: This mode is not supported in Tegra.
1	0x0	FEEDBACK	**INPUT_IO_CLK**: Feedback clock is selected by default. Software should not change this. Disabling Feedback clock will select Internal Clock that requires different TAP Value Programming. **0** = FEEDBACK **1** = INTERNAL
0	0x1	_NONE_	**SDMMC_CLK**: This is set when sdmmc_clk is supplied by the CAR module. Prior to sdmmc_clk switch OFF, this bit should be written as '0'. By writing zero,the asynchronous card interrupt is routed to the Interrupt controller. **0** = DISABLE **1** = ENABLE

SDMMCA_VENDOR_SYS_SW_CNTRL_0

Offset: 0x104
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x38600002 (0b0011,1000,011x,xxxx,x000,0000,x000,0010)

Bit	Reset	Description
31	0x0	**ENHANCED_STROBE_MODE**: Enables enhanced strobe mode in HS400 mode for eMMC5.x devices 0 - cmd_in(Resp) is sampled by loopback clock which requires tuning 1 - cmd_in(Resp) is sampled by DQS_in which requires no tuning SW has to set this bit appropriately based on device capability since this is an optional feature for eMMC5.x devices. If device supports this feature, SW should set this bit to avoid tuning.

Bit	Reset	Description
30	0x0	**USE_TMCLK_FOR_WR_CRC_STATUS_TIMEOUT**: When set, uses TMCLK data timeout counter for generating wr_crc_status data-timeout When cleared, uses sdmmc_clk for maintaining wr_crc_status data timeout counter
29	0x1	**USE_NCRC_FOR_WR_CRC_STATUS_TIMEOUT_VAL**: This field is valid only when USE_TMCLK_FOR_WR_CRC_STATUS_TIMEOUT is set to 0. When cleared, uses data timeout value as wr crc status timeout value (spec defined one) When set, uses Ncrc cycles as timeout value
28	0x1	**USE_TMCLK_FOR_DATA_TIMEOUT**: When set, uses TMCLK data timeout counter for generating legacy data timeout error (except wr_crc_status timeout) When cleared, uses sdmmc_clk for maintaining data timeout counter
27:24	0x8	**DEVICE_BUSY_WAIT_CYCLES**: This register field is used to load wait_cycles counter before device busy sampling in HS400 mode. Please note that this counter is used only in HS400 mode.
23	0x0	**ALLOW_CARD_CLK_STALLS_IN_WR**: When set, allows card clock stopping during transfer of data within a block in DDR52/HS400 writes.
22	0x1	**EMMC_IOBRICK_CLK_DATA**: Used to drive AP_CLK and AN_CLK input of iobrick. 0x1 - clk_out will be same as iobrick_clk_in 0x0 - clk_out will be inverted iobrick_clk_in
21	0x1	**QUALIFY_WITH_RD_DATA_VLD**: We have async FIFOs in both cmd_in and dat_in paths in padmacro which are used in tunable modes. When this bit set, rdata from FIFO is treated as valid data only when rd_req is high. This is needed to handle bubbles on 'rd_req' when MTBF is high.
14	0x0	**SD_BUS_POWER_ON_OFF_INT_STATUS**: SD_BUS_POWER was changed. System software can use this interrupt to implement power switch.
13	0x0	**VOLT_SWITCH_INT_STATUS**: VOLT_18_EN was changed. System software can use this interrupt to implement a UHS-I voltage switch procedure for a standard SD Host driver **0** = NO_INT **1** = GEN_INT
12	0x0	**TUNING_SYS_INT_STATUS**: CMD19 was issued while EXECUTE_TUNING was set. System software can use this interrupt to implement a UHS-I tuning procedure for a standard SD Host driver. **0** = NO_INT **1** = GEN_INT

Bit	Reset	Description
11:8	0x0	**TUNING_ASYNC_FIFO_ADDNL_DELAY**: This register field holds the additional delay in cycles which should be added to round trip delay Default value is - zero. SW should not update this field unless a new PROD setting is given.
6	0x0	**SD_BUS_POWER_ON_OFF_INT_ENABLE**: Enables a seperate system interrupt (i.e., not the standard SDHCI interrupt) when SD_BUS_POWER is changed. **0** = DISABLE **1** = ENABLE
5	0x0	**VOLT_SWITCH_INT_ENABLE**: Enables a seperate system interrupt (i.e., not the standard SDHCI interrupt) when VOLT_18_EN is changed. **0** = DISABLE **1** = ENABLE
4	0x0	**TUNING_SYS_INT_ENABLE**: Enables a seperate system interrupt (i.e., not the standard SDHCI interrupt) when CMD19 is issued while EXECUTE_TUNING is set. **0** = DISABLE **1** = ENABLE
3	0x0	**ASSERT_BUFF_RD_RDY_INT**: Write a 1 to this field to assert sdmmc_interrupt_status_0_buffer_read_ready. Used by the system software that implements the tuning procedure to signal to the standard SD driver that the tuning process has completed **0** = DISABLE **1** = ENABLE
2	0x0	**IO_TRIM_BYPASS**: Override bit for selecting between core trimmer (Vcore dependent) and io trimmer (custom trimmer) in IB clock path Default option is IO trimmer; SW should not set this field.
1	0x1	**INT_MASK_WHILE_TUNING**: As per spec, Host should not generate any interrupts (including cmd_complete and data_xfer_complete) except buffer_read_ready interrupt during tuning sequence is being performed SW can override this behavior by clearing this bit - but this leads to a spec violation **0** = DISABLE **1** = ENABLE
0	0x0	**SPI_MODE**: This mode is not supported in Tegra.

SDMMCA_VENDOR_ERR_INTR_STATUS_0

Legacy Interrupt Status Register

The fields are valid when a error interrupt has occurred.

Offset: 0x108
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000X0000 (0bxxxx,xxxx,xxxx,xx00,0000,xxx0,0000,0000)

Bit	Reset	Description
18	X	**SDMMC_LEGACY_CTLR_IDLE**: indicates legacy SD interface controller is idle - no active data transfers on legacy SD interface
17	0x0	**READ_DATA_TIMEOUT**: valid when a data timeout error occurs
16	0x0	**WRITE_CRC_STATUS_TIMEOUT**: valid when a data timeout error occurs
15	0x0	**WRITE_BUSY_TIMEOUT**: valid when a data timeout error occurs
14	0x0	**RESP_BUSY_TIMEOUT**: valid when a data timeout error occurs
13	0x0	**SPI_WRITE_BUSY_TIMEOUT**
12	0x0	**SPI_RX_START_TOKEN_TIMEOUT**
8:5	0x0	**SPI_DAT_ERR_TOKEN**: Data Error Token,while read from card.
4:0	0x0	**SPI_DAT_RESPONSE**: Data Response while write to card **5** = DATA_ACCEPTED **11** = CRC_ERR **13** = WRITE_ERR

SDMMCA_VENDOR_CAP_OVERRIDES_0

Capabilities override bits

Offset: 0x10c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00001107 (0bxxxx,xxxx,xxxx,xxxx,xx01,0001,xxxx,0111)

Bit	Reset	Description
13:8	0x11	**DQS_TRIM_VAL**: Tap value for incoming DQS path trimmer - used in HS400 modes
3	0x0	**DRV_LPBK_CLK_ON_CMD_LINE**: Loopback trimmed clock will be driven onto cmd line, if this bit set to 1. Should be set to zero during normal data transfers. Useful in debug.
2	0x1	**VOLTAGE_3_3_V_SUPPORT_OVERRIDE**: Voltage support 3_3_V override
1	0x1	**VOLTAGE_3_0_V_SUPPORT_OVERRIDE**: Voltage support 3_0_V override
0	0x1	**VOLTAGE_1_8_V_SUPPORT_OVERRIDE**: Voltage support 1_8_V override

SDMMCA_VENDOR_DEBOUNCE_COUNT_0

Debounce Counter Value Register

The Debounce Counter runs on 32 KHz clock. Keeping the default value to 100ms = (100 * 32cycles/1ms) = 3200 cycles for 100ms = 0xC80

Offset: 0x11c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000c80 (0bxxxx,xxxx,0000,0000,0000,1100,1000,0000)

Bit	Reset	Description
23:0	0xc80	**VALUE**: The number of 32KHz clock cycles is programed to meet Debounce period of the card slot.This register is valid for only SDMMC1.

SDMMCA_VENDOR_MISC_CNTRL_0

Misc Vendor Cntrl Register
SDMMC_SPARE0: Spare register bits with reset value of 0
SDMMC_SPARE0[0] : SW_RESET_CLKEN_OVERRIDE, override the sdmmc_clken when doing SW_RESET if set to 1.
SDMMC_SPARE0[1] : When set, allows SD clock to be stopped in the middle of a read data block while in SDR104/HS400 modes(allow_sdr104_intrablock_stalls).
Unsafe for some SD/eMMC cards, but may improve SDR104 DMA read performance in some cases.

SDMMC_SPARE0[2] : When set, SDR104 support is advertised in SDMMC_CAPABILITIES_HIGHER_0_SDR104
SDMMC_SPARE0[3] : When set, SDR50 support is advertised in SDMMC_CAPABILITIES_HIGHER_0_SDR50
SDMMC_SPARE0[5] : When 0, masks the pad macro's "high speed" enable to 0, causing the pad macro to always launch
data on the falling edge of the clock. This prevents the SD Host driver's setting of SDMMC_POWER_CONTROL_HOST_x_HIGH_SPEED_EN from undesireably affecting the output timing.
SDMMC_SPARE0[7:6] : Number of pipe stages.
SDMMC_SPARE0[8] : When set, DDR50 support is advertised in SDMMC_CAPABILITIES_HIGHER_0_DDR50.
SDMMC_SPARE1: Spare register bits with reset value of 1
SDMMC_SPARE1[0] : Reserved
SDMMC_SPARE1[1] : Reserved

Offset: 0x120
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffff0098 (0b1111,1111,1111,1111,0000,0000,1001,1000)

Bit	Reset	Description
31:16	0xffff	**SDMMC_SPARE1**: Spare register bits with reset value of 1
15:1	0x4c	**SDMMC_SPARE0**: Spare register bits with reset value of 0x4C
0	0x0	**ERASE_TIMEOUT_LIMIT**: Erase timeout value. **0** = FINITE :Finite,It is limited to the programmed value in the DATA_TIMEOUT_VALUE **1** = INFINITE :Infinite,Controller would be monitoring until the card is busy.

SDMMCA_VENDOR_MISC_CNTRL1_0
Offset: 0x124
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:16	0x0	**OVERRIDE_FOR_1_8V**: Maximum override for 1.8V VDD1
15:8	0x0	**OVERRIDE_FOR_3_0V**: Maximum override for 3.0V VDD1
7:0	0x0	**OVERRIDE_FOR_3_3V**: Maximum override for 3.3V VDD1

SDMMCA_VENDOR_MISC_CNTRL2_0

Offset: 0x128
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x62600000 (0b0110,001x,x11x,0000,0000,0000,0000,0000)
PROD: 0x00000000 (0bx0xx,xx0x,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	PROD	Description
31	0x0	_NONE_	**VGPIO_MODE_EN**: SW should set this to 1 when CD and WP pins are connected to external PMIC/VGPIO and not connected to SoC pins
30	0x1	0x0	**SDMMC_CLK_OVR_ON**: Master clk en Override bit for all SLCGs
29	0x1	_NONE_	**SD_CARD_DETECT_STATUS_N**: SW should read CD_N status from external PMIC/VGPIO controller and update this field to get SDMMC present state register gets updated. Present state register is read by standard SDHC driver to know card status. 0 - card detected 1 - no card present in slot
28	0x0	_NONE_	**SD_CARD_WP_STATUS**: SW should read Write Protect status from external PMIC/VGPIO controller and update this field to get SDMMC present state register gets updated. Present state register is read by standard SDHC driver to know card status. 0 - card is not write protected 1 - card is write protected
27	0x0	_NONE_	**CMD_TFIFO_HOT_RESET**: SW can reset CMD tuning FIFO present in padmacro incase there is any error or hang condition. Reset duration should be atleast 20 cycles. Set this bit to 1 and clear it after 20 cycles to reset FIFO.

Bit	Reset	PROD	Description
26	0x0	_NONE_	**DAT_TFIFO_HOT_RESET**: SW can reset DAT tuning FIFO present in padmacro incase there is any error or hang condition. Reset duration should be atleast 20 cycles. Set this bit to 1 and clear it after 20 cycles to reset FIFO.
25	0x1	NORMAL	**ADMA3_CLKEN_OVERRIDE**: Override for sdmmc_adma3_g_clk clken; **0** = NORMAL :0 -> sdmmc_adma3_g_clk is gated in nonADMA3 modes **1** = OVERRIDE :1 -> sdmmc_adma3_g_clk is not gated in nonADMA3 modes
22	0x1	_NONE_	**ADMA3_DESC_PREFETCH_EN**: When set to 1, enables ADMA3 descriptors pre-fetch feature. It helps improving perf when system memory is heavily fragmented (number of descriptors programmed per transfer will be more)
21	0x1	_NONE_	**ADMA2_DESC_PREFETCH_EN**: When set to 1, enables ADMA2 descriptors pre-fetch feature. It helps improving perf when system memory is heavily fragmented (number of descriptors programmed per transfer will be more)
19:16	0x0	_NONE_	**DATA_TIMEOUT_VAL_MULTIPLIER**: Used when SDMMC IO clock is used instead TMCLK for running data timeout counter (USE_TMCLK_FOR_DATA_TIMEOUT is set in VENDOR_SYS_SW_CNTRL register). Effective data timeout val = (multiplier+1) * data_timeout_val 0 - no multiplier
15:12	0x0	_NONE_	**DAT_TUNING_ASYNC_FIFO_ADDNL_DELAY**: This register field holds the additional delay in cycles which should be added to wdata/crc token round trip delay Default value is - zero. NOTE: SW should not update this field unless a new PROD setting is given.
11:8	0x0	_NONE_	**ADDITIONAL_NCR_CYCLES**: Additional Ncr wait time - useful for HW debug Default is 0 - SW should not modify this.
7:0	0x0	_NONE_	**OVERRIDE_FOR_1_8V_VDD2**: Maximum override for 1.8V VDD2

SDMMCA_VENDOR_IO_TRIM_CNTRL_0

Vendor IO trimmer control register
Used to configure IO trimmer
Truth table
Input Pins Output Comments

E_DPD SEL_VREG SEL_VREF CLKOUT
1 x x 0 The cell is in deep power down mode
0 1 x based on ip_clk_select selected clock input Trimmer is powered by VAUXC
0 0 x based on ip_clk_select selected clock input Trimmer is powered by regulated voltage
* 'x' indicates don't care

Offset: 0x1ac
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000205 (0b0000,xxxx,xxxx,xxxx,xxxx,x010,000x,x101)

Bit	Reset	Description
31:28	0x0	**TRIM_PAD_RFU_IN**: Unused comp pad input pins. Reserved for future usage.
10:8	VREF_750_MV	**SEL_VREF_LEVEL**: Selects Vref voltage level **0** = VREF_700_MV **1** = VREF_725_MV **2** = VREF_750_MV **3** = VREF_775_MV **4** = VREF_800_MV **5** = VREF_850_MV **6** = VREF_900_MV **7** = VREF_950_MV
7:6	0x0	**TRIM_SEL_ATEST**: For testing purpose only - should be used when sdmmc is in idle state Select analog test signals to send to comp pad. 0x0: Not used (float) 0x1: Regulator input voltage 0x2: Regulator output voltage before analog mux 0x3: Regulator output after analog mux
5	ENABLE	**TRIM_PWRSAVE**: Enables power saving mode by clock gating the unused taps in delay chain Active low signal, 0 - power saving mode enabled - clock gating is enabled for unused trimmer taps - may affect tap delay 1 - no power saving - all the trimmer taps are not clock gated **0** = ENABLE **1** = DISABLE

Bit	Reset	Description
2	0x1	**SEL_VREG**: By default, BG is disabled to save power if interface is not used. SW should select BG for error free SD/eMMC operation. For BG <-> VAUXC switching, SW should follow the switching sequence given in TRM/IAS. Select voltage supply for delay chain present in both Trimmer and DLLs PROD value: 0x0 ***SW should set this to 0x0 before accessing SD/eMMC. This setting makes IB trimmer delay independent of VDD_CORE*** 0 - selects regulated reference voltage for trimmer supply - default (recommended option for tunable SD/eMMC modes) 1 - selects VAUXC for trimmer supply and shut down BG+REG circuit (can be used in nonTunable modes for power saving) Power up time for BG+REG is ~3us(worst case). Power down time for BG+REG is ~1us(worst case). If SW wants to turn on/off BG+REG when SDMMC is idle, it has to take hit of 3us power on time. When SEL_VREG is toggled, both DLL and rx clock trimmer output could glitch irrespective of input clock state which could cause corresponding rx CMD and DATA FIFOs to go into bad state. SW should issue SW_RESET_DAT and SW_RESET_CMD to reset host FIFOs after BG <-> VAUXC switching. This would ensure error free data tranfers from there on. Powering down BG would need 3usec turn ON time which may cause IOPS reduction, if SW shut downs BG after every transfer and enables it on seeing new xfer req. Hence, it may not be possible to do dynamic shut down of BG without stalling new requests.
1	0x0	**SEL_VREF**: Select reference voltage for voltage regulator 0 - selects Bandgap Voltage Reference (recommended option for SD/eMMC tunable modes) 1 - selects resistor divider voltage reference and power down bandgap Switching time between the supplies is 1us. When switching from one supply to other supply, we need to wait for atleast 1us before doing any data transfers. Providing reference voltage from R divider network is just a backup plan, if A. Bandgap does not work or B. Bandgap works very well but we want to save bandgap power when Silicon Characterization results shows that the eMMC/ SDMMC interface perform well even by using R divider+REG+TRIMMER

Bit	Reset	Description
0	0x1	**PD_BGREG**: Not used - Dummy control Power down Band Gap voltage reference, voltage regulator and resistor chain voltage ref (BG+REG) present in custom IO trimmer used for SD/eMMC bus tuning. Active High signal, PD_BGREG=1 => Power down BG+REG; PD_BGREG=0 => power up Power down and up time for BG+REG is 1us. If SW wants to turn on/off BG+REG when SDMMC is idle, it has to follow 1us power on/off time. Back-up option for powering down BG. SW should clear this bit when it wants to turn ON BG by setting SEL_VREG=0. The original idea to have PD_BGREG pin is to provide power saving feature when the eMMC/SDMMC is in IDLE state, but not in DPD mode. This pin function is actually merged into SEL_VREG function -BG+REG circuit is shut-down when SEL_VREG=1 (trimmer powered by VAUXC)

SDMMCA_VENDOR_TUNING_CNTRL0_0

Vendor Tuning Control0 register

Offset: 0x1c0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x74020090 (0bx111,0100,0000,001x,0000,0000,1001,0000)
PROD: 0x00000040 (0bxxxx,xxxx,xxxx,xxxx,xxx0,0000,01xx,xxxx)

Bit	Reset	PROD	Description
30	0x1	_NONE_	**RD_DATA_CRC_ERR_EN**: If cleared, Re-tuning request/tuning error is not generated on read data crc error
29	0x1	_NONE_	**WR_DATA_CRC_ERR_EN**: If cleared, Re-tuning request/tuning error is not generated on write crc error
28	0x1	_NONE_	**CMD_CRC_ERR_EN**: If cleared, Re-tuning request/tuning error is not generated on cmd crc error
27	0x0	_NONE_	**RETUNING_REQ_EN_ON_CRC_ERR_DETECTION**: Re-tuning request is generated when set to initiate re-tuning by SW to compensate for temperature drift when any data or cmd CRC errors are detected in SDR50/SDR104/HS200 modes

Bit	Reset	PROD	Description
26	0x1	_NONE_	**TUNING_ERR_EN_ON_CRC_ERR_DETECTION**: Tuning error is generated to initiate re-tuning by SW to compensate for temperature drift when any data or cmd CRC errors are detected in SDR50/SDR104/HS200 modes
25:18	0x0	_NONE_	**START_TAP_VAL**: start tap value to be used by tuning; start_tap should be multiple of step_size chosen and its valid range is 0-255; Not valid when TAP_VAL_UPDATED_BY_HW is set to zero. Tuning algorithm uses this as the start tap value for scanning through trimmer taps.
17	0x1	_NONE_	**TAP_VAL_UPDATED_BY_HW**: This bit is functional only in tunable modes (SDR50, SDR104 and HS200) SW can choose to update the tap val by itself by clearing this bit; Preferred value is 1 - tap val is updated by HW. If this bit is cleared, HW does not update tap_val per every tuning iteration. SW can update it as desired. Tuning pattern match is indicated by sampling_clock_select per every tuning iteration. SW has to maintain the status of each tuning iteration and determine the best PASS window to fix the final sampling point. And SW can program NUM_TUNING_ITERATIONS as desired. Once the number of tuning commands issued reaches number of tuning iterations programmmed, execute_tuning bit will be cleared to indicate the completion of tuning procedure. Please note that using this option violates Host Spec but provided for legacy reasons. It helps us in using legacy SW tuning solution incase HW solution does not work.
15:13	TRIES_40	_NONE_	**NUM_TUNING_ITERATIONS**: The number of tuning iterations to be used by tuning circuit. **0** = TRIES_40 **1** = TRIES_64 **2** = TRIES_128 **3** = TRIES_192 **4** = TRIES_256
12:6	0x2	0x1	**MUL_M**: implements a multipler - M+1 Final tap value is derived from best passing window and calculated as follows. Final tap value = first_pass + ((last_pass - first_pass)*Q); where Q = percentage of pass window;default-75% Q = M+1/(2^N); N:1…7 M:should be in range [0:2^N-1];
5:3	0x2	_NONE_	**DIV_N**: implements a divider - 2^N; max div is 2^7 =>128

Bit	Reset	PROD	Description
2:0	0x0	_NONE_	**TUNING_WORD_SEL**: Selects desired word from 256-bit tuning status bitmap status_word[31:0] = status[255:0] >> (tuning_word_sel * 32)

SDMMCA_VENDOR_TUNING_CNTRL1_0

Vendor Tuning Control1 register
Different step size is required in SDR50 mode to cover two UI (100MHz => 2*10ns)
With 70ps/tap trimmer resolution, we can cover almost 2UI using step_size=8 in SDR50 and step_size=4 in SDR104.
Tuning will be done in HS200 - SDR mode only.
HS200 - tuning @200MHz - UHS_SEL should be SDR104 for executing tuning
Before initiating data transfers in HS400 mode, tuning procedure should be executed in HS200 mode with IO clock running @200MHz
DQ_OFFSET: offset between even and odd bits. Td is per-tap delay in trimmer. When the DQ offset function is turned off (DQ_OFFSET[1:0]=00), there is no offset between ZI0~7. When the DQ offset function is turned on
(DQ_OFFSET[1:0]=01, 10, 11),
extra delays are added at the odd bits (DQ1, 3, 5,7). Thus, there is an offset between even bits and odd bits. DQ offset function is turned on during auto-tuning to avoid the window-merged issue and turned off during normal operation.
00 no offset
01 1*Td
10 2*Td
11 3*Td

Offset: 0x1c4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000400 (0b00xx,xxxx,xxxx,xx00,0000,0100,x000,x000)

Bit	Reset	Description
31:30	0x0	**DQ_OFFSET**: offset between even and odd bits
17	0x0	**FIRST_PASS_WINDOW_SEL**: This enables the selection of the first pass window by the HW tuning engine. First pass window select feature is enabled only when STEP_SIZE is set to 0.

Bit	Reset	Description
16	0x0	**FALSE_PASS_MASK**: This enables masking of the false pass windows from the tap value selection. False pass mask feature is enabled only when STEP_SIZE is set to 0.
15:8	0x4	**MIN_PASS_WINDOW_WIDTH**: This is used to mask false passes.Tuning engine considers PASS windows of size > MIN_WIDTH for tap value calculation. Allowed range is 1<= MIN_PASS_WINDOW_WIDTH <=4.
6:4	0x0	**STEP_SIZE_SDR104_HS200**: tap_val is incremented by step_size for every tuning iteration - used in SDR104/HS200/HS400 mode increment = 2^step_size; step_size should be in range 0-4. Others are RSVD
2:0	0x0	**STEP_SIZE_SDR50**: tap_val is incremented by step_size for every tuning iteration - used in SDR50 mode increment = 2^step_size; step_size should be in range 0-4. Others are RSVD

SDMMCA_VENDOR_TUNING_STATUS0_0

Vendor Tuning Status0 register

Offset: 0x1c8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**STATUS_WORD**: Each bit indicates the status of each tuning iteration, when tap value is updated by HW (TAP_VAL_UPDATED_BY_HW=1); 0-Tuning pattern not matched 1-tuning pattern matched We have a total of 256 tap values. SW can issue a max.of 256 tuning commands for debug. SW needs to read this register eight times to get status of all 256 iterations by changing tuning_word_sel status[255:0] is left shifted and loaded into this register every time when tuning_word_sel is changed status_word[31:0] = status[255:0] >> (tuning_word_sel * 32)

SDMMCA_VENDOR_TUNING_STATUS1_0

Vendor Tuning Status1 register

Offset: 0x1cc
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**PASS_WINDOW_END_BEFORE_FINE_TUNING**: End tap value of best PASS window found by scan FSM
23:16	0x0	**PASS_WINDOW_START_BEFORE_FINE_TUNING**: Start tap value of best PASS window found by scan FSM
15:8	0x0	**PASS_WINDOW_END_AFTER_FINE_TUNING**: End tap value of best PASS window after fine tuning
7:0	0x0	**PASS_WINDOW_START_AFTER_FINE_TUNING**: Start tap value of best PASS window after fine tuning

SDMMCA_VENDOR_CLK_GATE_HYSTERESIS_COUNT_0

Vendor Clk gating Hysteresis Counter initial value

Offset: 0x1d0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,1111)

Bit	Reset	Description
5:0	0xf	**CLK_COUNT**: Before gating second level clocks controller will wait for these many cycles. we can recover if any idle windows are missed in clken equation

SDMMCA_VENDOR_PRESET_VAL0_0

Vendor Preset Value Registers
SD host spec defines one preset value register for each bus speed mode which should be set by host by some unique method.
Preset values vary based on the base frequency used which is in SW (SoC system driver) control. System driver supposed to set BASE_CLK_FREQ in VENDOR_CLOCK_CNTRL register before handling over the control to SD host standard driver.
In the similar way, system driver should set below vendor preset values based on the base clock

frequency and the desired card clock frequency in each
bus speed mode
This should be done after every time SDMMC is reset and after every soft reset.
This is important as all SDMMC controllers follow the same register map, but could be programmed with different frequencies depending on the use case.
Default values are set assuming base clock frequency=208 MHz.

Offset: 0x1d4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00201000 (0bxx00,0000,0010,0000,0001,0000,0000,0000)

Bit	Reset	Description
29:20	0x2	**SDCLK_FREQ_SEL_HIGH_SPEED**: System software programs 10-bit divider value to generate SD clk in default speed mode (<50MHz,3.3Vsignaling) This value is readable in the standard via PRESET_SDR12_AND_HIGH_0_SDCLK_FREQ_VAL_LOW register field Default val is 0x2 assuming 208MHz base clock
19:10	0x4	**SDCLK_FREQ_SEL_DEFAULT**: System software programs 10-bit divider value to generate SD clk in default speed mode (<25MHz,3.3Vsignaling) This value is readable in the standard via PRESET_DEFAULT_AND_INIT_0_SDCLK_FREQ_VAL_HIGH register field Default val is 0x4 assuming 208MHz base clock
9:0	0x0	**SDCLK_FREQ_SEL_INIT**: System software programs 10-bit divider value to generate desired SD clk frequency during initialization This value is readable in the standard via PRESET_DEFAULT_AND_INIT_0_SDCLK_FREQ_VAL_LOW register field For Eg., if 400KHz SDCLK is desired @base clk freq=48MHz, this register should be programmed with 0x3C

SDMMCA_VENDOR_PRESET_VAL1_0

Offset: 0x1d8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00100804 (0bxx00,0000,0001,0000,0000,1000,0000,0100)

Bit	Reset	Description
29:20	0x1	**SDCLK_FREQ_SEL_SDR50**: System software programs 10-bit divider value to generate SD clk in SDR50 mode (<100MHz,1.8Vsignaling) This value is readable in the standard via PRESET_SDR50_AND_SDR25_0_SDCLK_FREQ_VAL_HIGH register field Default val is 0x1 (gives 2N divider) assuming 208MHz base clock
19:10	0x2	**SDCLK_FREQ_SEL_SDR25**: System software programs 10-bit divider value to generate SD clk in SDR25 mode (<50MHz,1.8Vsignaling) This value is readable in the standard via PRESET_SDR50_AND_SDR25_0_SDCLK_FREQ_VAL_LOW register field Default val is 0x2 assuming 208MHz base clock
9:0	0x4	**SDCLK_FREQ_SEL_SDR12**: System software programs 10-bit divider value to generate SD clk in SDR12 mode (<25MHz,1.8Vsignaling) This value is readable in the standard via PRESET_SDR12_AND_HIGH_0_SDCLK_FREQ_VAL_HIGH register field

SDMMCA_VENDOR_PRESET_VAL2_0

Offset: 0x1dc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000800 (0bxxxx,xxxx,xxxx,0000,0000,1000,0000,0000)

Bit	Reset	Description
19:10	0x2	**SDCLK_FREQ_SEL_DDR50**: System software programs 10-bit divider value to generate SD clk in DDR50 mode (<50MHz,1.8Vsignaling) This value is readable in the standard via PRESET_DDR50_AND_SDR104_0_SDCLK_FREQ_VAL_HIGH register field Default val is 0x2 assuming 208MHz base clock
9:0	0x0	**SDCLK_FREQ_SEL_SDR104**: System software programs 10-bit divider value to generate SD clk in SDR104 mode (<208MHz,1.8Vsignaling) This value is readable in the standard via PRESET_DDR50_AND_SDR104_0_SDCLK_FREQ_VAL_LOW register field

SDMMCA_SDMEMCOMPPADCTRL_0

SDMEMCOMP Pad control register
This register is used to control COMP pad inputs.

Offset: 0x1e0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x08000000 (0b00x0,1xx0,0000,xxx0,0000,0000,0000,0000)
PROD: 0x00007000 (0bxxxx,xxxx,xxxx,xxx0,0111,xxxx,xxxx,xxxx)

Bit	Reset	PROD	Description
31	0x0	_NONE_	**PAD_E_INPUT_OR_E_PWRD**: used to control E_INPUT(for SDMMC1/3) and E_PWRD (for SDMMC4) input of pu/pd comp pad should be set at least 1usec before starting auto-cal and cleared once auto-calibration is done (for power saving) NOTE: E_PWRD = !PAD_E_INPUT_OR_E_PWRD and E_INPUT = PAD_E_INPUT_OR_E_PWRD
30	0x0	_NONE_	**COMP_PAD_E_PBIAS_BUF**: Active high. Enables internally generated bias levels for driver PMOS. We dont use this feature for SDMMC pads. SW *should not* set this bit. Keeping reg for debug purpose.
28:27	0x1	_NONE_	**COMP_PAD_DRV_TYPE**: used to control drv_type input of BDSDMEMLVCOMP_C pad
24:20	0x0	_NONE_	**COMP_PAD_DRVUP_OVR**: used to drive DRVUP input of COMP pad if AUTO_CAL_ENABLE is disabled
16:12	0x0	0x7	**COMP_PAD_DRVDN_OVR**: used to drive DRVDN input of COMP pad if AUTO_CAL_ENABLE is disabled
11	0x0	_NONE_	**COMP_PAD_E_TEST_OUT**: used to control e_test_out input of COMP pad
10:7	0x0	_NONE_	**COMP_PAD_RFU_IN**: Unused comp pad input pins. Reserved for future usage.
6:4	0x0	_NONE_	**COMP_PAD_TEST_SEL**: used to control test_sel input of COMP pad
3:0	0x0	_NONE_	**SDMMC2TMC_CFG_SDMEMCOMP_VREF_SEL**: Select different bias levels for driver PMOS when E_PBIAS_BUF=1. We dont use this feature for SDMMC pads. SW *should not* set this bit. Keeping reg for debug purpose.

SDMMCA_AUTO_CAL_CONFIG_0

SDMEMCOMP pad auto-calibration settings
AUTO_CAL_SLW_OVERRIDE
0 (Normal operation) pad DRVDN/UP_SLWR/F tied to AUTO_CAL output
DRDVDN/UP_SLWR/F[1:0] = AUTO_CAL_PULLDOWN/UP[4:3]
1 (override) use CFG2TMC_SDIO[1|3]*_DRVDN/UP_SLWR/F pins to control pad slew inputs
AUTO_CAL_OVERRIDE
0 (normal operation): use AUTO_CAL_PU/PD_OFFSET as an offset to the calibration state machine setting
1 (override) : use AUTO_CAL_PU/PD_OFFSET register values directly

Offset: 0x1e4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00010000 (0b0000,xxxx,xxxx,x001,xxx0,0000,xxx0,0000)
PROD: 0x20000000 (0bxx1x,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	PROD	Description
31	0x0	_NONE_	**AUTO_CAL_START**: Writing a one to this bit starts the calibration state machine. This bit must be set even if the override is set in order to latch in the override value.
30	0x0	_NONE_	**AUTO_CAL_OVERRIDE**: AUTOCAL override. **0** = NORMAL :0 (normal operation): use AUTO_CAL_PU/PD_OFFSET as an offset to the calibration state machine setting **1** = OVERRIDE :1 (override) : use AUTO_CAL_PU/PD_OFFSET register values directly
29	DISABLED	ENABLED	**AUTO_CAL_ENABLE**: AUTOCAL enable. **0** = DISABLED :0 (disabled): use sdmmc2tmc_cfg* register settings for pullup/dn **1** = ENABLED :1 (normal operation): use SDMMC generated pullup/dn (override or AUTOCAL)
28	0x0	_NONE_	**AUTO_CAL_SLW_OVERRIDE**: AUTOCAL slew rate override **0** = NORMAL :0 (Normal operation) pad DRVDN/UP_SLWR/F tied to AUTO_CAL output DRDVDN/UP_SLWR/F[1:0] = AUTO_CAL_PULLDOWN/UP[4:3] **1** = OVERRIDE :1 (override) use CFG2TMC_SDIO[1|3]*_DRVDN/UP_SLWR/F pins to control pad slew inputs
18:16	0x1	_NONE_	**AUTO_CAL_STEP**: calibration step interval (in microseconds)

Bit	Reset	PROD	Description
12:8	0x0	_NONE_	**AUTO_CAL_PD_OFFSET**: 2's complement offset for pull-down value
4:0	0x0	_NONE_	**AUTO_CAL_PU_OFFSET**: 2's complement offset for pull-up value

SDMMCA_AUTO_CAL_INTERVAL_0

SDMEMCOMP pad calibration interval

Offset: 0x1e8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**AUTO_CAL_INTERVAL**: 0: do calibration once Otherwise, auto-calibration occurs at intervals equivalent to the programmed number of microseconds.

SDMMCA_AUTO_CAL_STATUS_0

SDMEMCOMP pad calibration status

Offset: 0x1ec
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0xx0,0000,xxx0,0000,xxx0,0000,xxx0,0000)

Bit	Reset	Description
31	0x0	**AUTO_CAL_ACTIVE**: One when auto calibrate is active - valid only after auto calibrate sequence has completed (AUTO_CAL_ACTIVE == 0)
28:24	0x0	**AUTO_CAL_PULLDOWN_ADJ**: Pulldown code sent to pads
20:16	0x0	**AUTO_CAL_PULLUP_ADJ**: Pullup code sent to pads

Bit	Reset	Description
12:8	0x0	**AUTO_CAL_PULLDOWN**: Pulldown code generated by auto-calibration
4:0	0x0	**AUTO_CAL_PULLUP**: Pullup code generated by auto-calibration

SDMMCA_IO_SPARE_0

SPARE bits provied to use in eco process. These SPARE_OUT bits go to pipe -> pad and then come back as SPARE_IN
SPARE_OUT[3] : IO_SPARE[19] - used as MUX select which selects between one cycle delay and two cycle delay versions of cmd_oen to mask wdata of IB
capture flop.
0x0 : selects zero cycle delayed version
0x1 : selects one cycle delayed version - recommended
SPARE_OUT[2] : IO_SPARE[18] - used as active low enable for gating both CMD_IN and DAT_IN aysnc FIFOs wdata when we are driving CMD/DAT lines.
0x0 : write 1 when OEN is active and Zi value when OEN is not active into FIFO (default)
0x1 : write Zi value into FIFO irrespective of OEN state.

Offset: 0x1f0
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0008XXXX (0b0000,0000,0000,1000,xxxx,xxxx,xxxx,xxxx)

Bit	R/W	Reset	Description
31:16	RW	0x8	**SPARE_OUT**
15:0	RO	X	**SPARE_IN**

SDMMCA_CIF2AXI_CTRL_0

SDMMC CIF2AXI control register
DMA transaction (MC transaction) attributes

Offset: 0x1fc
Read/Write: R/W

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:8	0x0	**MC_WRITE_REQ_STREAM_ID**: MC write transaction stream ID
7:0	0x0	**MC_READ_REQ_STREAM_ID**: MC read transaction stream ID

SDMMCA_TZ_DMA_CTRL_0

SDMMC DMA requests security attribute control register for DMA transaction (MC transaction) security attributes.
This register is used to control write/read access to secure memory region.
This register can be accessed only by TZ. Non-secure writes to this register are dropped by controller and reads return all ones.
Controller will assert PSLVERR when this register is accessed by a non-secure master.
Usage:
wsb_ns = AWPROT[1] = ~{SDMMCxx_TZ_DMA_CTRL_MC_WRITE_REQ_TZ_ACCESS_EN};
rsb_ns = ARPROT[1] = ~{SDMMCxx_TZ_DMA_CTRL_MC_READ_REQ_TZ_ACCESS_EN};

Offset: 0x200
Read/Write: R/W
Parity Protection: N
Shadow: N
Secure: Trust Zone Protected
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**MC_WRITE_REQ_TZ_ACCESS_EN**: TZ must set this bit to 1 to enable write access to secure memory region. TZ must clear this bit to disable write access to secure memory region.
0	0x0	**MC_READ_REQ_TZ_ACCESS_EN**: TZ must set this bit to 1 to enable read access to secure memory region. TZ must clear this bit to disable read access to secure memory region.

SDMMCA_VENDOR_MISC_CNTRL3_0

Offset: 0x204

Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x4f01000f (0b0100,1111,xxxx,xxx1,0000,0000,xxx0,1111)

Bit	Reset	Description
31	0x0	**ALLOW_INTRABLOCK_CLK_STALLING_IN_SDR50**: This disables the intrablock clock stopping in SDR50 mode
30	0x1	**STOP_TRIM_IN_CLK_DURING_TUNING**: When set to 1, rx clk trimmer and rx fifos input clock is stopped when tap value is changed during tuning process. When set to 0, rx clk trimmer and rx fifos input clock is not stopped during tap value change. Trimmer output is also clamped to zero during tuning tap val change.
29:24	0xf	**TUNING_TRIMMER_RECOVERY_TIME**: Trimmer output may glitch for 2 or 3 cycles, when tap value is changed irrespective of its input clock state. We should not use trimmer output during this uncertainity window. If STOP_TRIM_IN_CLK_DURING_TUNING is set to 1, - trimmer input clock is stopped for TUNING_TRIMMER_RECOVERY_TIME cycles and - rx fifos input clock is stopped (trimmer output clock is clamped) for TUNING_TRIMMER_RECOVERY_TIME cycles. This is required not to propagate glitch into FIFOs and other downstream logic.
16	0x1	**EXTEND_SYNC_INTR_MASK_DURING_ABORT_OR_STOP_CMD**: When an async abort is issued during read operation, dat_fsm will move to idle state as soon as CMD12/CMD52 END bit is sent by core. This would start sync_intr_period. But due to the intermediate delay stages present in pipemacro and padmacro, END bit reaches device after some cycles. So, device would not stop data transmission till it sees END bit of ABORT CMD. During this time, core controller receives the data already driven by the device. If DAT[1] line has any zeroes during this period, SDIO sync interrupt detector will generate a spurious card interrupt. This register bit is used to mask sync intr detection period to avoid spurious interrupts. SW should not write into this field unless it is published in TRM/IAS.
15:8	0x0	**E_DIFF_DQ**: diff/Vref rx selection for DAT[7:0] If set to 1, enables differential amplitude receiver for DAT lines in EMMC IOBRICK. If set to 0, enables vref receiver for DAT lines in EMMC IOBRICK. (default)
4	0x0	**E_DIFF_CMD**: diff/Vref rx selection for CMD If set to 1, enables differential amplitude receiver for CMD in EMMC IOBRICK. If set to 0, enables vref receiver for CMD in EMMC IOBRICK. (default)
3	0x1	**DAT_OE_POSTAMBLE_EN**: If set,DAT pads output driver will be disabled one cycle after END bit of DATA pkt sent. If cleared, DAT pads output driver will be disabled in the same cycle END bit is sent.

Bit	Reset	Description
2	0x1	**DAT_OE_PREAMBLE_EN**: If set,DAT pads output driver will be enabled one cycle before START bit of DATA pkt is transmitted on DAT lines. If cleared, DAT pads output driver will be enabled in the same cycle START bit is sent.
1	0x1	**CMD_OE_POSTAMBLE_EN**: If set,CMD pad output driver will be disabled one cycle after END bit of CMD pkt sent. If cleared, CMD pad output driver will be disabled in the same cycle END bit is sent.
0	0x1	**CMD_OE_PREAMBLE_EN**: If set,CMD pad output driver will be enabled one cycle before START bit of CMD pkt is transmitted on DAT lines. If cleared, CMD pad output driver will be enabled in the same cycle START bit is sent.

SDMMCA_VENDOR_MISC_CNTRL4_0

SPARE register1

Offset: 0x20c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0xffff0000 (0b1111,1111,1111,1111,0000,0000,0000,0000)

Bit	Reset	Description
31:16	0xffff	**SDMMC_SPARE3**: Spare register bits with reset value of 1
15:0	0x0	**SDMMC_SPARE2**: Spare register bits with reset value of 0

SDMMCA_SDMEMCOMPPADCTRL_MISC_CTL_0

SDMEMCOMP Pad misc control register
This register is used to control COMP pad inputs.

Offset: 0x214
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:4	0x0	**COMP_PAD_SPARE_VDD**: Comp pad spare inputs
3:0	0x0	**COMP_PAD_SPARE_VAUXC**: Comp pad spare inputs

9.6 I2C Controller (I2C)

9.6.1 Overview

The general purpose high-speed I^2C controller enables system expansion using I^2C-based devices. It is compliant with the NXP® *I^2C-bus Specification*, revision 03 and the Arm® *AMBA APB Protocol Specification* v2.0. The number of devices that can be connected to the I^2C-bus is limited to a maximum bus capacitive loading of 400 pF, which is high enough to support many devices. This controller could also be connected to an I^2C I/O expander for more convenient use with multiple devices.

9.6.1.1 Features

- Supported Modes:
  - Standard Mode (SM)
  - Fast Mode (FM)
  - Fast Mode Plus (FM+)
  - High-Speed (HS)
- Simultaneous master controller and slave controller operation
- Clock stretching by slave device
- One to sixteen-word burst data transfers in DMA mode for the Master controller (packet mode of operation)
- 64 Kbyte transfers in packet mode. Can be extended beyond 64 Kbytes by breaking up transfers into multiple packets (in both DMA and non-DMA modes)
- 64 Kbyte transfers in FIFO mode from slave
- 7-bit or 10-bit addressing transactions (both master and slave)
- Master is capable of data transfers to/from two or more slaves consecutively with a repeated-start condition
- Fully programmable 7-bit or 10-bit address for the slave
- Bus clear operations to address SDA driven LOW issues

- General call addressing
- Recognition and transfer of data to peripherals that do not send an acknowledge (ACK)

Note: Packet based DMA support can reduce software complexity since the interaction between driver and hardware is minimized.

The I^2C controller supports DMA for master modes of operation over the APB bus; there is no DMA support for slave modes of operation. The I^2C controller also supports packet mode transfers where the data to be transferred is encapsulated in a predefined packet format as payload and sent to the I^2C controller over the APB bus. The header of the packet specifies the type of operation to be performed, the size, and other parameters (described in detail in subsequent sections). There are a total of nine I^2C instances; available I^2C ports depend on pin-muxing and system configuration options selected. The following table provides a summary of all I^2C instances.

Table 9.24 I^2C Controller Map

I^2C Instance	Verilog Ball Names
I2C1	GEN1_I2C_[SCL,SDA]
I2C2	GEN2_I2C_[SCL,SDA]
I2C3	CAM_I2C_[SCL, SDA]
I2C4	DP_AUX_CH1_[P,N]
I2C5	PWR_I2C_[SCL, SDA]
I2C6	DP_AUX_CH0_[P,N]
I2C7	DP_AUX_CH2_[P,N]
I2C8	GEN8_I2C_[SCL,SDA]
I2C9	DP_AUX_CH3_[P,N]

9.6.1.2 Clocking

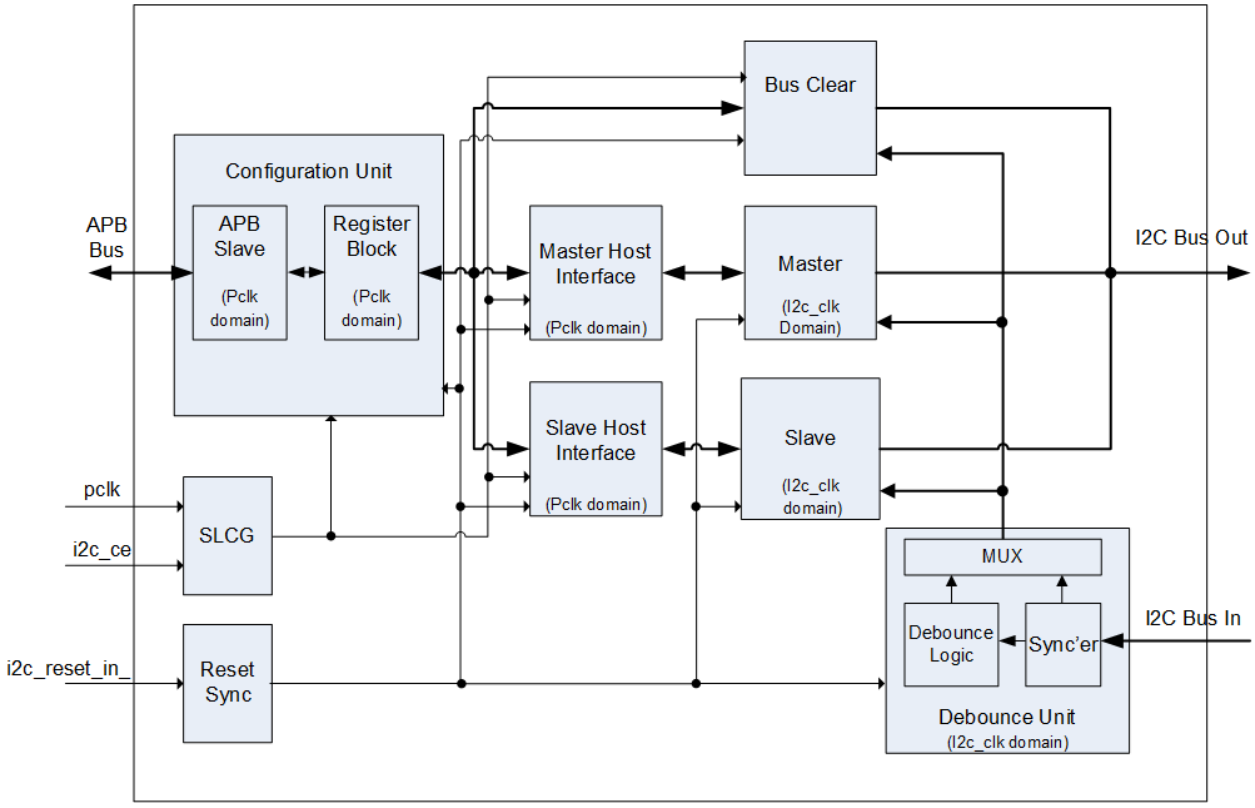
The I^2C controller has three clock domains:

1. The host interface runs on the APB clock (pclk).
2. The I^2C controller clock runs on a variable software-controlled frequency.
3. The time-out logic runs on a 32 kHz clock.

9.6.2 Functional Description

The I^{2}C controller supports both Master and Slave operations. The Master can address the internal slave (for basic testing) or an external 7-bit or 10-bit addressed Slave device. The Master can be programmed for either a single slave transaction or a two-slave transaction. The two-slave transaction is generally useful in a random read from an external device. When reading from an external device, the random read-address needs to be set with an initial dummy-write to the device, followed by a repeated-start and a read transaction to the same device. The internal Slave address can be programmed to be either a 7-bit or a 10-bit address. The figure below shows the I^{2}C controller's block diagram with master, slave, and bus clear controllers.

Figure 9.25 I^{2}C Controller Block Diagram



9.6.2.1 Second Level Clock Gating (SLCG)

The I^{2}C implements hardware-controlled Second Level Clock Gating (SLCG) in both master and slave controller logic to save power. It is enabled by default. There is also a provision to override SLCG using programmable register bit fields. The override bits are available per SLCG block in

controller register bit fields. Also, the CAR module provides a global clock override signal using which all SCLGs can be disabled simultaneously.

9.6.2.2 Bus Clear Master Operation

Bus clear logic helps to resolve I^2C bus hang issues. If the data line (SDA) is permanently pulled low because a slave device is pulling it low continuously for some unknown reason, the I^2C master loses the bus arbitration and stops driving the I^2C bus. In this case, software can use the Bus Clear master to get the SDA line released by sending clock pulses on the SCL line. Per the I^2C specification, the device that held the bus Low should release it sometime within nine clock pulses. But if the device needs more clock pulses than the default nine pulses, software has an option to do so by programming a configurable register bit field (I2C_I2C_BUS_CLEAR_CONFIG_0[BC_SCLK_THRESHOLD]) to the required number of clock pulses.

9.6.2.2.1 **Bus Clear Programming Procedure:**

Upon ARB_LOST notification from the I^2C controller, software initiates the Bus Clear operation. Note that ARB_LOST could be due to many reasons; for example, in a multi-master platform, any other master could occupy the I^2C-bus already and if software initiates a data transfer from the I^2C controller at this time, the I^2C master would back-off or it may lose the bus during the bus-arbitration procedure if both masters start transfers simultaneously. In this case the I^2C master notifies it with ARB_LOST status in packet mode of operation. Other ARB_LOST cases included programming issues such as controller register configuration, pin-muxing, pin tri-state clear, etc.

The Bus Clear operation needs to be used only if the ARB_LOST notification is due to a slave device pulling the SDA line continuously low. I^2C hardware does not identify on its own whether ARB_LOST is due to bus busy or another master transfer or slave device continuously pulling the SDA line low.

1. Continue with the same clock settings as before or change if high or low frequency is needed.
2. Reset the I^2C controller using module reset bit CLK_RST_CONTROLLER_RST_DEV_I2Cx_0 [SWR_I2Cx_RST] (where x=1,2,3,..,10) in CAR module after ARB_LOST notification from the controller.
3. Program I2C_I2C_BUS_CLEAR_CONFIG_0[BC_SCLK_THRESHOLD] bit field to the required number of clock pulses. If BC_TERMINATE=THRESHOLD is chosen, program BC_SCLK_THRESHOLD to N+1 where “N” is the required number of clock pulses on SCL line.
4. Select STOP or NO_STOP condition using I2C_I2C_BUS_CLEAR_CONFIG_0[BC_STOP_COND] bit field.
5. Program I2C_I2C_BUS_CLEAR_CONFIG_0[BC_TERMINATE] as needed.
6. Set I2C_I2C_CONFIG_LOAD_0[MSTR_CONFIG_LOAD] register bit field to 1 and wait until this bit field is auto-cleared by Hardware to zero.

7. Finally set I2C_I2C_BUS_CLEAR_CONFIG_0[BC_ENABLE] bit field to start the bus clear operation. Hardware auto clears this bit after bus clear operation is done.
8. Once enabled, bus clear logic starts sending clock pulses on the SCL line.
9. Based on BC_TERMINATE settings,
   a. IMMEDIATE: Bus Clear logic sends clock pulses until SDA is released or the threshold count is reached whichever is earlier or
   b. THRESHOLD: Send threshold number of clock pulses irrespective of SDA line release status (released before threshold is reached or not-released with threshold count reached)
10. If SDA is released within the programmed number of clock pulses, Bus Clear logic terminates the transaction with STOP/No-STOP condition based on BC_STOP_COND settings.
11. Then set the bus_clear operation status in I2C_I2C_BUS_CLEAR_STATUS_0[BC_STATUS] register bit field.
12. Finally sets the BUS_CLEAR_DONE bit field in I2C_INTERRUPT_STATUS_REGISTER_0 and also interrupts the system if corresponding interrupt enable bit is set in I2C_INTERRUPT_MASK_REGISTER_0. So Software has to wait until it receives the BUS_CLEAR_DONE interrupt if interrupt method is used or BUS_CLEAR_DONE status bit set if status-polling method is used before going for BUS CLEAR operation successful/fail status check in I2C_I2C_BUS_CLEAR_STATUS_0[BC_STATUS]
13. After receiving BUS_CLEAR_DONE interrupt, also check if I2C_I2C_BUS_CLEAR_CONFIG_0[BC_ENABLE] bit is auto-cleared by Hardware. If this bit is not cleared, it means, I^2C clock settings are not correct. Fix the clocks programming.
14. After the Bus Clear operation is done, an I^2C controller reset is required in certain cases. See the following table for the details.

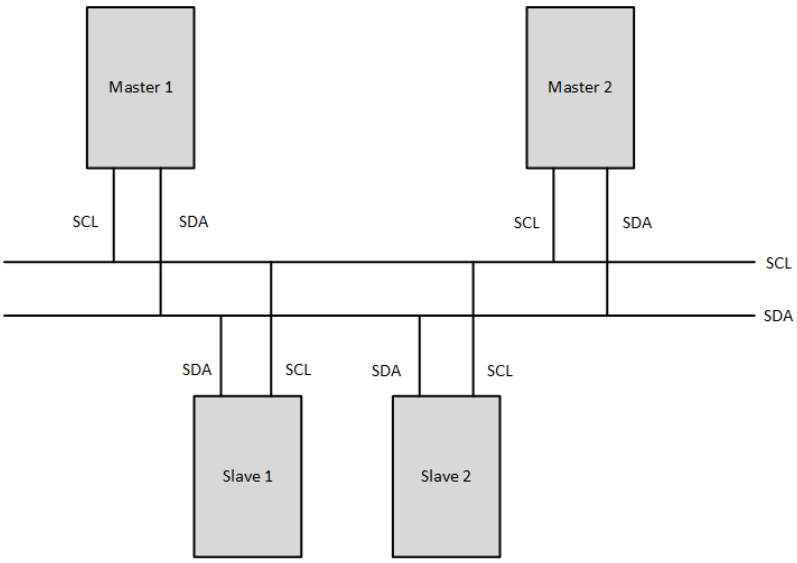
Table 9.25 I^2C Controller Reset Required After Bus Clear

Initial I^2C Bus State	BC_TERMINATE Setting	BC_STOP_COND Setting	Reset Required
BUSY	IMMEDIATE	STOP	No
BUSY	IMMEDIATE	NO_STOP	**Yes**
BUSY	THRESHOLD	STOP	No
BUSY	THRESHOLD	NO_STOP	**Yes**
Free	IMMEDIATE	STOP	No
Free	IMMEDIATE	NO_STOP	No
Free	THRESHOLD	STOP	No
Free	THRESHOLD	NO_STOP	No

9.6.2.3 Multi-Master Operation

The I^2C bus supports multi-master operation. That means there can be more than one master device connected to the bus as shown in the figure below and each master can drive the bus to perform the data transfers. This could cause collisions when two or more masters try to drive the bus simultaneously and could result into bus corruption. To avoid this, there needs to be a method for deciding which device takes the bus control and complete its transmission. I^2C uses clock synchronization and arbitration to achieve multi-master functionality.

Figure 9.26 I^2C Multi-Master Bus



Clock synchronization is a process to synchronize the clocks of two or more master devices. For successful bus arbitration, a synchronized clock is needed. For this, each master device should monitor the bus SCL line and perform low and high transitions on its internal clock (SCL) line accordingly. Once a master pulls the SCL low, it holds the SCL line in that state until all masters have released it to put the clock into a high state. Similarly, the SCL line is high until the first master pulls it low.

Arbitration is a process to determine which master can use the bus when two or more masters try to use the bus simultaneously. In normal transfers, only one master drives the bus if the bus is free. But if two masters initiate transfers simultaneously, arbitration is needed to determine which master wins the bus and complete its transaction. Arbitration is done using the SDA line. During every bit while the SCL line is HIGH, each master checks to see if the SDA line level matches what it has driven. If there is a mismatch, this master loses the arbitration and stops driving the SDA line. Other master goes ahead and completes its transaction. So no information is lost during the arbitration.

Though data loss is not seen in multi-master arbitration but is given in the I^2C specification, there are undefined conditions if the arbitration procedure is still in progress and the following happens:

- Master 1 sends a repeated START condition and Master 2 sends a data bit
- Master 1 sends a STOP condition and Master 2 sends a data bit
- Master 1 sends a repeated START condition and Master 2 sends a STOP condition

Note: The I^2C specification does not indicate how to handle these cases. So the system designer has to consider these cases while designing an I^2C multi-master system.

9.6.2.3.1 Multi-Master Requirements

1. Each master in a multi-master system should monitor the bus status so that it starts its transmissions only when the bus is free. The bus is considered to be busy after the START condition. The bus is considered to be free again a certain time (minimum tBUF as given in the I^2C specification) after the STOP condition. So if a device which needs the bus for its transmissions detects that the bus is busy, then it has to wait until the STOP condition is seen on the bus and then transmit its data.
2. All devices connected to a multi-master I^2C bus must be multi-master capable. Otherwise, a device with single master capability only may not detect the bus busy status and could interfere with on-going transactions on the bus causing the bus corruption.

9.6.2.3.2 I2C Usage Guidelines for Multi-Master Operation

1. The I^2C follows Multi-master bus requirements. When initiating a transfer by software, it checks the I^2C bus status and starts the transmission if the bus is free. It uses the bus START condition for Bus busy detection.
2. Bus Busy Detection: The I^2C Master sets its internal Bus Busy status when it sees START condition on the bus. The status remains busy until a STOP condition is seen.
3. In a multi-master system where I^2C is used as one of the master devices, the I^2C controller needs to be ON:
   a. either from the beginning of the I^2C bus enabled or
   b. when the bus is free
      So that the master detects the START condition of a transaction (if any) issued by another master so bus busy does not interrupt on-going communications.
4. For the I^2C controller to be ON, its clocks need to be enabled and the reset needs to be released. The clocks enable and Reset controls are in the CAR module. See the Clock and Reset Controller Registers section in the Clock Controller chapter. The clock controls from the CAR module are called first level clocks to the I^2C controller.
5. If software implements any transaction level clock gating, it has to be disabled in multi-master systems.

6. The requirement is that the I^{2}C controller should be ON as long as the bus is used in multi-master operation.

Multi-Master in Normal Mode

In Multi-master cases, there is a difference between Normal and Packet Modes in the way I^{2}C works. In Normal Mode, when software initiates a transfer from the I^{2}C Master controller but the bus is busy, the controller waits until the bus becomes free and then goes on to complete the transaction.

Multi-Master in Packet Mode

In Packet mode, when the I^{2}C master is initiated for a data transfer but the bus is busy, the controller does not drive the bus. It cancels the current transaction and sets the ARB_LOST status. Software has to handle the ARB_LOST event with the procedure given in the section on Error Handling.

9.6.2.3.3 Low-Power Operation

I^{2}C is a low-speed serial interface with only two bus lines (SCL, SDA) required. So from the I^{2}C specification side, there is no low-power operation specifically defined for the interface. But to save the power, use the lowest possible speed, if performance is not needed. The following bus speeds are supported by I^{2}C:

- Standard mode (Sm)
- Fast mode (Fm)
- Fast mode Plus (FM+
- High-speed mode (HS mode)

SLCG

I^{2}C has implemented SLCG, a power saving feature, and it is ON by default. Unless required, keep SLCG enabled during the functional transfers in real use cases.

Slave Controller Disable

Slave logic has an enable bit, but keep it disabled if Slave logic is not needed.

Idle Power Programming

In Single Master bus configurations, when the module is not used or interface is idle, disable the module clock by clearing the CLK_ENB_I2C bits in the CAR module. For example, the I2C1 clock

enable bit is CLK_RST_CONTROLLER_CLK_OUT_ENB_I2C1_0[CLK_ENB_I2C1]. Software can clear this bit when it is seen that no transfers are needed from the I2C1 instance.

9.6.2.3.4 Software Interfaces

A typical peripheral controller might require the following steps to communicate or control an external peripheral:

1. Set up the registers
2. Program the 'go' bit to start the interaction with the external peripheral
3. The peripheral controller collects the response from the peripheral and interrupts software
4. Software reads the response

9.6.2.3.5 Packet Flow

Information exchange between the hardware peripheral controller and software as described above can be classified into:

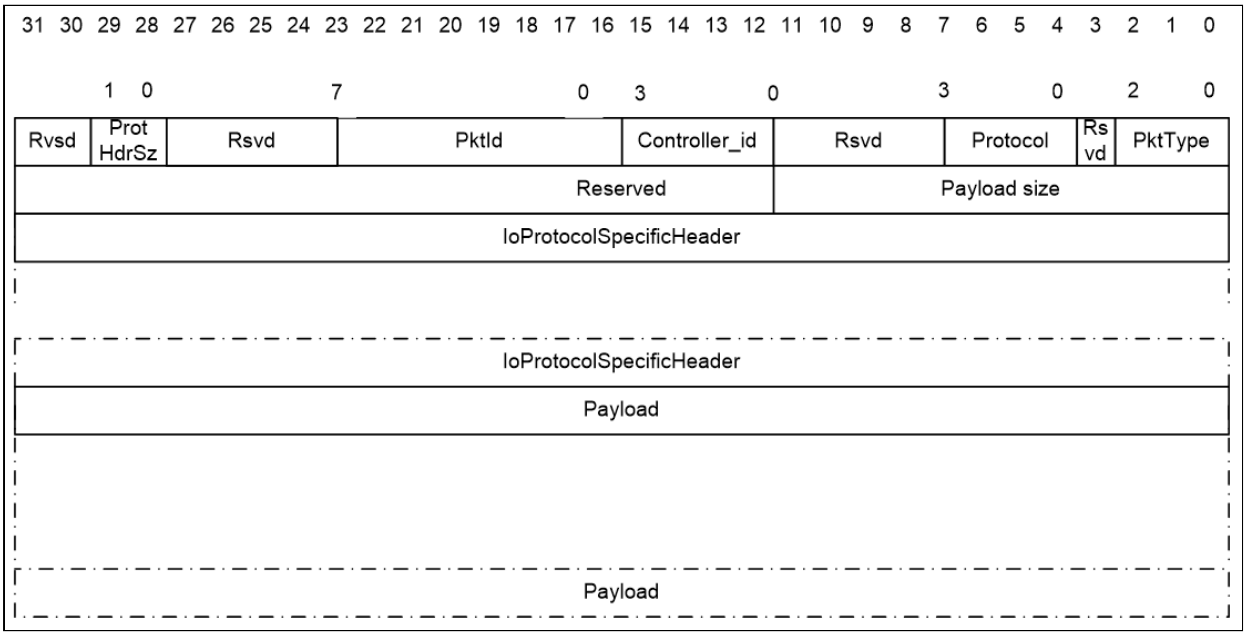
- 'Request' flow which can represent the register writes
- 'Response' flow for peripheral controller responses sent back to software
- 'Interrupt' flow for out-of-band handshakes between hardware and software

Packets within a flow can be exchanged in a manner that requires full intervention from the CPU (PIO mode) or least intervention from the CPU (DMA mode). The I^2C I/O packet structure is described below.

9.6.2.3.6 I/O Packet Format

An I/O packet consists of a header and payload. The header consists of two parts: two words of generic header and one-four words of protocol-specific header.

Figure 9.27 I/O Packet



I/O Packet Word 0

Word 0 of the I/O packet contains the Generic Header Word 0. The contents of generic header word 0 are described in the table below.

Table 9.26 Generic I/O Header Word 0

Bits	Field	Description
31:30	Reserved	Reserved
29:28	ProtHdrSz	Size of the protocol-specific header in words 0 = 1 words 1 = 2 words 2 = 3 words 3 = 4 words For I^2C, ProtHdrSz = 0 for request packets and 1 for response packets.
27:24	Reserved	Reserved
23:16	PktId	PktId is an 8-bit field that identifies the packet. A peripheral controller might choose to log the packet ID for a packet that causes errors
15:12	Controller ID	There might be multiple controllers supporting any format. For example, there might be four I^2C controllers. The Controller ID field specifies the controller for which this packet is destined.
11:8	Reserved	Reserved

Bits	Field	Description
7:4	Protocol	The protocol is indicated in this field 0 = reserved 1 = I^2C 2-15 = reserved
3:3	Reserved	Reserved
2:0	PktType	Packet type information 0 = request 1 = response 2 = interrupt 3 = stop 4~7 = reserved

I/O Packet Word 1

Word 1 of the I/O packet contains Generic Header Word 1. The contents of Generic Header Word 1 are described in the table below.

Table 9.27 Generic I/O Header Word 1

Bits	Field	Description
31:16	Reserved	Reserved
15:0	PayloadSize	Payload size in bytes

Note about Payload Size: For request packets that transmit data, the total packet size is: 8 bytes of generic header + 4 bytes of protocol-specific header + payload_size. However, for request packets with a read command, the payload size indicates the amount of data to be received. Hence the packet size is, 8 bytes of generic header + 4 bytes of protocol-specific header, because there is no payload in this packet, just the receive command.

9.6.2.3.7 Transferring Packets

Packets can be transferred using PIO or DMA modes. The header is a minimum of three words with the first two words reserved for packet information and the third word reserved for protocol-specific requirements. A minimum of four words is needed to transfer one byte to the external peripheral.

9.6.2.3.8 Protocol-Specific Header

I^2C has Protocol-Specific Header with a size of one word. The contents of the I^2C protocol-specific header are described in the next section.

9.6.2.3.9 I2C Master Packet Protocol-Specific Headers

The I^2C master supports two kinds of packets.

1. Transmit packets that flow from host to controller.
2. Receive data packets that flow from controller to host.

Table 9.28 I2C Master Transmit Packet Header

Bit	Name	Description
31:26	Reserved	Reserved
25	Reserved	Reserved. Program 0 to this bit field
24	Reserved	Reserved. Program 0 to this bit field
23	Reserved	Reserved
22	HS_MODE	Enable High speed mode (3.4 MHz operation)
21	CONTINUE_ON_NACK	Enable mode to handle devices that do not generate ACK upon the reception of a byte.
20	SEND_START_BYTE	1 = Send a start byte at the beginning of the transaction
19	READ/WRITE	1 = READ
18	Address mode	1 = 10-bit mode 0 = 7-bit mode
17	IE	Generate interrupt upon packet completion. This is needed only for Packet_Xfer_Complete interrupt only. All_Packets_Xfer_Complete interrupt has no dependency on this bit settings.
16	REPEAT_START/STOP	1 = Put a repeat start condition on the bus (to continue transaction) 0 = Put a stop condition on the bus
15	ContinueXfer	This bit overrides the REPEAT_START/STOP command above. 0: Introduce repeat start or stop condition based on bit 16. 1: Continue with current transfer without stop or repeat start condition.
14:12	HS_MASTER_ADDR	High Speed mode Master code
11:10	Reserved	Reserved
9:0	SLAVE_ADDR	Slave address. Bit 0 is ignored for 7-bit addressing, but should always match bit 19 (READ/WRITE). I2C transfer with Slave-address byte only is not supported in the controller.

9.6.2.3.10 Master Controller: Speed Modes in Normal and Packet Modes

Table 9.29 Speed Modes in Normal/Packet Mode for I2C Master Controller

Normal Mode	Packet Mode
Standard Mode (SM) Fast Mode (FM) Fast Mode Plus (FM+)	Standard Mode (SM) Fast Mode (FM) Fast Mode Plus (FM+) HS Mode (HS)

9.6.2.3.11 DMA Capabilities and Usage Model

Only the I^2C Master controller has DMA capability. I^2C Slave controllers have no DMA capability.

I^2C Master:

1. I/O Transfer size is always known ahead
2. DMA burst size should match I2C_FIFO_CONTROL_0[TX_FIFO_TRIG] level settings
3. For DMA burst size not a multiple of the I/O transfer size in words:
   - Use DMA + PIO or
   - Use lowest DMA burst size possible (for example, 1 word)

I/O Transfer Size	DMA Burst Size	I/O FIFO Threshold
16*(DMA Burst size)	16	Supported
8*(DMA Burst size)	8	Supported
4*(DMA burst size)	4	Supported
1*(DMA Burst size)	1	Supported

Example:

DMA Burst Size	I/O Transfer Size	Supported/Not Supported
16	144 bytes = 36 words	Not Supported
8	80 bytes = 20 words	Not Supported
4	72 bytes = 18 words	Not Supported
1	Any size	Supported

9.6.3 I2C Programming Guidelines

9.6.3.1 I2C Frequency Divisor Register

The FREQUENCY DIVISOR register (CLK_SOURCE_I2C register) must be programmed as a function of the CLK_SOURCE selected for I^2C as follows:

Note

In the below expressions, TLOW is used to represent I2C_I2C_INTERFACE_TIMING_0[TLOW]. Similarly

- THIGH: I2C_I2C_INTERFACE_TIMING_0[THIGH]
- HS_TLOW: I2C_I2C_HS_INTERFACE_TIMING_0_0[HS_TLOW]
- HS_THIGH: I2C_I2C_HS_INTERFACE_TIMING_0_0[HS_THIGH]

Standard/Fast-mode/Fm+:

$$SCL\ Rate = \frac{I2Cx_CLK_SRC}{(I2Cx_CLK_DIVISOR + 1) * (TLOW + THIGH + 2 + Round_Trip_Delay_In_DivClk_cycles) * (I2C_CLK_DIVISOR_STD_FAST_MODE + 1)}$$

HS mode:

$$SCL\ Rate = \frac{I2Cx_CLK_SRC}{(I2Cx_CLK_DIVISOR + 1) * (HS_TLOW + HS_THIGH + 2 + Round_Trip_Delay_In_DivClk_cycles) * (I2C_CLK_DIVISOR_HSMODE + 1)}$$

Where x = 1, 2, 3… 10

The clock enable (e.g., CLK_ENB_I2C1 bit for I2C1 instance) must also be set before any write and read access to the I^2C registers.

9.6.3.1.1 Example Settings for Various I2C Speeds

SM/FM/FM+ Speed Mode settings:

I2C SpeedMode	I2C_CLK_SRC	I2C_CLK_SRC Frequency (in MHz)	I2C_CLK_DIVISOR+1	I2C_CLK_DIVISOR_STD_FAST_MODE+1	I2C_I2C_INTERFACE_TIMING_0_0		I2C_I2C_INTERFACE_TIMING_1_0				I2C DataRate	Debounce
					(TLOW+1)	(THIGH+1)	(TSU_STA+1)	(THD_STA+1)	(TSU_STO+1)	(TBUF+1)		
FM+	PLLP_OUT0	408	(2+1) = 3	(36+1) = 37	(1+1) = 2	(1+1) = 2	(2+1) = 3	(2+1) = 3	(2+1) = 3	(2+1) = 3	918kHz	0
FM	PLLP_OUT0	408	(2+1) = 3	(88+1) = 89	(1+1) = 2	(1+1) = 2	(2+1) = 3	(2+1) = 3	(2+1) = 3	(2+1) = 3	382kHz	2
SM	PLLP_OUT0	408	(2+1) = 3	(79+1) = 80	(8+1) = 9	(7+1) = 8	(8+1) = 9	(8+1) = 9	(8+1) = 9	(8+1) = 9	100kHz	2

High-Speed (HS) Mode settings:

Note: For HS transfers, both Non-HS and HS timing registers need to be programmed as Master code is sent in a Non-HS speed mode. For this, use one of speed modes (SM or FM or FM+) settings from the above table. For data transfers, use the HS timing settings as given in the table below.

I2C SpeedMode	I2C_CLK_SRC	I2C_CLK_SRC Frequency (in MHz)	I2C_CLK_DIVISOR+1	I2C_CLK_DIVISOR_HSMODE+1	I2C_I2C_HS_INTERFACE_TIMING_0_0		I2C_I2C_HS_INTERFACE_TIMING_1_0			I2C DataRate	Debounce
					(HS_TLOW+1)	(HS_THIGH+1)	(HS_TSU_STA+1)	(HS_THD_STA+1)	(HS_TSU_STO+1)		
HS	PLLP_OUT0	408	(2+1) = 3	(2+1) = 3	(8+1) = 9	(3+1) = 4	(9+1) = 10	(9+1) = 10	(9+1) = 10	3.48MHz	0

I2C Timing register settings with I2C_CLK_SRC: CLK_M

Note: HS mode is not possible with CLK_M frequency.

I2C SpeedMode	I2C_CLK_SRC	I2C_CLK_SRC Frequency (in MHz)	I2C_CLK_DIVISOR+1	I2C_CLK_DIVISOR_STD_FAST_MODE+1	I2C_I2C_INTERFACE_TIMING_0_0		I2C_I2C_INTERFACE_TIMING_1_0				I2C DataRate	Debounce
					(TLOW+1)	(THIGH+1)	(TSU_STA+1)	(THD_STA+1)	(TSU_STO+1)	(TBUF+1)		
FM+	CLK_M	19.2	(0+1) = 1	(5+1) = 6	(1+1) = 2	(1+1) = 2	(2+1) = 3	(2+1) = 3	(2+1) = 3	(2+1) = 3	800kHz	0
FM	CLK_M	19.2	(0+1) = 1	(12+1) = 13	(1+1) = 2	(1+1) = 2	(2+1) = 3	(2+1) = 3	(2+1) = 3	(2+1) = 3	~370kHz	0
SM	CLK_M	19.2	(0+1) = 1	(19+1)= 20	(4+1) = 5	(4+1) = 5	(4+1) = 5	(4+1) = 5	(4+1) = 5	(4+1) = 5	96kHz	2

De-bounce impacts the I^2C interface data rate in the following way.

$$I2C\ SCL\ rate = (CLK_SOURCE)/(SOURCE_DIV * (Tlow + Thigh + 2 + Round_(Trip_Delay\ ))\ * STD_(or_(HS_DIV\ ))\ )$$

$$Round_(Trip_Delay\ in)\ I2C_Div\ cycles\ = (Chip_(internal_delay\ ) + Load_Delay + ((((DEBOUNCE_CNT > 0)(DEBOUNCE_CNT * 2 + 2))in\ i2c_clk\ cycles + Synchronizer\ Delay\ in\ i2c_clk\ cycles) * I2C_(Clk_Period\ ))\ ))/(\ STD_(FM_Div\ or)\ HS_Div * I2C_(Clk_Period\ ))$$

Example:

CLK_SOURCE = 408 MHz and Source Div = 5:

I2C_Clock frequency = 408 / 5 = 81.6 MHz, thus I^2C clock period = 12.25 ns

Chip internal delay = 10 ns, Load_Delay = 55 ns, De-bounce = 2, Synchronizer Delay = ~3

Round_Trip_Delay = (10 ns + 55 ns + 9 * 12.25 ns) = ~175.25 ns

FM+ speed case: STD_FM_Div = 10

STD_FM_Div * I2C_Clk_Period = 10 * 12.255 ns = 122.55 ns

Round_Trip_Delay in I2C_Div cycles = Round_Trip_Delay/(STD_FM_Div * I2C_Clk_Period) = 1 (Rounded down)

So, I^2C SCL rate = 408 MHz / (5 * (8+1) * 10) = 906 kHz

If de-bounce is disabled, the Round_Trip_delay (in I2C_Div cycles) becomes less than 1.

So, I^2C SCL rate = 408 MHz / (5 * 8 * 10) = 1002 kHz = 1.02 MHz

9.6.3.1.2 I2C Command ADDR Registers (used in Normal Mode Only)

These registers, I2C_CMD_ADDR0 and I2C_CMD_ADDR1, contain the address of the slave with which a transaction is intended. The I^2C Master Controller can be programmed to transact with both 7-bit and 10-bit addressed slaves. Bit [0] of the I2C_CNFG register determines the size of the slave address to be transacted. If a 7-bit or a 10-bit slave address is chosen, the respective address is taken from I2C_CMD_ADDR0 [9:0], and the Read/Write command is taken from bits 6 and 7 of the I2C_CNFG Register. In a 2-Slave configuration, the slave 1 address (7-bit or 10-bit) is taken from I2C_CMD_ADDR0 [9:0] and Slave 2 address (7-bit or 10-bit) from I2C_CMD_ADDR1 [9:0]. The transfer length is the same in 2-slave operation and is taken from the I2C_CNFG [LENGTH] bit field. In 2-slave operation, the maximum possible transfer size is 4 bytes.

9.6.3.1.3 I2C Data Registers (used in Normal Mode Only)

There are two data registers, I2C_CMD_DATA1 and I2C_CMD_DATA2, each of 32 bit length, which are to be loaded with the data to be transmitted or received as an I^2C Master. When being used as I^2C slave, the controller uses a separate byte-wide register to store the data to be transmitted or received (I2C_SL_RCVD register).

9.6.3.1.4 I2C Configuration Register

This register is to be configured with the following data:

- Number of bytes to be transmitted or received
- Select for 7-bit or 10-bit slave address
- Program to send a Start Byte
- Single or two slave operations
- Support of NOACK from an external peripheral

The I2C_CNGF [9] bit is used to issue a "Begin-Transaction" command to the I^2C Master Controller. This bit is auto cleared by the hardware. The firmware should first configure all the other registers and bits [8:0] of the I2C_CNFG register before the I2C_CNFG [9] bit is programmed to one.

9.6.3.1.5 I2C Configuration Load Register

This CONFIG load register transfers the software programmed configuration in the I^2C registers to hardware internal registers used in the logic. It has three bit fields:

1. MSTR_CONFIG_LOAD for regular master and Bus Clear master logic
2. SLV_CONFIG_LOAD for slave controller logic
3. TIMEOUT_CONFIG_LOAD for SMBUS timeout logic.

Software must set these fields to 1 for the actual register configuration to take effect. Thus software is programming only shadow registers through regular configuration. When these load_config bit fields are set to 1, it causes the regular/shadow registers configuration to be transferred to the hardware internal active registers. So software has to set these bit fields at the end of all regular register configurations. However, in normal or non-packet mode, these bit fields should be set to 1 before the I2C_I2C_CNFG_0[SEND] bit is set to 1.

The I2C_I2C_CNFG_0[SEND] bit triggers a transfer on the I^2C bus and should be programmed at the end of the entire configuration. These config_load bits are hardware auto-clear bits. Hardware clears these bit fields once the register configuration is moved to hardware internal active registers. Software must wait until these bits are auto-cleared before continuing with further programming. The wait time is not more than 100 ns for apb_clk = 100 MHz and i2c_clk = 100 MHz. When apb_clk = 20 MHz and i2c_clk = 20 MHz, it is not more than 500 ns time.

Below are the register bit fields associated with each CONFIG_LOAD bit.

MSTR_CONFIG_LOAD needs to be set for the following registers:

- I2C_I2C_CNFG_0
- I2C_I2C_CMD_ADDR0_0
- I2C_I2C_CMD_ADDR1_0
- I2C_I2C_CMD_DATA1_0

- I2C_I2C_CMD_DATA2_0
- I2C_I2C_CLK_DIVISOR_REGISTER_0
- I2C_I2C_BUS_CLEAR_CONFIG_0
- I2C_I2C_INTERFACE_TIMING_0_0
- I2C_I2C_INTERFACE_TIMING_1_0
- I2C_I2C_HS_INTERFACE_TIMING_0_0
- I2C_I2C_HS_INTERFACE_TIMING_1_0

SLV_CONFIG_LOAD covers the following registers:

- I2C_I2C_SL_CNFG_0
- I2C_I2C_SL_ADDR1_0
- I2C_I2C_SL_ADDR2_0
- I2C_I2C_TLOW_SEXT_0[RST_SL_ON_TIMEOUT]
- I2C_I2C_SL_DELAY_COUNT_0

TIMEOUT_CONFIG_LOAD covers the following register bit fields:

- I2C_I2C_TLOW_SEXT_0[TIMEOUT]
- I2C_I2C_TLOW_SEXT_0[TLOW_SEXT]
- I2C_I2C_TLOW_SEXT_0[TLOW_MEXT]
- I2C_I2C_TLOW_SEXT_0[TIMEOUT_EN]
- I2C_I2C_TLOW_SEXT_0[TLOW_SEXT_EN]
- I2C_I2C_TLOW_SEXT_0[TLOW_MEXT_EN]

9.6.3.1.6 I2C Controller Status Register

This register (I2C_STATUS) gives the status of the I^2C Master operation. It includes the busy status of the I^2C controller and the completion status of the command executed.

9.6.3.1.7 I2C Interface Timing Registers

These registers provide flexibility to tune the I^2C interface bus timing, if needed. Separate bit fields are provided for non-HS (STD/FM/FM+) and HS modes timing.

9.6.3.1.8 I2C Master Controller Soft Reset Register

This register has a SOFT_RESET bit field that provides the option to reset the master controller without losing the register configuration. This does not impact the slave controller. If the software finds a requirement to reset the master controller alone, this bit field is useful. Master and slave controllers can be used simultaneously in an application. Writing 1 to I2C_I2C_MASTER_RESET_CNTRL_0[SOFT_RESET] resets all internal states of the master logic,

including FIFOs. Clear this bit to 0 for normal operation. For I^2C applications where Timeout logic is not used, the software needs to wait for 2 µs after assertion and de-assertion of this soft reset. For SMBus applications, wait for 5 ms after assertion and deassertion of soft reset.

9.6.3.1.9 I2C Slave ADDR Registers

Two slave ADDR registers are used to configure the address of the internal slave, I2C_SL_ADDR1 and I2C_SL_ADDR2. The bit [0] of I2C_SL_ADDR2 determines the address-size of the internal slave. When this bit is programmed to 0, the 7-bit slave address is taken from I2C_SL_ADDR1 [6:0]. When this bit is programmed to 1, the most significant two bits of the 10-bit slave address are taken from I2C_SL_ADDR2[2:1] and the least significant bits are taken from I2C_SL_ADDR1[7:0].

9.6.3.1.10 I2C Controller Slave Status Register

This register (I2C_SL_STATUS) contains the status of an I^2C slave.

It has the following status bits:

- Bit 1: Direction of transfer
- Bit 2: New Transaction Received
- Bit 3: Receive Interrupt
- Bit 4: END_TRANS; this bit is set when an I^2C transfer is complete (as indicated by either a STOP or NoACK from master).
- Bit 5: RST_SL; the byte received following the receipt of general call address is 0x06. Therefore programmable part of the slave address needs to be reset and reprogrammed.
- Bit 6: REPROG_SL; the byte received following the receipt of general call address is 0x04. Therefore reprogram the programmable part of the slave address.
- Bit 7: HW_MSTR_INT; when set this bit indicates that master address has been transmitted after general call address.
- Bits [14:8]: Hardware master address; the contents of this field are meaningful only when bit 7 (HW_MSTR_INT) is set.

Note that the I^2C controller generates an interrupt at the completion of each data byte received. In response to this interrupt, firmware must read the I2C_SL_STATUS register to confirm the cause of the interrupt and the direction of data transfer. If the interrupt is due to a data byte received (R/W=0), firmware must read the data register and write 1 to I2C_SL_STATUS [SL_IRQ] to clear the interrupt and get the SCL line released. If the interrupt is due to read operation (R/W=1), data should be written to I2C_SL_RCVD in order to release the SCL line from clock stretching. The I^2C controller delays the release of the ACK handshake on the I^2C bus until the SL.IRQ bit has been cleared by the firmware. The software has to write 1 to clear SL_STATUS register bits.
I2C_SL_STATUS needs to be either polled or the I^2C interrupt must be enabled for transfers to occur in slave mode.

If the device slave is receiving data:

- I2C_SL_STATUS.SL_IRQ bit set indicates that byte has been received.
- Check the transfer direction from the RNW field.
- Read data byte from I2C_SL_RCVD register.
- Write 1 to I2C_SL_STATUS [SL_IRQ] to clear SL_IRQ which causes the bus to be released.
- SL_IRQ is set again upon reception of next byte.

If the device slave is transmitting data:

- I2C_SL_STATUS.SL_IRQ bit set indicates that address has been received.
- Check the transfer direction from the RNW field.
- Write 1 to I2C_SL_STATUS [SL_IRQ] to clear SL_IRQ.
- Write data byte to I2C_SL_RCVD register. This write causes the bus to be released.
- SL_IRQ is set again upon transfer of the byte. Another byte can be written to continue the operation.

The slave stretches the SCL line in the following cases:

- Pending Interrupts from I2C_SL_STATUS register after a byte received in write or a data byte needs to be sent in read operation. The software has to clear them by writing 1 to the reg bit fields, which releases the SCL line.
- TX-FIFO is empty and no data is available to send to the master during a read operation. The software to write data to TX-FIFO to get the SCL line released.
- RX-FIFO is full and no room remains in the FIFO to receive any more data from the master. The software to read the RX-FIFO to allow the SCL line released.
- During address cycle, the SCL stretch occurs after the ACK bit. The stretch time includes the interrupt clear time and the SL_DELAY_COUNT. If it is a read operation, the clock stretch is extended until the data byte is written to the I2C_SL_RCVD register.
- During data phase, the SCL stretch occurs before and after the ACK bit. The stretch before ACK is for a duration of SL_DELAY_COUNT and the stretch time after the ACK bit is the interrupt clear time plus the SL_DELAY_COUNT period. If it is a read operation, the clock stretch is extended until the data byte is written to the I2C_SL_RCVD register.

9.6.3.1.11 Clock Gating Over Ride Enable Register

The second level clock gating improves power saving (SLCG is implemented for both master and slave operations). Clock gating is enabled by default. For any reason, if clock gating needs to be disabled, it can be done through a programmable I2C_CLKEN_OVERRIDE register. Separate bit fields are provided for the master, bus clear master, and slave controllers.

9.6.3.1.12 Interrupt Generation in Non-Packet (Normal) Mode

In non-packet mode, the I^2C controller generates interrupts upon the completion of transmission or reception of the specified number of bytes. A Master interrupt is generated when the number of bytes of the transaction, as programmed in the LENGTH field of the I2C_CNFG register, is complete.

A Slave-interrupt is generated at the transmission/reception of each byte of data including address byte by the internal slave. Slave interrupts are to be cleared by the firmware as mentioned above.

9.6.3.2 Programming Guidelines for Master in Normal Mode

9.6.3.2.1 Master-Transmit Programming Model

This programming example is for 7-byte writes from master to an external slave:

- Program the I2C_CLK_DIVISOR register to get the required data rate based on I2C CLK_SOURCE register programming in CAR module
- Write the slave address in I2C_CMD_ADDR0 register based on 7-bit/10-bit addressing mode
- Write the first 4 bytes data in I2C_CMD_DATA1 register
- Write the remaining 3 bytes in I2C_CMD_DATA2 register
- Program I2C_CNFG[DEBOUNCE_CNT] to the required value
- Set I2C_CNFG[A_MOD] to 7-bit or 10-bit addressing based on slave device address width
- Program I2C_CNFG [LENGTH]. This bit field works in (n+1) fashion so for a 7 byte transfer, 6 needs to be programmed in the LENGTH bit field
- Set I2C_CNFG [SLV2] = 0 as this is one is slave access
- Set CMD1 = 0 for write operation
- Set NOACK to 0 or 1 based on Slave type
- Set the MSTR_CONFIG_LOAD bit in I2C_CONFIG_LOAD register
- Wait until the I2C_CONFIG_LOAD[MSTR_CONFIG_LOAD] is auto-cleared by hardware to 0
- Finally, set I2C_CNFG [SEND] to 1 to begin a write transaction on the interface
- Wait until the transaction is complete – either wait until interrupt is received or I2C_STATUS[BUSY] bit becomes zero.
- Check I2C_STATUS [CMD1_STAT] to see if the transaction is successful or a NOACK from the slave for any of the bytes transferred.

9.6.3.2.2 Master-Receive Programming Model

The read operation is divided into two steps based on random read or immediate read after a write. In a typical random read:

1. Write command: Send slave register/memory index first from which the data needs to be read
2. Send read command to read the data

The above operation can be done in two separate configurations or in a single configuration using Repeated Start in 2-slave config

1. Write: Sending slave's 1-byte register index first
   - Write the slave address in I2C_CMD_ADDR0 register based on 7-bit /10-bit addressing mode
   - Write the slave's internal register index (from which the data has to be read) in I2C_CMD_DATA1 register
   - Program I2C_CNFG[DEBOUNCE_CNT] to the required value
   - Set I2C_CNFG[A_MOD] to 7-bit or 10-bit addressing based on slave device address width
   - Program I2C_CNFG[LENGTH] = 0 for a 1-byte register index transfer
   - Set I2C_CNFG[SLV2] = 0
   - Set I2C_CNFG[CMD1] = 0 for write operation
   - Set I2C_CNFG[NOACK] to 0 or 1 based on Slave type
   - Set the MSTR_CONFIG_LOAD bit in I2C_CONFIG_LOAD register
   - Wait until I2C_CONFIG_LOAD[MSTR_CONFIG_LOAD] is auto-cleared by hardware to 0
   - Finally, set I2C_CNFG[SEND] to begin write transaction on the interface
   - Wait until the transaction is complete – either wait until interrupt is received or I2C_STATUS [BUSY] bit becomes zero.
   - Check I2C_STATUS[CMD1_STAT] to see if the transaction is successful or a NOACK from the slave
2. Read: Send read command
   - Program I2C_CNFG[LENGTH] = 6 for 7-bytes read
   - Set I2C_CNFG[SLV2] = 0 as this is one slave access
   - Set I2C_CNFG[CMD1] = 1 for read operation
   - Set the MSTR_CONFIG_LOAD bit in I2C_CONFIG_LOAD register
   - Wait until I2C_CONFIG_LOAD[MSTR_CONFIG_LOAD] is auto-cleared by hardware to 0
   - Finally, set I2C_CNFG[SEND] to begin write transaction on the interface
   - Wait until the transaction is complete – either wait until interrupt is received or I2C_STATUS [BUSY] bit becomes zero.
   - Check I2C_STATUS[CMD1_STAT] to see if the transaction is successful or a NOACK from the slave
   - Read the first 4 bytes data in the I2C_CMD_DATA1 register
   - Read the remaining 3 bytes in the I2C_CMD_DATA2 register

Read using repeated-start in 2-slave mode. Note that maximum transfer size possible in 2-slave mode is 4 bytes.

- Write the slave address in the I2C_CMD_ADDR0 register with the LSB bit set to “0”
- Write the slave address in the I2C_CMD_ADDR1 register with the LSB bit set to “1”
- Write the slave’s internal register index byte (from which the data has to be read) in I2C_CMD_DATA1 register
- Program I2C_CNFG[DEBOUNCE_CNT] to the required value as needed
- Set I2C_CNFG[A_MOD] to 7-bit or 10-bit addressing
- Program I2C_CNFG[LENGTH] = 0 for 1 byte transfer
- Set I2C_CNFG[SLV2] = 1
- Set I2C_CNFG[CMD1] = 0 for write operation
- Set I2C_CNFG[CMD2] = 1 for read operation
- Set I2C_CNFG[NOACK] to 0 or 1 based on the Slave type
- Set the MSTR_CONFIG_LOAD bit in the I2C_CONFIG_LOAD register
- Wait until I2C_CONFIG_LOAD[MSTR_CONFIG_LOAD] is auto-cleared by hardware to 0
- Finally, set I2C_CNFG[SEND] to begin write transaction on the interface
- Wait until the transaction is complete – either wait until interrupt is received or I2C_STATUS [BUSY] bit becomes zero.
- Check I2C_STATUS [CMD1_STAT] to see if the transaction is successful or a NOACK from the slave
- Read data from the I2C_CMD_DATA2 register

9.6.3.3 Programming Guidelines for Packet-Based Interface

9.6.3.3.1 Packet-Based Interface Registers

The following registers are used for a packet-based interface:

- I2C_I2C_MASTER_RESET_CNTRL_0
- I2C_I2C_TX_PACKET_FIFO_0
- I2C_I2C_RX_FIFO_0
- I2C_PACKET_TRANSFER_STATUS_0
- I2C_MST_PACKET_TRANSFER_CNT_STATUS_0
- I2C_I2C_HS_INTERFACE_TIMING_0_0
- I2C_I2C_HS_INTERFACE_TIMING_1_0
- I2C_I2C_INTERFACE_TIMING_0_0
- I2C_I2C_INTERFACE_TIMING_1_0
- I2C_MST_FIFO_CONTROL_0
- I2C_MST_FIFO_STATUS_0
- I2C_INTERRUPT_MASK_REGISTER_0
- I2C_INTERRUPT_STATUS_REGISTER_0

- I2C_I2C_INTERRUPT_SOURCE_REGISTER_0
- I2C_I2C_INTERRUPT_SET_REGISTER_0
- I2C_I2C_CLKEN_OVERRIDE_0
- I2C_I2C_DEBUG_CONTROL_0
- I2C_I2C_CLK_DIVISOR_REGISTER_0
- I2C_I2C_CNFG_0
- I2C_I2C_CONFIG_LOAD_0

All other registers/fields have no meaning in packet mode and should be used only in normal mode.

9.6.3.3.2 Packet Header and Payload Programming

Software should post the packets to the I2C_TX_PACKET_FIFO register. It writes the packet header followed by the payload to the I2C_TX_PACKET_FIFO register. The header size can vary from three to four words. For I^2C, the size is a three-word header for request packets and four words for response packets. The first two words of the header contain a generic header. The third word of the packet (and fourth as well for response packets) contains I^2C transaction-specific information. The payload contains the actual data to be written to the slave. In case of read operations, payload is nil, and hence, the packet contains only a header.

9.6.3.3.3 Reading from the I2C_RX_FIFO

The data received from the I^2C bus is pushed into the I2C_I2C_RX_FIFO_0 register. In PIO mode, a request interrupt is generated when the FIFO attention level is reached, and the corresponding interrupt enable bit is set. Software then reads the I2C_I2C_RX_FIFO_0 register to get the data. In DMA mode, a request is asserted to DMA after the attention level is reached.

9.6.3.3.4 REPEAT_START/STOP Bit in the Protocol Specific Header

This field indicates whether or not to put a stop or repeated-start condition after the current transaction. By default, this bit is zero and a stop condition is put on the bus. If this bit is set to 1, a repeat start is put on the bus before proceeding with the next packet. The REPEAT_START/STOP bit can be used for combining read operations and write operations within a single transaction with a repeat start, or to do transfers beyond the 64 Kbyte limit. This bit is present in the protocol-specific header.

9.6.3.3.5 ContinueXfer Bit in the Protocol Specific Header

The I^2C controller supports transfer sizes beyond 64 Kbytes without a repeat start condition, again by combining multiple packets. This is in addition to the ability to use a repeat start condition to do continuous transfers over 64 Kbyte limit. With the ContinueXfer field, the current transfer can continue without a stop or a repeat start condition.

9.6.3.3.6 Transfers with START Byte

Transfers needing a longer start procedure can use the START Byte feature. In this mode, START Byte (8'b00000001) is transmitted after START condition and a dummy acknowledge (NACK) would be received. The actual transfer starts with a Repeated START condition then. For START byte transfers, set SEND_START_BYTE to 1 in the protocol specific header of the packet.

9.6.3.3.7 Transfers with Slaves Having No-ACK Capability

There could be slave devices which do not generate ACK upon reception of a byte. The I^2C Master has the capability to do the transfers with these devices too. This would be done by setting CONTINUE_ON_NACK bit field of the protocol specific header to 1. In this case, the I^2C master ignores the NACK bit and continues the transfers.

9.6.3.3.8 HS_MODE Transfers

HS_MODE transfers are supported in packet mode only. For HS transfers, write HS master code byte in HS_MASTER_ADDR field of the protocol specific header and set HS_MODE bit to 1. If HS_MODE bit is programmed to 0, all the transfers would be done in Non-HS modes (Standard/Fm/Fm+) based on the I^2C clocks frequency programming. In HS mode, Master code byte is transmitted in (Standard/Fm/Fm+) speed mode and the data transfers would happen in HS mode. I2C_I2C_HS_INTERFACE_TIMING_0_0 and I2C_I2C_HS_INTERFACE_TIMING_1_0 registers would be used for HS timing generation on the I^2C bus.

9.6.3.3.9 Master FIFO Control

In DMA mode, I2C_MST_FIFO_CONTROL_0[TX_FIFO_TRIG] indicates the number of words that need to be empty in I2C_I2C_TX_PACKET_FIFO_0 for the DMA trigger to be asserted. I2C_MST_FIFO_CONTROL_0[RX_FIFO_TRIG] indicates the number of words that need to be full in I2C_I2C_RX_FIFO_0 for the DMA trigger to be asserted.

In PIO mode, the INTERRUPT_STATUS_REGISTER[TFIFO_DATA_REQ] is be set if the I2C_I2C_TX_PACKET_FIFO_0 empty count is more than the value programmed in TX_FIFO_TRIG. Similarly INTERRUPT_STATUS_REGISTER[RFIFO_DATA_REQ] is set if the I2C_I2C_RX_FIFO_0 full count is more than the value programmed in RX_FIFO_TRIG. The CPU is interrupted if the status bits are set and their corresponding INT_EN bit is set in the Interrupt Mask Register. The depth of the TX and RX FIFOs is 128 words.

For Transmit and Receive Data requests in DMA or PIO mode, the I2C_I2C_CNFG_0[PACKET_MODE_EN] bit needs to be set to 1 along with the FIFO_TRIG level settings. Otherwise, requests are not generated.

This register has separate FLUSH control bits for the Master controller TX_FIFO and RX_FIFO. Writing 1 to these bits initiates the FIFO flush operation. They are auto-cleared to 0 once the flush operation is complete.

9.6.3.3.10 Master FIFO Status

This register indicates the number of entries that are empty in the Master controller TX_FIFO (bit field: I2C_MST_FIFO_STATUS_0[TX_FIFO_EMPTY_CNT]) and the number of entries that are full in the RX_FIFO (bit field: I2C_MST_FIFO_STATUS_0[RX_FIFO_FULL_CNT]).

9.6.3.3.11 Slave FIFO Control

I2C_SLV_FIFO_CONTROL_0[TX_FIFO_TRIG] indicates the number of words that need to be empty in I2C_I2C_SLV_TX_PACKET_FIFO_0 for the data transmit request to be asserted. I2C_SLV_FIFO_CONTROL_0[RX_FIFO_TRIG] indicates the number of words that need to be full in I2C_I2C_SLV_RX_FIFO_0 for the received data request to be asserted.

In PIO mode, INTERRUPT_STATUS_REGISTER[SLV_TFIFO_DATA_REQ] is set if the I2C_I2C_SLV_TX_PACKET_FIFO_0 empty count is more than the value programmed in TX_FIFO_TRIG. Similarly INTERRUPT_STATUS_REGISTER [SLV_RFIFO_DATA_REQ] is set if I2C_I2C_SLV_RX_FIFO_0 full count is more than the value programmed in RX_FIFO_TRIG. CPU is interrupted if status bits are set and their corresponding INT_EN is set in the Interrupt Mask Register. The depth of Tx and Rx FIFOs is 128 words.

For Transmit and Receive Data requests in FIFO mode, I2C_I2C_SL_CNFG_0[FIFO_XFER_EN] bit needs to be set to 1 along with FIFO_TRIG level settings. Otherwise, requests are not generated.

This register has separate FLUSH control bits for the Slave controller TX_FIFO and RX_FIFO. Write 1 to these bits initiates the FIFO flush operation. They are auto-cleared to 0 once the flush operation is complete.

9.6.3.3.12 Slave FIFO Status

This register indicates the number of entries that are empty in the Slave controller TX_FIFO (bit field: I2C_SLV_FIFO_STATUS_0[TX_FIFO_EMPTY_CNT]) and the number of entries that are full in the RX_FIFO (bit field:I2C_SLV_FIFO_STATUS_0[RX_FIFO_FULL_CNT]).

9.6.3.3.13 Interrupt Status Register

This register (INTERRUPT_STATUS_REGISTER) gives the status of the I^2C Master in Packet Mode operation and also the status of Slave controller in FIFO mode and SMBUS timeout status. The register bits are set to 1 when the corresponding event happens. The bit fields are sticky. Once set to 1, they stay 1. Write 1 to clear them to 0. However, TFIFO_DATA_REQ and RFIFO_DATA_REQ fields are exceptions to this as they depend on the FIFO trigger levels and cannot be cleared.

9.6.3.3.14 Interrupt Mask Register

This register (I2C_INTERRUPT_MASK_REGISTER_0) contains the interrupt enable bits for the status bits given in INTERRUPT_STATUS_REGISTER. A separate interrupt enable bit is available for each bit field of INTERRUPT_STATUS_REGISTER. Setting an INT_EN bit in this register enables signaling an interrupt to the system when corresponding status bit is set in the INTERRUPT_STATUS_REGISTER. Clearing it will mask the interrupt.

9.6.3.3.15 Interrupt Source Register

This read-only register (I2C_INTERRUPT_SOURCE_REGISTER) returns the AND of Interrupt Mask and Interrupt Status Registers bit fields. A bit field in this register would be set if the corresponding bit fields in I2C_INTERRUPT_MASK_REGISTER_0 and INTERRUPT_STATUS_REGISTER are set to 1.

Example:

I2C_INTERRUPT_SOURCE_REGISTER[ARB_LOST] = AND(I2C_INTERRUPT_MASK_REGISTER [ARB_LOST_INT_EN], INTERRUPT_STATUS_REGISTER [ARB_LOST])

9.6.3.3.16 Packet Transfer Status

In addition to I2C_INTERRUPT_STATUS_REGISTER_0, packet transfer status can be obtained from I2C_PACKET_TRANSFER_STATUS_0 and I2C_MST_PACKET_TRANSFER_CNT_STATUS_0 registers as well with additional information of the packet. Software can read this register to know which packet is currently being transferred, how many bytes are transferred in the on-going packet or packet is successfully transferred or any Bus lost or NACK happened during the transfer.

- TRANSFER_PKT_ID gives the ID of on-going packet or last transmitted packet ID.
- TRANSFER_BYTENUM in the I2C_MST_PACKET_TRANSFER_CNT_STATUS_0 register gives the information about the number of bytes that have been transferred so far in the current packet. Once the packet transfer is done, this field becomes zero.
- TRANSFER_COMPLETE field would be set when a packet (with STOP=1) is successfully transferred.
- ARB_LOST shows the loss of the bus.
- NOACK_FOR_ADDR for the NOACK received for Address byte.
- NOACK_FOR_DATA for the NOACK received for data byte.
- CONTROLLER_BUSY shows PACKET_MODE_EN bit status. It is 1 as long as packet mode is enabled.

9.6.3.3.17 Error Handling

In case of an error (e.g., NOACK or loss of arbitration, TX_FIFO overflow or RX_FIFO underflow, the following steps describe the sequence that need to be followed before proceeding with any further transactions.

- When an error occurs, the I^2C Master controller stops further packet transfers and sets the corresponding status bits in I2C_INTERRUPT_STATUS_REGISTER for the event (ARB_LOST/ NOACK/TFIFO_OVF/ RFIFO_UNF) occurred.
- PACKET_TRANSFER_STATUS register is updated with the current error packet ID, at which the error is occurred.
  - Handling TFIFO_OVF and RFIFO_UNF events
    - TFIFO_OVF or RFIFO_UNF can potentially lead to I^2C controller hang as there would be data drop due to FIFO overflow or packets are combined with Repeated Start in multi-packet transfers.
    - TX_FIFO overflow or Rx underflow happens when software writes to an already full TX_FIFO or reading an empty RX_FIFO, **software must avoid these conditions**.
    - The controller reset (from CAR) is required to recover from a I^2C-bus hang; reset results in the Master controller losing the Bus Busy status information (this is a problem in multi-master mode).
  - Handling ARB_LOST and NOACK events
    - If DMA is used for the transfers, disable the DMA and wait until DMA becomes idle.
    - Clear the I^2C FIFOs entries using FIFO FLUSH bits
    - Clear the ARB_LOST and NOACK bits in I2C_INTERRUPT_STATUS_Register
    - After completing the above, the I^2C controller is ready for the transfers again
    - Restart the DMA if used
    - Start new transfer following regular programming guidelines

9.6.3.3.18 Slave ACK WithHold Feature in FIFO Mode

In FIFO mode, slave holds the I^2C bus by pulling SCL line low (clock stretching) after sending the ACK for the very last byte of transfer. Register bit fields ACK_LAST_BYTE and ACK_LAST_BYTE_VALID fields are used for this purpose. When Slave receives its Rx Buffer full data, it holds the I^2C bus on the very last byte of the transfer. Software can ACK or NACK the transfer by setting the ACK_LAST_BYTE bit field to the required value. It has to set ACK_LAST_BYTE_VALID bit field next to say that ACK_LAST_BYTE bit field has been updated. Slave sends the ACK/NACK to the interface for the next byte accordingly. Until that time, Slave will hold the I^2C bus by doing clock stretching.

In SMBUS applications, Host (Software) is responsible for providing this indication to I^2C slave in a timely fashion before timeouts happen.

9.6.3.3.19 Slave Usage in FIFO Mode

Slave data transfers can be done using FIFO. Data can be transmitted and received through I2C_I2C_SLV_TX_PACKET_FIFO_0 and I2C_I2C_SLV_RX_FIFO_0 registers respectively. For this, FIFO_XFER_EN bit in I2C_I2C_SL_CNFG_0 register needs to be set.

In FIFO-XFER mode, slave by default works in one byte mode, i.e., initially when a single byte is received, I^2C bus is held and SLV_RX_BUFFER_FILLED interrupt is generated. Software should indicate whether to ACK or not to the next byte in the transfer by writing into ACK_LAST_BYTE field of I2C_I2C_SL_CNFG_0 register. Note that ACK_LAST_BYTE_VALID field also needs to be set to indicate to hardware that ACK_LAST_BYTE is being updated. When software receives interrupt due to SLV_RX_BUFFER_FILLED, it can check ACK_WITHHELD of I2C_I2C_SLV_PACKET_STATUS_0 register to see if slave is withholding the ACK for the last byte.

Software can update the buffer size from default to any number by programming I2C_SLV_PAYLOAD_0[BUFFER_SIZE].

9.6.3.4 Master in Packet Mode Programming Guidelines

9.6.3.4.1 Programming Model for Master-Transmit, Interrupt Method

Initial Configuration

Before accessing the I^2C controller, set up the following:

- I^2C clocks programming
- Pinmux programming
- I^2C controller reset release

I2C Controller Registers Initialization

- Write I2C_I2C_MASTER_RESET_CNTRL_0[SOFT_RESET] = CLEARED
- Write I2C_I2C_TLOW_SEXT_0 = 0x0
- Write I2C_I2C_CMD_ADDR0_0 = 0x0
- Write I2C_I2C_CMD_ADDR1_0 = 0x0
- Write I2C_I2C_CMD_DATA1_0 = 0x0
- Write I2C_I2C_CMD_DATA2_0 = 0x0
- Write I2C_I2C_CLKEN_OVERRIDE_0 = 0x0 (for single master bus); 0x1 (for multi-master bus)
- Write I2C_I2C_DEBUG_CONTROL_0 = 0x0
- Write I2C_I2C_INTERRUPT_SET_REGISTER_0 = 0x0

Program I2C_I2C_CLK_DIVISOR_REGISTER_0 Register Based on the Required I^2C-Bus Speed

Refer to the I^2C Frequency Divisor Register section for details.

Configure I2C Interface Timing Registers

Recommendation is to use default, but if needed these registers can be programmed to achieve the required I^2C interface timing.

- Write I2C_I2C_INTERFACE_TIMING_0_0 = 0x0204
- Write I2C_I2C_INTERFACE_TIMING_1_0 = 0x04070404
- Write I2C_I2C_HS_INTERFACE_TIMING_0_0 = 0x0308
- Write I2C_I2C_HS_INTERFACE_TIMING_1_0 = 0x0b0b0b

Configure FIFO Trigger Levels for Data Requests in PIO Mode

Below settings are for example only. Program this bit field to the required trigger levels.

- Write I2C_FIFO_CONTROL_0[TX_FIFO_TRIG] = 0x3 (0x3 = System gets request from the I^2C controller when at least four words are empty in the I2C_I2C_TX_PACKET_FIFO_0).

Configure I2C Master to Packet Mode

- Write I2C_I2C_CNFG_0[MULTI_MASTER_MODE] = DISABLE (for single-master bus); ENABLE (for multi-master bus)
- Write I2C_I2C_CNFG_0[HS_RND_TRIP_DLY_EFFECT] = DISABLE
- Write I2C_I2C_CNFG_0[MSTR_CLR_BUS_ON_TIMEOUT] = 0
- Write I2C_I2C_CNFG_0[DEBOUNCE_CNT] = 0x2 Example only, program as required. Refer to the I^2C Frequency Divisor Register section for details
- Write I2C_I2C_CNFG_0[NEW_MASTER_FSM] = ENABLE
- Write I2C_I2C_CNFG_0[PACKET_MODE_EN] = GO
- Write I2C_I2C_CNFG_0[SEND] = NOP
- Write I2C_I2C_CNFG_0[NOACK] = DISABLE
- Write I2C_I2C_CNFG_0[CMD2] = DISABLE
- Write I2C_I2C_CNFG_0[CMD1] = DISABLE
- Write I2C_I2C_CNFG_0[START] = DISABLE
- Write I2C_I2C_CNFG_0[SLV2] = DISABLE
- Write I2C_I2C_CNFG_0[LENGTH] = 0
- Write I2C_I2C_CNFG_0[A_MOD] = 0

Activate the Configuration

- I2C_I2C_CONFIG_LOAD_0[TIMEOUT_CONFIG_LOAD] = ENABLE

- I2C_I2C_CONFIG_LOAD_0[SLV_CONFIG_LOAD] = DISABLE
- I2C_I2C_CONFIG_LOAD_0[MSTR_CONFIG_LOAD] = ENABLE
- Wait for I2C_I2C_CONFIG_LOAD_0[TIMEOUT_CONFIG_LOAD] and I2C_I2C_CONFIG_LOAD_0[MSTR_CONFIG_LOAD] bits to get cleared by the hardware

Enable Interrupts for I2C Master Controller in Packet Mode

- Write I2C_INTERRUPT_MASK_REGISTER_0[ARB_LOST_INT_EN] = ENABLE
- Write I2C_INTERRUPT_MASK_REGISTER_0[NOACK_INT_EN] = ENABLE
- Write I2C_INTERRUPT_MASK_REGISTER_0[TFIFO_OVF_INT_EN] = ENABLE
- Write I2C_INTERRUPT_MASK_REGISTER_0[ALL_PACKETS_XFER_COMPLETE_INT_EN] = ENABLE
- Write I2C_INTERRUPT_MASK_REGISTER_0[PACKET_XFER_COMPLETE_INT_EN] = ENABLE
- Write I2C_INTERRUPT_MASK_REGISTER_0[TFIFO_DATA_REQ_INT_EN] = ENABLE

This completes the configuration. Next step is posting the packets to the I^2C controller to initiate data transfers on the I^2C-bus.

Packets Posting Based on TFIFO Data Requests

1. There could be one packet or more than one packet that need to be transferred from I^2C.
2. One packet transfer size(in words) = packet_header (in words) +payload_size (in words)
3. Total transfer size = packet**1**_transfer_size + packet**2**_transfer_size +..... +packet**N**_transfer_size
4. When the TX_FIFO attention level is reached, I^2C sets the I2C_INTERRUPT_STATUS_REGISTER_0[TFIFO_DATA_REQ] bit and sends an interrupt to system if the interrupt is enabled with I2C_INTERRUPT_MASK_REGISTER_0[TFIFO_DATA_REQ_INT_EN] = ENABLE setting

Transmit Data Request Interrupt Handling

1. Read I2C_I2C_INTERRUPT_SOURCE_REGISTER_0 register and check which bit fields are set to 1.
2. For master transmit data requests, I2C_I2C_INTERRUPT_SOURCE_REGISTER_0[TFIFO_DATA_REQ] would be set to 1.
3. Disable the interrupt by writing I2C_INTERRUPT_MASK_REGISTER_0[TFIFO_DATA_REQ_INT_EN] = DISABLE
4. Write the packet words to I2C_I2C_TX_PACKET_FIFO_0. The total words size that would be written for every TFIFO_DATA_REQ request should be less than or equal to the FIFO trigger level. Otherwise, it causes the TX_FIFO to overflow.
5. Decrement the total transfer size with the number of words written to the FIFO in the previous step (total_transfer_size = total_transfer_size - #words written to TX_FIFO). And check if the new transfer size is zero. If not zero, enable interrupt for TFIFO_DATA_REQ with

I2C_INTERRUPT_MASK_REGISTER_0[TFIFO_DATA_REQ_INT_EN] = ENABLE and wait for the new interrupt.

The above steps continue until all the packets are posted to the I2C_I2C_TX_PACKET_FIFO_0.
Packet parsing: once a packet is available in the TX_FIFO, I^2C parses it and starts doing transfers on the I^2C bus. The transfers could be successful or unsuccessful. Following sections describe how to identify and handle such transfers.

Successful Transfers

1. Once a packet is transmitted successfully by the I^2C Master, I2C_INTERRUPT_STATUS_REGISTER_0[PACKET_XFER_COMPLETE] bit is set to 1 to reflect the status. And if the packet header is configured to put STOP condition on the bus as the last packet of that transfer, I2C_INTERRUPT_STATUS_REGISTER_0[ALL_PACKETS_XFER_COMPLETE] bit is also set.
2. If interrupt enables are set, corresponding bits would be set in I2C_I2C_INTERRUPT_SOURCE_REGISTER_0 register and signals an interrupt to the system.

Interrupt Handling for Successful Transfers

1. Read I2C_I2C_INTERRUPT_SOURCE_REGISTER_0 register and check which bit fields are set to 1.
2. If a packet is transmitted successfully, I2C_I2C_INTERRUPT_SOURCE_REGISTER_0 [PACKET_XFER_COMPLETE] would be set. If the packet was configured to have STOP bit, I2C_I2C_INTERRUPT_SOURCE_REGISTER_0[ALL_PACKETS_XFER_COMPLETE] would also be set.
3. Clear them by writing 1 to I2C_INTERRUPT_STATUS_REGISTER_0[PACKET_XFER_COMPLETE] and I2C_INTERRUPT_STATUS_REGISTER_0[ALL_PACKETS_XFER_COMPLETE] bits accordingly. This clears the interrupt.

Transfers Completion

1. Check DMA status to know if all the packets are transferred to the I^2C controller from memory.
2. Use PACKET_XFER_COMPLETE (if packets have RepeatedStart or ContinueXfer settings) or ALL_PACKETS_XFER_COMPLETE alerts (if packets have only STOP) to track how many packets are successfully transferred to the I^2C bus. When all packets are transferred successfully to the bus, it matches total alerts that I^2C signals to the system.

Unsuccessful Transfers

A packet transfer could be unsuccessful due to various reasons. The I^2C master logs this information in I2C_INTERRUPT_STATUS_REGISTER_0 register. Transfer could be unsuccessful due to I^2C bus related ARB_LOST/NACK events or TX_PACKET_FIFO Overflow errors.

1. I2C_INTERRUPT_STATUS_REGISTER_0[ARB_LOST] is set to 1 if the I^2C master sees the bus is already occupied by some other device when a transfer is initiated or if it loses the bus during the arbitration in multi-master case or if a slave device pulls the SDA line low continuously for some unknown reason.
2. I2C_INTERRUPT_STATUS_REGISTER_0[NOACK] is set to 1 if slave device responds with NACK for address or data byte transfer of a transaction.
3. I2C_INTERRUPT_STATUS_REGISTER_0[TFIFO_OVF] is set to 1 if DMA or CPU writes to I2C_I2C_TX_PACKET_FIFO_0 even it is full.
4. When any of these error events happens, the I^2C controller stops further packet transfers and flushes the data from the I2C_I2C_TX_PACKET_FIFO_0 register.
5. If interrupt enables are set, corresponding bits would be set in I2C_I2C_INTERRUPT_SOURCE_REGISTER_0 register and signals an interrupt to the system.

Interrupt Handling for Unsuccessful Transfers

1. Read I2C_I2C_INTERRUPT_SOURCE_REGISTER_0 register and check which bit fields are set to 1 (ARB_LOST/NOACK/TFIFO_OVF).
2. In all these cases, an I^2C controller reset is required. Follow the below sequence before going for further transfers.
3. Reset the I^2C controller.
4. Redo the controller configuration and go for the transfers.

Note: I^2C issues the DATA requests to the system based on FIFO trigger levels settings and the I2C_I2C_CNFG_0[PACKET_MODE_EN] setting. PACKET_MODE_EN bit needs to be set to 1 for requests generation.

9.6.3.4.2 Programming Model for Master-Transmit, DMA Method

Initial Configuration

Before accessing I^2C controller, set up the following:

1. I^2C clocks programming
2. Pinmux programming
3. I^2C controller reset release

I^2C Controller Registers Initialization

- Write I2C_I2C_MASTER_RESET_CNTRL_0[SOFT_RESET] = CLEARED
- Write I2C_I2C_TLOW_SEXT_0 = 0x0
- Write I2C_I2C_CMD_ADDR0_0 = 0x0
- Write I2C_I2C_CMD_ADDR1_0 = 0x0

- Write I2C_I2C_CMD_DATA1_0 = 0x0
- Write I2C_I2C_CMD_DATA2_0 = 0x0
- Write I2C_I2C_CLKEN_OVERRIDE_0 = 0x0 (for single master bus); 0x1 (for multi-master bus)
- Write I2C_I2C_DEBUG_CONTROL_0 = 0x0
- Write I2C_I2C_INTERRUPT_SET_REGISTER_0 = 0x0

Program I2C_I2C_CLK_DIVISOR_REGISTER_0 Register Based on the Required I^{2}C Bus Speed

Refer to the I^{2}C Frequency Divisor Register section for details.

Configure I2C Interface Timing Registers

Recommendation is to use default, but if needed these registers can be programmed to achieve the required I^{2}C interface timing.

- Write I2C_I2C_INTERFACE_TIMING_0_0 = 0x0204
- Write I2C_I2C_INTERFACE_TIMING_1_0 = 0x04070404
- Write I2C_I2C_HS_INTERFACE_TIMING_0_0 = 0x0308
- Write I2C_I2C_HS_INTERFACE_TIMING_1_0 = 0x0b0b0b

Configure FIFO Trigger Levels for DMA Requests

Below settings are for example only. Program this bit field to the required trigger levels.

- Write I2C_MST_FIFO_CONTROL_0[TX_FIFO_TRIG] = 0x3

(0x3 = DMA gets request from the I^{2}C controller when at least four words are empty in the I2C_I2C_TX_PACKET_FIFO_0 register).

Enable Interrupts for I2C Master Controller in Packet Mode

- Write I2C_INTERRUPT_MASK_REGISTER_0[ARB_LOST_INT_EN] = ENABLE
- Write I2C_INTERRUPT_MASK_REGISTER_0[NOACK_INT_EN] = ENABLE
- Write I2C_INTERRUPT_MASK_REGISTER_0[TFIFO_OVF_INT_EN] = ENABLE
- Write I2C_INTERRUPT_MASK_REGISTER_0[ALL_PACKETS_XFER_COMPLETE_INT_EN] = ENABLE for tracking packet transfers completion status
- Write I2C_INTERRUPT_MASK_REGISTER_0[PACKET_XFER_COMPLETE_INT_EN] = ENABLE for tracking packet transfers completion status

Post I^{2}C Packets (Headers and Data) to the Memory for the Write Transfers

Refer to the I/O Packet Format section for packet structure. Program RESP_PKT_ENABLE = 0 in the packet header.

Configure DMA for I2C Transfers

DMA can be one of GPCDMA/BPMPDMA/AONDMA/SCEDMA. Check which DMA needs to be used based on the I^2C-bus. Note the following when configuring DMA for I^2C transfers:

1. Set DMA source pointer to the memory location where the I^2C packets are stored.
2. Set DMA destination pointer to I^2C instance base address + I2C_I2C_TX_PACKET_FIFO offset.
3. DMA burst size should match I2C_FIFO_CONTROL_0[TX_FIFO_TRIG] level settings.
4. Configure DMA to Flow control mode.
5. DMA Transfer size should match the total packets size in words. For example, three Write packets. First packet = 3 bytes transfer, second packet = 8 bytes transfer and third packet = 11 bytes transfer. Total size = 4+5+6=15 words.

Configure I2C Master to Packet Mode

- Write I2C_I2C_CNFG_0[MULTI_MASTER_MODE] = DISABLE (for single-master bus); ENABLE (for multi-master bus)
- Write I2C_I2C_CNFG_0[HS_RND_TRIP_DLY_EFFECT] = DISABLE
- Write I2C_I2C_CNFG_0[MSTR_CLR_BUS_ON_TIMEOUT] = 0
- Write I2C_I2C_CNFG_0[DEBOUNCE_CNT] = 0x2. Example only, program as required. Refer to the I^2C Frequency Divisor Register section for details.
- Write I2C_I2C_CNFG_0[NEW_MASTER_FSM] = ENABLE
- Write I2C_I2C_CNFG_0[PACKET_MODE_EN] = GO
- Write I2C_I2C_CNFG_0[SEND] = NOP
- Write I2C_I2C_CNFG_0[NOACK] = DISABLE
- Write I2C_I2C_CNFG_0[CMD2] = DISABLE
- Write I2C_I2C_CNFG_0[CMD1] = DISABLE
- Write I2C_I2C_CNFG_0[START] = DISABLE
- Write I2C_I2C_CNFG_0[SLV2] = DISABLE
- Write I2C_I2C_CNFG_0[LENGTH] = 0
- Write I2C_I2C_CNFG_0[A_MOD] = 0

Activate the Configuration

- I2C_I2C_CONFIG_LOAD_0[TIMEOUT_CONFIG_LOAD] = ENABLE
- I2C_I2C_CONFIG_LOAD_0[SLV_CONFIG_LOAD] = DISABLE
- I2C_I2C_CONFIG_LOAD_0[MSTR_CONFIG_LOAD] = ENABLE

Wait for I2C_I2C_CONFIG_LOAD_0[TIMEOUT_CONFIG_LOAD] and I2C_I2C_CONFIG_LOAD_0[MSTR_CONFIG_LOAD] bits to get cleared by the hardware.

With this, it initiates the write transfers from the I^2C master to the bus. The transfers could be successful or unsuccessful. Following sections describe how to identify and handle such transfers.

Successful Transfers

1. Once a packet is transmitted successfully by the I^2C Master, I2C_INTERRUPT_STATUS_REGISTER_0[PACKET_XFER_COMPLETE] bit is set to 1 to reflect the status. And if the packet header configured to put STOP condition on the bus as the last packet of that transfer, I2C_INTERRUPT_STATUS_REGISTER_0[ALL_PACKETS_XFER_COMPLETE] bit will also be set.
2. If interrupt enables are set, corresponding bits would be set in I2C_I2C_INTERRUPT_SOURCE_REGISTER_0 register and signals an interrupt to the system

Interrupt Handling for Successful Transfers

1. Read I2C_I2C_INTERRUPT_SOURCE_REGISTER_0 register and check which bit fields are set to 1.
2. If a packet is transmitted successfully, I2C_I2C_INTERRUPT_SOURCE_REGISTER_0 [PACKET_XFER_COMPLETE] would be set. If the packet was configured to have STOP bit, I2C_I2C_INTERRUPT_SOURCE_REGISTER_0[ALL_PACKETS_XFER_COMPLETE] would also be set.
3. Clear them by writing 1 to I2C_INTERRUPT_STATUS_REGISTER_0[PACKET_XFER_COMPLETE] and I2C_INTERRUPT_STATUS_REGISTER_0[ALL_PACKETS_XFER_COMPLETE] bits accordingly. This clears the interrupt.

Transfers Completion

1. Check DMA status to know if all the packets are transferred to the I^2C controller from memory.
2. Use PACKET_XFER_COMPLETE (if packets have RepeatedStart or ContinueXfer settings) or ALL_PACKETS_XFER_COMPLETE alerts (if packets have only STOP) to track how many packets are successfully transferred to the I^2C bus. When all packets are transferred successfully to the bus, it matches total alerts that I^2C signals to the system.

Unsuccessful Transfers

A packet transfer could be unsuccessful due to various reasons. The I^2C master logs this information in I2C_INTERRUPT_STATUS_REGISTER_0 register. The transfer could be unsuccessful due to I^2C bus related ARB_LOST/NACK events or TX_PACKET_FIFO Overflow errors.

1. I2C_INTERRUPT_STATUS_REGISTER_0[ARB_LOST] is set to 1 if the I^2C master sees the bus is already occupied by some other device when a transfer is initiated or if it loses the bus during the arbitration in multi-master case or if a slave device pulls the SDA line low continuously for some unknown reason.

2. I2C_INTERRUPT_STATUS_REGISTER_0[NOACK] is set to 1 if slave device responds with NACK for address or data byte transfer of a transaction.
3. I2C_INTERRUPT_STATUS_REGISTER_0[TFIFO_OVF] is set to 1 if DMA or CPU writes to I2C_I2C_TX_PACKET_FIFO_0 even it is full.
4. When any of these error events happens, the I^2C controller stops further packet transfers and flushes the data from the I2C_I2C_TX_PACKET_FIFO_0 register.
5. If interrupt enables are set, corresponding bits would be set in I2C_I2C_INTERRUPT_SOURCE_REGISTER_0 register and signals an interrupt to the system.

Interrupt Handling for Unsuccessful Transfers

1. Read I2C_I2C_INTERRUPT_SOURCE_REGISTER_0 register and check which bit fields are set to 1 (ARB_LOST/NOACK/TFIFO_OVF)
2. In all these cases, an I^2C controller reset is required. Follow the below sequence before going for further transfers
3. Assert I^2C controller reset
4. Disable DMA
5. Release I^2C controller reset
6. Redo the I^2C configuration, including DMA, and go for the transfers.

Notes:

1. I^2C issues the requests to DMA based on FIFO trigger levels settings and the I2C_I2C_CNFG_0[PACKET_MODE_EN] setting. PACKET_MODE_EN bit needs to be set to 1 for requests generation.
2. Packet transfer status is also available in I2C_PACKET_TRANSFER_STATUS_0 and I2C_MST_PACKET_TRANSFER_CNT_STATUS_0 registers. Software can read these registers to know which packet is currently being transferred (TRANSFER_PKT_ID), how many bytes are transferred in the on-going packet (TRANSFER_BYTENUM) or if the previous packet with STOP setting is successfully transferred (TRANSFER_COMPLETE and TRANSFER_PKT_ID) or any ARB_LOST/NACK event happened (ARB_LOST, NOACK_FOR_ADDR/NOACK_FOR_DATA).

9.6.3.4.3 Programming Model for Master-Receive, DMA Method

Initial Configuration

Before accessing an I^2C controller, set up the following:

- I^2C clocks programming
- Pinmux programming
- I^2C controller reset release

I2C Controller Registers Initialization

- Write I2C_I2C_MASTER_RESET_CNTRL_0[SOFT_RESET] = CLEARED
- Write I2C_I2C_TLOW_SEXT_0 = 0x0
- Write I2C_I2C_CMD_ADDR0_0 = 0x0
- Write I2C_I2C_CMD_ADDR1_0 = 0x0
- Write I2C_I2C_CMD_DATA1_0 = 0x0
- Write I2C_I2C_CMD_DATA2_0 = 0x0
- Write I2C_I2C_CLKEN_OVERRIDE_0 = 0x0 (for single master bus); 0x1 (for multi-master bus)
- Write I2C_I2C_DEBUG_CONTROL_0 = 0x0
- Write I2C_I2C_INTERRUPT_SET_REGISTER_0 = 0x0

Program I2C_I2C_CLK_DIVISOR_REGISTER_0 Register Based on the Required I^2C Bus Speed

Refer to the I^2C Frequency Divisor Register section for details.

Configure I2C Interface Timing Registers

Recommendation is to use default, but if needed these registers can be programmed to achieve the required I^2C interface timing.

- Write I2C_I2C_INTERFACE_TIMING_0_0 = 0x0204
- Write I2C_I2C_INTERFACE_TIMING_1_0 = 0x04070404
- Write I2C_I2C_HS_INTERFACE_TIMING_0_0 = 0x0308
- Write I2C_I2C_HS_INTERFACE_TIMING_1_0 = 0x0b0b0b

Configure FIFO Trigger Levels for DMA Requests

Below settings are for example only. Program the bit fields to the required trigger levels.

- Write I2C_MST_FIFO_CONTROL_0[TX_FIFO_TRIG] = 0x1
- Write I2C_MST_FIFO_CONTROL_0[RX_FIFO_TRIG] = 0x3

  (0x1 = DMA gets request from the I^2C controller when at least two words are empty in in the I2C_I2C_TX_PACKET_FIFO_0).
  (0x3 = DMA gets request from the I^2C controller when at least four words are full in the I2C_I2C_RX_FIFO_0 register).

Enable Interrupts for I2C Master Controller in Packet Mode

- Write I2C_INTERRUPT_MASK_REGISTER_0[ARB_LOST_INT_EN] = ENABLE
- Write I2C_INTERRUPT_MASK_REGISTER_0[NOACK_INT_EN] = ENABLE
- Write I2C_INTERRUPT_MASK_REGISTER_0[TFIFO_OVF_INT_EN] = ENABLE

- Write I2C_INTERRUPT_MASK_REGISTER_0[RFIFO_UNF_INT_EN] = ENABLE
- Write I2C_INTERRUPT_MASK_REGISTER_0[ALL_PACKETS_XFER_COMPLETE_INT_EN] = ENABLE for tracking packet transfers completion status.
- Write I2C_INTERRUPT_MASK_REGISTER_0[PACKET_XFER_COMPLETE_INT_EN] = ENABLE for tracking packet transfers completion status.

Prepare and Save I2C Packets in the Memory for the Read Transfers

Refer to the I/O Packet Format section for packet structure. Program RESP_PKT_ENABLE = 0 in the packet header.

Configure DMA for I2C Transfers

DMA can be one of GPCDMA/BPMPDMA/AONDMA/SCEDMA. Check which DMA needs to be used based on the I^2C bus, and for complete DMA programming model, refer to the corresponding DMA programming guidelines document. It is out of scope for this document. Note the following when configuring DMA for I^2C transfers:

For packets transfer from memory to the I^2C controller, if DMA CH0 is used:

- Set DMA CH0 source pointer to the memory location where the I^2C packets are stored
- Set DMA CH0 destination pointer to I^2C instance base-address + I2C_I2C_TX_PACKET_FIFO offset
- DMA burst size should match I2C_FIFO_CONTROL_0[TX_FIFO_TRIG] level settings
- Configure DMA to Flow control mode
- DMA CH0 transfer size should match the total packets size in words. Since it is read packets only, only packet headers would be there. For example, 3 Read packets. Total size = 3+3+3= 9 words

To transfer the received-data from the I^2C controller to memory, if DMA CH1 is used:

- Set DMA CH1 source pointer to I^2C instance base address + I2C_I2C_RX_FIFO_0 offset
- Set DMA CH1 destination pointer to the memory location where the I^2C data needs to be saved.
- DMA CH1 burst size should match I2C_FIFO_CONTROL_0[RX_FIFO_TRIG] level settings
- Configure DMA to Flow control mode
- DMA Transfer size should match the total words that need to be read from I2C_I2C_RX_FIFO_0.

Configure I2C Master to Packet Mode

- Write I2C_I2C_CNFG_0[MULTI_MASTER_MODE] = DISABLE (for single-master bus); ENABLE (for multi-master bus)

- Write I2C_I2C_CNFG_0[HS_RND_TRIP_DLY_EFFECT] = DISABLE
- Write I2C_I2C_CNFG_0[MSTR_CLR_BUS_ON_TIMEOUT] = 0
- Write I2C_I2C_CNFG_0[DEBOUNCE_CNT] = 0x2. Example only, program as required.
- Write I2C_I2C_CNFG_0[NEW_MASTER_FSM] = ENABLE
- Write I2C_I2C_CNFG_0[PACKET_MODE_EN] = GO
- Write I2C_I2C_CNFG_0[SEND] = NOP
- Write I2C_I2C_CNFG_0[NOACK] = DISABLE
- Write I2C_I2C_CNFG_0[CMD2] = DISABLE
- Write I2C_I2C_CNFG_0[CMD1] = DISABLE
- Write I2C_I2C_CNFG_0[START] = DISABLE
- Write I2C_I2C_CNFG_0[SLV2] = DISABLE
- Write I2C_I2C_CNFG_0[LENGTH] = 0
- Write I2C_I2C_CNFG_0[A_MOD] = 0

Activate the Configuration

- I2C_I2C_CONFIG_LOAD_0[TIMEOUT_CONFIG_LOAD] = ENABLE
- I2C_I2C_CONFIG_LOAD_0[SLV_CONFIG_LOAD] = DISABLE
- I2C_I2C_CONFIG_LOAD_0[MSTR_CONFIG_LOAD] = ENABLE
- Wait for I2C_I2C_CONFIG_LOAD_0[TIMEOUT_CONFIG_LOAD] and
- I2C_I2C_CONFIG_LOAD_0[MSTR_CONFIG_LOAD] bits to get cleared by the hardware

With this, it will initiate the read transfers by the I^2C master on the bus. The transfers could be successful or unsuccessful. Following sections describe how to identify and handle such transfers.

Successful Transfers

1. Once a packet transfer is successfully done by the I^2C Master, I2C_INTERRUPT_STATUS_REGISTER_0[PACKET_XFER_COMPLETE] bit is set to 1 to reflect the status. If the packet header configured to put STOP condition on the bus as the last packet of that transfer, I2C_INTERRUPT_STATUS_REGISTER_0[ALL_PACKETS_XFER_COMPLETE] bit will also be set.
2. If interrupt enables are set, corresponding bits would be set in I2C_I2C_INTERRUPT_SOURCE_REGISTER_0 register and signals an interrupt to the system.

Interrupt Handling for Successful Transfers

1. Read I2C_I2C_INTERRUPT_SOURCE_REGISTER_0 register and check which bit fields are set to 1.
2. If a packet is transmitted successfully, I2C_I2C_INTERRUPT_SOURCE_REGISTER_0 [PACKET_XFER_COMPLETE] would be set. And if the packet was configured to have STOP bit, I2C_I2C_INTERRUPT_SOURCE_REGISTER_0[ALL_PACKETS_XFER_COMPLETE] would also be set.

3. Clear them by writing 1 to I2C_INTERRUPT_STATUS_REGISTER_0[PACKET_XFER_COMPLETE] and I2C_INTERRUPT_STATUS_REGISTER_0[ALL_PACKETS_XFER_COMPLETE] bits accordingly. This clears the interrupt.

Transfers Completion

1. Check DMA status to know if all the packets are transferred to the I^2C controller from memory.
2. Use PACKET_XFER_COMPLETE (if packets have RepeatedStart or ContinueXfer settings) or ALL_PACKETS_XFER_COMPLETE (if packets have only STOP) alerts to track how many packets are successfully transferred to the I^2C bus. When all packets are transferred successfully to the bus, it matches total alerts that I^2C sends to the system.
3. Check DMA status if it read all the data bytes from the I^2C controller and saved it to the memory.

Unsuccessful Transfers

A packet transfer could be unsuccessful due to various reasons. The I^2C master logs this information in I2C_INTERRUPT_STATUS_REGISTER_0 register. The transfer could be unsuccessful due to the I^2C bus related ARB_LOST/NACK events or TX_PACKET_FIFO Overflow errors or I2C_I2C_RX_FIFO_0 Underflow errors.

1. I2C_INTERRUPT_STATUS_REGISTER_0[ARB_LOST] is set to 1 if the I^2C master sees the bus is already occupied by some other device when a transfer is initiated or if it loses the bus during the arbitration in multi-master case or if a slave device pulls the SDA line low continuously for some unknown reason. I2C_INTERRUPT_STATUS_REGISTER_0[ARB_LOST] is set to 1 if the I^2C master sees the bus is already occupied by some other device when a transfer is initiated or if it loses the bus during the arbitration in multi-master case or if a slave device pulls the SDA line low continuously for some unknown reason.
2. I2C_INTERRUPT_STATUS_REGISTER_0[NOACK] is set to 1 if slave device responds with NACK for address or data byte transfer of a transaction.
3. I2C_INTERRUPT_STATUS_REGISTER_0[TFIFO_OVF] is set to 1 if DMA or CPU writes to I2C_I2C_TX_PACKET_FIFO_0 even it is full.
4. I2C_INTERRUPT_STATUS_REGISTER_0[RFIFO_UNF] is set to 1 if DMA or CPU reads I2C_I2C_RX_FIFO_0 even it is empty.
5. When any of these error events happens, the I^2C controller stops further packet transfers and flushes the data from the FIFOs.
6. If interrupt enables are set, corresponding bits would be set in I2C_I2C_INTERRUPT_SOURCE_REGISTER_0 register and signals an interrupt to the system.

Interrupt Handling for Unsuccessful Transfers

1. Read I2C_I2C_INTERRUPT_SOURCE_REGISTER_0 register and check which bit fields are set to 1 (ARB_LOST/NOACK/TFIFO_OVF/ RFIFO_UNF)

2. In all these cases, the I^2C controller reset is required. Follow the below sequence before going for further transfers
3. Assert I^2C controller reset
4. Disable DMA
5. Release I^2C controller reset
6. Redo the I^2C configuration, including DMA, and go for the transfers.

9.6.3.5 Programming Guidelines for Slave in Byte Mode

9.6.3.5.1 Programming Model for Slave-Receive in 7-bit Addressing Mode (Transfer Direction not Changed), Interrupt Method

Initial Configuration

Before accessing the I^2C controller, set up the following:

- I^2C clocks programming
- Pinmux programming
- I^2C controller reset release

I2C Controller Registers Initialization

- Write I2C_I2C_SLV_RESET_CNTRL[SOFT_RESET] = CLEARED
- Write I2C_FIFO_CONTROL = 0x0
- Write I2C_I2C_INTERRUPT_SET_REGISTER = 0x0
- Write I2C_I2C_CLKEN_OVERRIDE = 0x0
- Write I2C_I2C_DEBUG_CONTROL = 0x0
- Write I2C_I2C_TLOW_SEXT_0 = 0x0
- Write I2C_I2C_SL_INT_SET_0 = 0x0
- Write I2C_I2C_SL_DELAY_COUNT_0 = 0x1e

Select Slave Address Register and Configure it in 7-bit Addressing Mode

- Write I2C_I2C_SL_ADDR1_0[SL_ADDR1] = 0
- Write 7-bit slave address in I2C_I2C_SL_ADDR1_0[SL_ADDR0] = {1'b0,SlaveAddress[6:0]}
- To use SL_ADDR0 bit field for Slave address, write I2C_I2C_SL_ADDR2_0[SELECT_SLAVE] = 0
- Write I2C_I2C_SL_ADDR2_0[SL1_ADDR_HI] = 0
- Write I2C_I2C_SL_ADDR2_0[SL1_VLD] = SEVEN_BIT_ADDR_MODE
- Write I2C_I2C_SL_ADDR2_0[SL_ADDR_HI] = 0
- Write I2C_I2C_SL_ADDR2_0[VLD] = SEVEN_BIT_ADDR_MODE

Configure Interrupt Mask Registers to Enable the Interrupts

- Write I2C_INTERRUPT_MASK_REGISTER_0 = 0x0
- Write I2C_I2C_SL_INT_MASK_0[HW_MSTR_INT] = DISABLE
- Write I2C_I2C_SL_INT_MASK_0[REPROG_SL] = DISABLE
- Write I2C_I2C_SL_INT_MASK_0[RST_SL] = DISABLE
- Write I2C_I2C_SL_INT_MASK_0[END_TRANS] = ENABLE
- Write I2C_I2C_SL_INT_MASK_0[SL_IRQ] = ENABLE
- Write I2C_I2C_SL_INT_MASK_0[RCVD] = ENABLE
- Write I2C_I2C_SL_INT_MASK_0[RCVD] = ENABLE
- Write I2C_I2C_SL_INT_MASK_0[ZA] = DISABLE

Configure SL_CNFG Register to Enable the Slave Controller

- Write I2C_I2C_SL_CNFG_0[ENABLE_SL] = 1
- Write I2C_I2C_SL_CNFG_0[PKT_MODE_EN] = 0
- Write I2C_I2C_SL_CNFG_0[NEWSL] = 1
- Write I2C_I2C_SL_CNFG_0[FIFO_XFER_EN] = 0
- Write I2C_I2C_SL_CNFG_0[SLV_XFER_ERR_CLK_STRETCH_EN] = 0
- Write I2C_I2C_SL_CNFG_0[BUFFER_SIZE] = 0
- Write I2C_I2C_SL_CNFG_0[ACK_LAST_BYTE_VALID] = 0
- Write I2C_I2C_SL_CNFG_0[RESP] = 0
- Write I2C_I2C_SL_CNFG_0[ACK_WITHHOLD_EN] = 0
- Write I2C_I2C_SL_CNFG_0[ACK_LAST_BYTE] = 0
- Write I2C_I2C_SL_CNFG_0[NACK] = DISABLE

Activate the Configuration with I2C_CONFIG_LOAD Register Programming

- I2C_I2C_CONFIG_LOAD_0[TIMEOUT_CONFIG_LOAD] = ENABLE
- I2C_I2C_CONFIG_LOAD_0[SLV_CONFIG_LOAD] = ENABLE
- I2C_I2C_CONFIG_LOAD_0[MSTR_CONFIG_LOAD] = DISABLE

Wait for the Hardware to Clear the I2C_CONFIG_LOAD Register Bit Fields

1. Wait until I2C_I2C_CONFIG_LOAD_0[TIMEOUT_CONFIG_LOAD] becomes zero
2. Wait until I2C_I2C_CONFIG_LOAD_0[SLV_CONFIG_LOAD] becomes zero

With this, the Slave controller is ready for the data transfers. Start a Write transaction from the I^2C Master device.

Slave Response to Address Phase of the Write Transaction

When Slave receives address byte from the bus and if the address matches its programmed address, it does the following.

1. Sets I2C_I2C_SL_STATUS_0[RNW] bit field with received address byte[0] bit which indicates transfer direction. 0 = WRITE, 1=READ
2. Saves the received address byte in I2C_I2C_SL_RCVD_0 register
3. Sets I2C_I2C_SL_STATUS_0[RCVD] and I2C_I2C_SL_STATUS_0[SL_IRQ] bits indicating that a new transaction is received.
4. Generates an interrupt to the system if interrupt enables bits (I2C_I2C_SL_INT_MASK_0[RCVD], I2C_I2C_SL_INT_MASK_0[SL_IRQ]) are set
5. Holds the I^2C bus by doing clock stretching (holds I^2C SCL line low) until the interrupt is cleared

New Transaction Interrupt Handling

1. Read I2C_I2C_SL_INT_SOURCE_0 register and check which bit fields are set to 1.
2. Each bit field signifies an event occurrence.
3. If a bit field is set to 1, it means, the current interrupt is because of the relevant event to this bit.
4. In the new transaction received case, I2C_I2C_SL_INT_SOURCE_0[RCVD] and I2C_I2C_SL_INT_SOURCE_0[SL_IRQ] bits would be set to 1.
5. Check I2C_I2C_SL_STATUS_0[RNW] bit status. For write transactions, I2C_I2C_SL_STATUS_0[RNW] = WRITE would be set.
6. Clear I2C_I2C_SL_STATUS_0[RCVD] and I2C_I2C_SL_STATUS_0[SL_IRQ] bits by writing 1 to them.
7. This will clear the interrupt and release the SCL line allowing the Master to go to next phase of the transaction.

Slave Response to Data Phase of the Write Transaction

When the Master sends a data byte, the Slave controller does the following:

1. Saves the received byte in I2C_I2C_SL_RCVD_0 register.
2. Sets I2C_I2C_SL_STATUS_0[SL_IRQ] bit indicating a byte received.
3. Generates an interrupt to the system if interrupt enable bit I2C_I2C_SL_INT_MASK_0[SL_IRQ] is set.
4. Slave will hold the I^2C bus by doing clock stretching (holds I^2C SCL line low) until the interrupt is cleared.

Interrupts Handling When Received a Data Byte

1. Read I2C_I2C_SL_INT_SOURCE_0 register and check which bit fields are set to 1.
2. In the data_byte received case, I2C_I2C_SL_INT_SOURCE_0[SL_IRQ] bit would be set to 1.

3. Read the data byte from I2C_I2C_SL_RCVD_0 register
4. To continue with the transfer, keep I2C_I2C_SL_CNFG_0[NACK] = DISABLE setting. Otherwise, to signal the Master to stop the transfer, configure I2C_I2C_SL_CNFG_0[NACK] = ENABLE
5. Clear I2C_I2C_SL_STATUS_0[SL_IRQ] by writing 1 to it. This clears the interrupt and releases the SCL line, allowing the Master to go ahead with next phase of the transaction

Slave Response to STOP Condition of the Write Transaction

If the Master sends a STOP condition next, the Slave does the following:

1. Sets I2C_I2C_SL_STATUS_0[END_TRANS] and I2C_I2C_SL_STATUS_0[SL_IRQ] bits indicating STOP condition is received
2. Generates an interrupt to the system if Interrupt bits (I2C_I2C_SL_INT_MASK_0[END_TRANS], I2C_I2C_SL_INT_MASK_0[SL_IRQ]) are enabled.

Interrupts Handling When Stop Condition is Received

1. Read I2C_I2C_SL_INT_SOURCE_0 register and check which bit fields are set to 1.
2. In the STOP condition case, I2C_I2C_SL_INT_SOURCE_0[END_TRANS] and I2C_I2C_SL_INT_SOURCE_0[SL_IRQ] bits would be set to 1
3. Clear I2C_I2C_SL_STATUS_0[END_TRANS] and I2C_I2C_SL_STATUS_0[SL_IRQ] bits by writing 1 to them. This will clear the interrupt.

Notes:

1. If the Master terminates the transaction while the Slave controller is ready to accept the data, I2C_INTERRUPT_STATUS_REGISTER_0[SLV_PKT_XFER_ERR] bit is set at the end of the transaction when STOP condition is received. This bit can be cleared by writing 1 to it or it can be ignored.
2. If the Slave controller sends NACK to any of the bytes during the write transfer, Master will terminate the transaction after seeing NACK bit. In this case, No status bits (END_TRANS, SL_IRQ or SLV_PKT_XFER_ERR) would be set when STOP condition is received. No interrupts either.

9.6.3.5.2 Programming Model for Slave-Transmit in 7-bit Addressing Mode (Transfer Direction not Changed), Interrupt Method

Initial Configuration

Before accessing the I^2C controller, set up the following:

- I^2C clocks programming
- Pinmux programming
- I^2C controller reset release

I^{2}C Controller Registers Initialization

- Write I2C_I2C_SLV_RESET_CNTRL[SOFT_RESET] = CLEARED
- Write I2C_SLV_FIFO_CONTROL = 0x0
- Write I2C_I2C_INTERRUPT_SET_REGISTER = 0x0
- Write I2C_I2C_CLKEN_OVERRIDE = 0x0
- Write I2C_I2C_DEBUG_CONTROL = 0x0
- Write I2C_I2C_TLOW_SEXT_0 = 0x0
- Write I2C_I2C_SL_INT_SET_0 = 0x0
- Write I2C_I2C_SL_DELAY_COUNT_0 = 0x1e

Select Slave Address Register and Configure it in 7-bit Addressing Mode

- Write I2C_I2C_SL_ADDR1_0[SL_ADDR1] = 0
- Write 7-bit slave address in I2C_I2C_SL_ADDR1_0[SL_ADDR0] = {1'b0, SlaveAddress[6:0]}
- To use SL_ADDR0 bit field for Slave address, write I2C_I2C_SL_ADDR2_0[SELECT_SLAVE] = 0
- Write I2C_I2C_SL_ADDR2_0[SL1_ADDR_HI] = 0
- Write I2C_I2C_SL_ADDR2_0[SL1_VLD] = SEVEN_BIT_ADDR_MODE
- Write I2C_I2C_SL_ADDR2_0[SL_ADDR_HI] = 0
- Write I2C_I2C_SL_ADDR2_0[VLD] = SEVEN_BIT_ADDR_MODE

Configure Interrupt Mask Registers to Enable the Interrupts

- Write I2C_INTERRUPT_MASK_REGISTER_0 = 0x0
- Write I2C_I2C_SL_INT_MASK_0[HW_MSTR_INT] = DISABLE
- Write I2C_I2C_SL_INT_MASK_0[REPROG_SL] = DISABLE
- Write I2C_I2C_SL_INT_MASK_0[RST_SL] = DISABLE
- Write I2C_I2C_SL_INT_MASK_0[END_TRANS] = ENABLE
- Write I2C_I2C_SL_INT_MASK_0[SL_IRQ] = ENABLE
- Write I2C_I2C_SL_INT_MASK_0[RCVD] = ENABLE
- Write I2C_I2C_SL_INT_MASK_0[ZA] = DISABLE

Configure SL_CNFG Register to Enable the Slave Controller

- Write I2C_I2C_SL_CNFG_0[ENABLE_SL] = 1
- Write I2C_I2C_SL_CNFG_0[PKT_MODE_EN] = 0
- Write I2C_I2C_SL_CNFG_0[NEWSL] = 1
- Write I2C_I2C_SL_CNFG_0[FIFO_XFER_EN] = 0
- Write I2C_I2C_SL_CNFG_0[SLV_XFER_ERR_CLK_STRETCH_EN] = 0
- Write I2C_I2C_SL_CNFG_0[BUFFER_SIZE] = 0
- Write I2C_I2C_SL_CNFG_0[ACK_LAST_BYTE_VALID] = 0

- Write I2C_I2C_SL_CNFG_0[RESP] = 0
- Write I2C_I2C_SL_CNFG_0[ACK_WITHHOLD_EN] = 0
- Write I2C_I2C_SL_CNFG_0[ACK_LAST_BYTE] = 0
- Write I2C_I2C_SL_CNFG_0[NACK] = DISABLE

Activate the Configuration with I2C_CONFIG_LOAD Register Programming

- I2C_I2C_CONFIG_LOAD_0[TIMEOUT_CONFIG_LOAD] = ENABLE
- I2C_I2C_CONFIG_LOAD_0[SLV_CONFIG_LOAD] = ENABLE
- I2C_I2C_CONFIG_LOAD_0[MSTR_CONFIG_LOAD] = DISABLE

Wait for the Hardware to Clear the I2C_CONFIG_LOAD Register Bit Fields

1. Wait until I2C_I2C_CONFIG_LOAD_0[TIMEOUT_CONFIG_LOAD] becomes zero
2. Wait until I2C_I2C_CONFIG_LOAD_0[SLV_CONFIG_LOAD] becomes zero

With this, the Slave controller is ready for the data transfers. Start a Read transaction from the I^2C Master device.

Slave Response to Address Phase of Read Transaction

When the Slave receives an address byte from the bus and if the address matches its programmed address, it does the following:

1. Sets I2C_I2C_SL_STATUS_0[RNW] bit field with received address byte[0] bit which indicates transfer direction. 0 = WRITE, 1=READ
2. Saves the received address byte in I2C_I2C_SL_RCVD_0 register.
3. Sets I2C_I2C_SL_STATUS_0[RCVD] and I2C_I2C_SL_STATUS_0[SL_IRQ] bits indicating that a new transaction is received.
4. Generates an interrupt to the system if Interrupt enables bits (I2C_I2C_SL_INT_MASK_0[RCVD], I2C_I2C_SL_INT_MASK_0[SL_IRQ]) are set.
5. Holds the I^2C bus by doing clock stretching (holds I^2C SCL line low) until the interrupt is cleared and a data byte is available in the I2C_I2C_SL_RCVD_0 register.

New Transaction Interrupt Handling

1. Read I2C_I2C_SL_INT_SOURCE_0 register and check which bit fields are set to 1.
2. In the new transaction received case, I2C_I2C_SL_INT_SOURCE_0[RCVD] and I2C_I2C_SL_INT_SOURCE_0[SL_IRQ] bits are set to 1.
3. Check I2C_I2C_SL_STATUS_0[RNW] bit status. For read transactions, I2C_I2C_SL_STATUS_0[RNW] = READ is set.
4. Clear I2C_I2C_SL_STATUS_0[RCVD] and I2C_I2C_SL_STATUS_0[SL_IRQ] bits by writing 1 to them. This will clear the interrupt.
5. Write Data byte to the I2C_I2C_SL_RCVD_0 register. This will release the SCL line. Master can read this byte now.

Slave Response to Data Phase of Read Transaction

If Master ACKs previous byte transfer, it means the Master wants to continue the read operation further. The Slave controller does the following in this case:

1. Sets I2C_I2C_SL_STATUS_0[SL_IRQ] bit indicating a request to transmit a data byte.
2. Generates an interrupt to the system if interrupt enable bit I2C_I2C_SL_INT_MASK_0[SL_IRQ] is set.
3. Slave will hold the I^2C bus by doing clock stretching (holds I^2C SCL line low) until a data byte is available in I2C_I2C_SL_RCVD_0 register to transmit to the bus.

Interrupts Handling in Data Byte Transmit Case

1. Read I2C_I2C_SL_INT_SOURCE_0 register and check which bit fields are set to 1.
2. In the data transmit case, only I2C_I2C_SL_INT_SOURCE_0[SL_IRQ] bit is set to 1.
3. Clear I2C_I2C_SL_STATUS_0[SL_IRQ] bit by writing 1 to it. This will clear the interrupt.
4. Write a Data byte to the I2C_I2C_SL_RCVD_0 register. This will release the SCL line and allow the Master to continue the transfer.

Slave Response to End of the Read Transaction

If Master NACKs previous byte transfer, it means the Master wants to end the transaction. The Slave controller does the following then:

1. Sets I2C_I2C_SL_STATUS_0[END_TRANS] and I2C_I2C_SL_STATUS_0[SL_IRQ] bits indicating the end of current transaction.
2. Generates an interrupt to the system if interrupt bits (I2C_I2C_SL_INT_MASK_0[END_TRANS], I2C_I2C_SL_INT_MASK_0[SL_IRQ]) are enabled.

Interrupts Handling When Transaction END is Received During Read

1. Read the I2C_I2C_SL_INT_SOURCE_0 register and check which bit fields are set to 1.
2. In the transaction END case, I2C_I2C_SL_INT_SOURCE_0[END_TRANS] and I2C_I2C_SL_INT_SOURCE_0[SL_IRQ] bits are set to 1
3. Clear I2C_I2C_SL_STATUS_0[END_TRANS] and I2C_I2C_SL_STATUS_0[SL_IRQ] bits by writing 1 to them. This will clear the interrupt.

9.6.3.5.3 Programming Model for Slave-Receive-Transmit (Transfer Direction Changed from Receive-to-Transmit with RepeatedStart) in 7-bit Addressing Mode, Interrupt Method

Initial Configuration

Before accessing the I^2C controller, set up the following:

- I^2C clocks programming

- Pinmux programming
- I^2C controller reset release

I^2C Controller Registers Initialization

- Write I2C_I2C_SLV_RESET_CNTRL[SOFT_RESET] = CLEARED
- Write I2C_SLV_FIFO_CONTROL = 0x0
- Write I2C_I2C_INTERRUPT_SET_REGISTER = 0x0
- Write I2C_I2C_CLKEN_OVERRIDE = 0x0
- Write I2C_I2C_DEBUG_CONTROL = 0x0
- Write I2C_I2C_TLOW_SEXT_0 = 0x0
- Write I2C_I2C_SL_INT_SET_0 = 0x0
- Write I2C_I2C_SL_DELAY_COUNT_0 = 0x1e

Select Slave Address Register and Configure it in 7-bit Addressing Mode

- Write I2C_I2C_SL_ADDR1_0[SL_ADDR1] = 0
- Write 7-bit slave address in I2C_I2C_SL_ADDR1_0[SL_ADDR0] = {1'b0, SlaveAddress[6:0]}
- To use SL_ADDR0 bit field for Slave address, write I2C_I2C_SL_ADDR2_0[SELECT_SLAVE] = 0
- Write I2C_I2C_SL_ADDR2_0[SL1_ADDR_HI] = 0
- Write I2C_I2C_SL_ADDR2_0[SL1_VLD] = SEVEN_BIT_ADDR_MODE
- Write I2C_I2C_SL_ADDR2_0[SL_ADDR_HI] = 0
- Write I2C_I2C_SL_ADDR2_0[VLD] = SEVEN_BIT_ADDR_MODE

Configure Interrupt Mask Registers to Enable the Interrupts

- Write I2C_INTERRUPT_MASK_REGISTER_0[SLV_WR2RD_INT_EN] = ENABLE
- Write I2C_I2C_SL_INT_MASK_0[HW_MSTR_INT] = DISABLE
- Write I2C_I2C_SL_INT_MASK_0[REPROG_SL] = DISABLE
- Write I2C_I2C_SL_INT_MASK_0[RST_SL] = DISABLE
- Write I2C_I2C_SL_INT_MASK_0[END_TRANS] = ENABLE
- Write I2C_I2C_SL_INT_MASK_0[SL_IRQ] = ENABLE
- Write I2C_I2C_SL_INT_MASK_0[RCVD] = ENABLE
- Write I2C_I2C_SL_INT_MASK_0[ZA] = DISABLE

Configure SL_CNFG Register to Enable the Slave Controller

- Write I2C_I2C_SL_CNFG_0[ENABLE_SL] = 1
- Write I2C_I2C_SL_CNFG_0[PKT_MODE_EN] = 0
- Write I2C_I2C_SL_CNFG_0[NEWSL] = 1
- Write I2C_I2C_SL_CNFG_0[FIFO_XFER_EN] = 0

- Write I2C_I2C_SL_CNFG_0[SLV_XFER_ERR_CLK_STRETCH_EN] = 0
- Write I2C_I2C_SL_CNFG_0[BUFFER_SIZE] = 0
- Write I2C_I2C_SL_CNFG_0[ACK_LAST_BYTE_VALID] = 0
- Write I2C_I2C_SL_CNFG_0[RESP] = 0
- Write I2C_I2C_SL_CNFG_0[ACK_WITHHOLD_EN] = 0
- Write I2C_I2C_SL_CNFG_0[ACK_LAST_BYTE] = 0
- Write I2C_I2C_SL_CNFG_0[NACK] = DISABLE

Activate the Configuration with I2C_CONFIG_LOAD Register Programming

- I2C_I2C_CONFIG_LOAD_0[TIMEOUT_CONFIG_LOAD] = ENABLE
- I2C_I2C_CONFIG_LOAD_0[SLV_CONFIG_LOAD] = ENABLE
- I2C_I2C_CONFIG_LOAD_0[MSTR_CONFIG_LOAD] = DISABLE

Wait for the Hardware to Clear the I2C_CONFIG_LOAD Register Bit Fields

1. Wait until I2C_I2C_CONFIG_LOAD_0[TIMEOUT_CONFIG_LOAD] becomes zero
2. Wait until I2C_I2C_CONFIG_LOAD_0[SLV_CONFIG_LOAD] becomes zero

With this, Slave controller is ready for the data transfers. Start a transaction from the I^2C Master device.

Slave Response to Address Phase of Write Transaction

When the Slave receives address byte from the bus and if the address matches its programmed address, it does the following:

1. Sets I2C_I2C_SL_STATUS_0[RNW] bit field with received address byte[0] bit which indicates transfer direction. 0 = WRITE, 1=READ
2. Saves the received address byte in the I2C_I2C_SL_RCVD_0 register.
3. Sets I2C_I2C_SL_STATUS_0[RCVD] and I2C_I2C_SL_STATUS_0[SL_IRQ] bits indicating that a new transaction is received.
4. Generates an interrupt to the system if interrupt enables bits (I2C_I2C_SL_INT_MASK_0[RCVD], I2C_I2C_SL_INT_MASK_0[SL_IRQ]) are set.
5. Holds the I^2C bus by doing clock stretching (holds I^2C SCL line low) until the interrupt is cleared.

New Transaction Interrupt Handling

1. Read I2C_I2C_SL_INT_SOURCE_0 register and check which bit fields are set to 1.
2. Each bit field signifies an event occurrence.
3. If a bit field is set to 1, it means, the current interrupt is because of the relevant event to this bit.

4. In the new transaction received case, I2C_I2C_SL_INT_SOURCE_0[RCVD] and I2C_I2C_SL_INT_SOURCE_0[SL_IRQ] bits would be set to 1.
5. Check I2C_I2C_SL_STATUS_0[RNW] bit status. For write transactions, I2C_I2C_SL_STATUS_0[RNW] = WRITE would be set.
6. Clear I2C_I2C_SL_STATUS_0[RCVD] and I2C_I2C_SL_STATUS_0[SL_IRQ] bits by writing 1 to them.
7. This clears the interrupt and releases the SCL line, allowing the Master to go to next phase of the transaction.

Slave Response to Data Phase of the Write Transaction

When the Master sends a data byte, the Slave controller does the following:

1. Saves the received byte in the I2C_I2C_SL_RCVD_0 register.
2. Sets I2C_I2C_SL_STATUS_0[SL_IRQ] bit indicating a byte received.
3. Generates an interrupt to the system if interrupt enable bit I2C_I2C_SL_INT_MASK_0[SL_IRQ] is set.
4. The Slave will hold the I^2C bus by doing clock stretching (holds I^2C SCL line low) until the interrupt is cleared.

Interrupts Handling When Received a Data Byte

1. Read I2C_I2C_SL_INT_SOURCE_0 register and check which bit fields are set to 1.
2. In the data_byte received case, I2C_I2C_SL_INT_SOURCE_0[SL_IRQ] bit is set to 1.
3. Read the data byte from the I2C_I2C_SL_RCVD_0 register.
4. To continue with the transfer, keep I2C_I2C_SL_CNFG_0[NACK] = DISABLE setting. Otherwise, to signal the Master to stop the transfer, configure I2C_I2C_SL_CNFG_0[NACK] = ENABLE
5. Clear I2C_I2C_SL_STATUS_0[SL_IRQ] by writing 1 to it. This clears the interrupt and releases the SCL line, allowing the Master to go ahead with next phase of the transaction.

Slave Response to Write-to-Read Transfer Direction Change with RepeatedStart Condition

When the Master sends a RepeatedStart condition and changes the transfer direction from write to read, the Slave controller does the following:

1. Sets I2C_I2C_SL_STATUS_0[RNW] bit field with received address byte[0] bit which indicates the new transfer direction. 0 = WRITE, 1=READ
2. Saves the received address byte in the I2C_I2C_SL_RCVD_0 register.
3. Sets I2C_I2C_SL_STATUS_0[RCVD] and I2C_I2C_SL_STATUS_0[SL_IRQ] bits indicating that a new transaction is received.
4. Generates an interrupt to the system if interrupt enables bits (I2C_I2C_SL_INT_MASK_0[RCVD], I2C_I2C_SL_INT_MASK_0[SL_IRQ]) are set.

5. Holds the I^2C bus by doing clock stretching (holds I^2C SCL line low) until the interrupt is cleared and a data byte is available in I2C_I2C_SL_RCVD_0 register to transmit to the Master.

New Transaction Interrupt Handling

1. Read I2C_I2C_SL_INT_SOURCE_0 register and check which bit fields are set to 1.
2. In the new transaction received case, I2C_I2C_SL_INT_SOURCE_0[RCVD] and I2C_I2C_SL_INT_SOURCE_0[SL_IRQ] bits are set to 1.
3. Check I2C_I2C_SL_STATUS_0[RNW] bit status. For read transactions, it would be I2C_I2C_SL_STATUS_0[RNW] = READ.
4. Clear I2C_I2C_SL_STATUS_0[RCVD] and I2C_I2C_SL_STATUS_0[SL_IRQ] bits by writing 1 to them. This will clear the interrupt.
5. This will set I2C_INTERRUPT_STATUS_REGISTER_0[SLV_WR2RD] bit to 1 now, and an interrupt is generated. Clear the interrupt by writing 1 to I2C_INTERRUPT_STATUS_REGISTER_0[SLV_WR2RD] bit field.
6. I2C_INTERRUPT_STATUS_REGISTER_0[SLV_TX_BUFFER_REQ] bit is set now. This can be ignored.
7. Write Data byte to I2C_I2C_SL_RCVD_0 register. This will release the SCL line. The Master can read the byte now.

Slave Response to Data Phase of Read Transaction

If Master ACKs previous byte of the read transfer, it means the Master wants to continue the read operation further. The Slave controller does the following in this case:

1. Sets I2C_I2C_SL_STATUS_0[SL_IRQ] bit indicating a request to transmit a data byte.
2. Generates an interrupt to the system if interrupt enable bit I2C_I2C_SL_INT_MASK_0[SL_IRQ] is set.
3. The Slave will hold the I^2C bus by doing clock stretching (holds I^2C SCL line low) until a data byte is available in the I2C_I2C_SL_RCVD_0 register to transmit to the bus.

Interrupts Handling in Data Byte Transmit Case

1. Read I2C_I2C_SL_INT_SOURCE_0 register and check which bit fields are set to 1.
2. In the data transmit case, only I2C_I2C_SL_INT_SOURCE_0[SL_IRQ] bit is set to 1.
3. Clear I2C_I2C_SL_STATUS_0[SL_IRQ] bit by writing 1 to it. This will clear the interrupt.
4. Write a Data byte to the I2C_I2C_SL_RCVD_0 register. This will release the SCL line and allow the Master to continue the transfer.

Slave Response to End of the Read Transaction

If Master NACKs previous byte transfer, it means the Master wants to end the transaction. The Slave controller does the following then:

1. Sets I2C_I2C_SL_STATUS_0[END_TRANS] and I2C_I2C_SL_STATUS_0[SL_IRQ] bits indicating the end of current transaction.

2. Generates an interrupt to the system if interrupt bits (I2C_I2C_SL_INT_MASK_0[END_TRANS], I2C_I2C_SL_INT_MASK_0[SL_IRQ]) are enabled.

Interrupts Handling When Transaction END is Received During Read

1. Read the I2C_I2C_SL_INT_SOURCE_0 register and check which bit fields are set to 1.
2. In the transaction END case, I2C_I2C_SL_INT_SOURCE_0[END_TRANS] and I2C_I2C_SL_INT_SOURCE_0[SL_IRQ] bits are set to 1.
3. Clear I2C_I2C_SL_STATUS_0[END_TRANS] and I2C_I2C_SL_STATUS_0[SL_IRQ] bits by writing 1 to them. This will clear the interrupt.

9.6.3.5.4 Programming Model for Slave-Transmit-Receive (transfer direction changed from Transmit-to-Receive with RepeatedStart) in 7-bit Addressing Mode, Interrupt Method

Initial Configuration

Before accessing the I^2C controller, set up the following:

- I^2C clocks programming
- Pinmux programming
- I^2C controller reset release

I^2C Controller Registers Initialization

- Write I2C_I2C_SLV_RESET_CNTRL[SOFT_RESET] = CLEARED]
- Write I2C_SLV_FIFO_CONTROL = 0x0
- Write I2C_I2C_INTERRUPT_SET_REGISTER = 0x0
- Write I2C_I2C_CLKEN_OVERRIDE = 0x0
- Write I2C_I2C_DEBUG_CONTROL = 0x0
- Write I2C_I2C_TLOW_SEXT_0 = 0x0
- Write I2C_I2C_SL_INT_SET_0 = 0x0
- Write I2C_I2C_SL_DELAY_COUNT_0 = 0x1e

Select Slave Address Register and Configure it in 7-bit Addressing Mode

- Write I2C_I2C_SL_ADDR1_0[SL_ADDR1] = 0
- Write 7-bit slave address in I2C_I2C_SL_ADDR1_0[SL_ADDR0] = {1'b0, SlaveAddress[6:0]}
- To use SL_ADDR0 bit field for Slave address, write I2C_I2C_SL_ADDR2_0[SELECT_SLAVE] = 0
- Write I2C_I2C_SL_ADDR2_0[SL1_ADDR_HI] = 0
- Write I2C_I2C_SL_ADDR2_0[SL1_VLD] = SEVEN_BIT_ADDR_MODE
- Write I2C_I2C_SL_ADDR2_0[SL_ADDR_HI] = 0
- Write I2C_I2C_SL_ADDR2_0[VLD] = SEVEN_BIT_ADDR_MODE

Configure Interrupt Mask Registers to Enable the Interrupts

- Write I2C_INTERRUPT_MASK_REGISTER_0 = 0x0
- Write I2C_I2C_SL_INT_MASK_0[HW_MSTR_INT] = DISABLE
- Write I2C_I2C_SL_INT_MASK_0[REPROG_SL] = DISABLE
- Write I2C_I2C_SL_INT_MASK_0[RST_SL] = DISABLE
- Write I2C_I2C_SL_INT_MASK_0[END_TRANS] = ENABLE
- Write I2C_I2C_SL_INT_MASK_0[SL_IRQ] = ENABLE
- Write I2C_I2C_SL_INT_MASK_0[RCVD] = ENABLE
- Write I2C_I2C_SL_INT_MASK_0[ZA] = DISABLE

Configure SL_CNFG Register to Enable the Slave Controller

- Write I2C_I2C_SL_CNFG_0[ENABLE_SL] = 1
- Write I2C_I2C_SL_CNFG_0[PKT_MODE_EN] = 0
- Write I2C_I2C_SL_CNFG_0[NEWSL] = 1
- Write I2C_I2C_SL_CNFG_0[FIFO_XFER_EN] = 0
- Write I2C_I2C_SL_CNFG_0[SLV_XFER_ERR_CLK_STRETCH_EN] = 0
- Write I2C_I2C_SL_CNFG_0[BUFFER_SIZE] = 0
- Write I2C_I2C_SL_CNFG_0[ACK_LAST_BYTE_VALID] = 0
- Write I2C_I2C_SL_CNFG_0[RESP] = 0
- Write I2C_I2C_SL_CNFG_0[ACK_WITHHOLD_EN] = 0
- Write I2C_I2C_SL_CNFG_0[ACK_LAST_BYTE] = 0
- Write I2C_I2C_SL_CNFG_0[NACK] = DISABLE

Activate the Configuration with I2C_CONFIG_LOAD Register Programming

- I2C_I2C_CONFIG_LOAD_0[TIMEOUT_CONFIG_LOAD] = ENABLE
- I2C_I2C_CONFIG_LOAD_0[SLV_CONFIG_LOAD] = ENABLE
- I2C_I2C_CONFIG_LOAD_0[MSTR_CONFIG_LOAD] = DISABLE

Wait for the Hardware to Clear the I2C_CONFIG_LOAD Register Bit Fields

1. Wait until I2C_I2C_CONFIG_LOAD_0[TIMEOUT_CONFIG_LOAD] becomes zero
2. Wait until I2C_I2C_CONFIG_LOAD_0[SLV_CONFIG_LOAD] becomes zero

With this, the Slave controller is ready for the data transfers. Start a transaction from the I^2C Master device.

Slave Response to Address Phase of Read Transaction

When the Slave receives an address byte from the bus and if the address matches its programmed address, it does the following:

1. Sets I2C_I2C_SL_STATUS_0[RNW] bit field with received address byte[0] bit which indicates transfer direction. 0 = WRITE, 1 = READ
2. Saves the received address byte in the I2C_I2C_SL_RCVD_0 register.
3. Sets I2C_I2C_SL_STATUS_0[RCVD] and I2C_I2C_SL_STATUS_0[SL_IRQ] bits indicating that a new transaction is received.
4. Generates an interrupt to the system if Interrupt enable bits (I2C_I2C_SL_INT_MASK_0[RCVD], I2C_I2C_SL_INT_MASK_0[SL_IRQ]) are set.
5. Holds the I^2C bus by doing clock stretching (holds I^2C SCL line low) until the interrupt is cleared and a data byte is available in the I2C_I2C_SL_RCVD_0 register.

New Transaction Interrupt Handling

1. Read the I2C_I2C_SL_INT_SOURCE_0 register and check which bit fields are set to 1.
2. In the new transaction received case, I2C_I2C_SL_INT_SOURCE_0[RCVD] and I2C_I2C_SL_INT_SOURCE_0[SL_IRQ] bits are set to 1.
3. Check I2C_I2C_SL_STATUS_0[RNW] bit status. For read transactions, I2C_I2C_SL_STATUS_0[RNW] = READ is set.
4. Clear I2C_I2C_SL_STATUS_0[RCVD] and I2C_I2C_SL_STATUS_0[SL_IRQ] bits by writing 1 to them. This will clear the interrupt.
5. Write Data byte to the I2C_I2C_SL_RCVD_0 register. This will release the SCL line. The Master can read this byte now.

Slave Response to Data Phase of Read Transaction

If Master ACKs previous byte transfer, it means, Master wants to continue the read operation further. Slave controller does the following in this case.

1. Sets the I2C_I2C_SL_STATUS_0[SL_IRQ] bit indicating a request to transmit a data byte.
2. Generates an interrupt to the system if interrupt enable bit I2C_I2C_SL_INT_MASK_0[SL_IRQ] is set.
3. The Slave will hold the I^2C bus by doing clock stretching (holds I^2C SCL line low) until a data byte is available in the I2C_I2C_SL_RCVD_0 register to transmit to the bus.

Interrupts Handling in Data Byte Transmit Case

1. Read the I2C_I2C_SL_INT_SOURCE_0 register and check which bit fields are set to 1.
2. In the data transmit case, only I2C_I2C_SL_INT_SOURCE_0[SL_IRQ] bit is set to 1.
3. Clear I2C_I2C_SL_STATUS_0[SL_IRQ] bit by writing 1 to it. This will clear the interrupt.
4. Write a Data byte to the I2C_I2C_SL_RCVD_0 register. This will release the SCL line and allow the Master to continue the transfer.

Slave Response to Read-to-Write Transfer Direction Change with RepeatedStart Condition

When Master wants to send RepeatedStart condition to change the transfer direction from read to write, it will NACK the last byte of the read transfer before sending Repeated condition.

The Slave does the following when this happens:

1. Sets I2C_I2C_SL_STATUS_0[END_TRANS] and I2C_I2C_SL_STATUS_0[SL_IRQ] bits indicating the end of read transaction.
2. Generates an interrupt to the system if interrupt enables bits (I2C_I2C_SL_INT_MASK_0[END_TRANS], I2C_I2C_SL_INT_MASK_0[SL_IRQ]) are set.

Interrupts Handling When Transaction END is Received During Read

1. Read the I2C_I2C_SL_INT_SOURCE_0 register and check which bit fields are set to 1.
2. In the transaction END case, I2C_I2C_SL_INT_SOURCE_0[END_TRANS] and I2C_I2C_SL_INT_SOURCE_0[SL_IRQ] bits are set to 1.
3. Clear I2C_I2C_SL_STATUS_0[END_TRANS] and I2C_I2C_SL_STATUS_0[SL_IRQ] bits by writing 1 to them. This will clear the interrupt.

When New Transaction (READ) started with RepeatedStart, The Slave does the following after receiving the address byte of the new transaction:

1. Sets I2C_I2C_SL_STATUS_0[RNW] bit field with received address byte[0] bit which indicates transfer direction. 0 = WRITE, 1 = READ
2. Saves the received address byte in the I2C_I2C_SL_RCVD_0 register.
3. Sets I2C_I2C_SL_STATUS_0[RCVD] and I2C_I2C_SL_STATUS_0[SL_IRQ] bits indicating that a new transaction is received.
4. Generates an interrupt to the system if interrupt enables bits (I2C_I2C_SL_INT_MASK_0[RCVD], I2C_I2C_SL_INT_MASK_0[SL_IRQ]) are set.
5. Holds the I^2C bus by doing clock stretching (holds I^2C SCL line low) until the interrupt is cleared.

New Transaction Interrupt Handling

1. Read I2C_I2C_SL_INT_SOURCE_0 register and check which bit fields are set to 1.
2. In the new transaction received case, I2C_I2C_SL_INT_SOURCE_0[RCVD] and I2C_I2C_SL_INT_SOURCE_0[SL_IRQ] bits are set to 1.
3. Check I2C_I2C_SL_STATUS_0[RNW] bit status. For write transactions, I2C_I2C_SL_STATUS_0[RNW] = WRITE is set.
4. Clear I2C_I2C_SL_STATUS_0[RCVD] and I2C_I2C_SL_STATUS_0[SL_IRQ] bits by writing 1 to them. This clears the interrupt and releases the SCL line, allowing the Master to go to next phase of the transaction

Slave Response to Data Phase of the Write Transaction

When the Master sends a data byte, the Slave controller does the following:

1. Saves the received byte in the I2C_I2C_SL_RCVD_0 register.
2. Sets I2C_I2C_SL_STATUS_0[SL_IRQ] bit indicating a byte received.
3. Generates an interrupt to the system if interrupt enable bit I2C_I2C_SL_INT_MASK_0[SL_IRQ] is set.

4. The Slave will hold the I^2C bus by doing clock stretching (holds I^2C SCL line low) until the interrupt is cleared.

Interrupts Handling When Received a Data Byte

1. Read I2C_I2C_SL_INT_SOURCE_0 register and check which bit fields are set to 1.
2. In the data_byte received case, I2C_I2C_SL_INT_SOURCE_0[SL_IRQ] bit is set to 1.
3. Read the data byte from the I2C_I2C_SL_RCVD_0 register.
4. To continue with the transfer, keep I2C_I2C_SL_CNFG_0[NACK] = DISABLE setting. Otherwise, to signal the Master to stop the transfer, configure I2C_I2C_SL_CNFG_0[NACK] = ENABLE.
5. Clear I2C_I2C_SL_STATUS_0[SL_IRQ] by writing 1 to it. This clears the interrupt and releases the SCL line, allowing the Master to go ahead with next phase of the transaction.

Slave Response to STOP Condition of the Write Transaction

If the Master sends STOP condition next, the Slave does the following:

1. Sets I2C_I2C_SL_STATUS_0[END_TRANS] and I2C_I2C_SL_STATUS_0[SL_IRQ] bits indicating STOP condition is received.
2. Generates an interrupt to the system if Interrupt bits (I2C_I2C_SL_INT_MASK_0[END_TRANS], I2C_I2C_SL_INT_MASK_0[SL_IRQ]) are enabled.

Interrupts Handling When Stop Condition is Received

1. Read the I2C_I2C_SL_INT_SOURCE_0 register and check which bit fields are set to 1.
2. In the STOP condition case, I2C_I2C_SL_INT_SOURCE_0[END_TRANS] and I2C_I2C_SL_INT_SOURCE_0[SL_IRQ] bits are set to 1.
3. Clear I2C_I2C_SL_STATUS_0[END_TRANS] and I2C_I2C_SL_STATUS_0[SL_IRQ] bits by writing 1 to them. This will clear the interrupt.

Notes:

1. If the Master terminates the transaction while the Slave controller is ready to accept the data during write transfer, I2C_INTERRUPT_STATUS_REGISTER_0[SLV_PKT_XFER_ERR] bit is set at the end of the transaction when STOP condition is received. This bit can be cleared by writing 1 to it or it can be ignored.
2. If the Slave controller sends NACK to any of the bytes during the write transfer, the Master will terminate the transaction after seeing the NACK bit. In this case, no status bits (END_TRANS, SL_IRQ or SLV_PKT_XFER_ERR) would be set when STOP condition is received. No interrupts either.

9.6.4 I2C Registers

Refer to "Reading Register Tables" in the Introduction chapter for the register table protocol as well as recommendations for accessing registers.

This register set is identical for all the I^2C controllers.

I2C_I2C_CNFG_0

IC Controller Configuration Register

The I2C_CNFG register is used to configure:

- the number of bytes to be transmitted or received
- the slave device type (either a 7-bit device or a 10-bit device)
- enable mode to send Start-byte or not
- to select either a single slave transaction or two slave transactions
- enable mode to handle devices that do not generate an ACK.

Offset: 0x0
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00020800 (0bxxxx,xxxx,xxxx,xx10,0000,1000,0000,0000)

Bit	R/W	Reset	Description
17	RW	ENABLE	**MULTI_MASTER_MODE**: Used to select single or multi master mode 0 = single master; 1 = multi_master In single_master mode, no Arbitration checks would happen **0** = DISABLE **1** = ENABLE
15	RW	0x0	**MSTR_CLR_BUS_ON_TIMEOUT**: When this bit is set, I2C master will force clock low for an extended period of time(>TIMEOUT) to force all SMB slaves to release the bus 0 = Don't clear the bus on TLow: SEXT/TIMEOUT timeout 1 = Clear the bus on TLow: SEXT/TIMEOUT timeout
14:12	RW	0x0	**DEBOUNCE_CNT**: Debounce period for SDA and SCL lines 0 = No debounce 1 = 2T 2 = 4T 3 = 6T, etc. Where T is the period of the clk source coming to I2C. Maximum debounce period programmable is 14T. A debounce period of >50ns is desirable
11	RO	0x1	**NEW_MASTER_FSM**: Maintained for compatibility sake. **0** = DISABLE **1** = ENABLE
10	RW	0x0	**PACKET_MODE_EN**: Write 1 to initiate transfer in packet mode. **0** = NOP **1** = GO

Bit	R/W	Reset	Description
9	RW	0x0	**SEND**: Writing a 1 causes the master to initiate the transaction in normal mode. Values of other bits are not affected when this bit is 1, cleared by hardware. Other bits of the register are masked for writes when this bit is programmed to one. Hence, firmware should first configure all other registers and bits [8:0] of I2C_CNFG register before the bit I2C_CNFG[9] is programmed to Zero. **0** = NOP **1** = GO
8	RW	0x0	**NOACK**: Enable mode to handle devices that do not generate ACK. 1 - don't look for an ack at the end of the Enable **0** = DISABLE **1** = ENABLE
7	RW	0x0	**CMD2**: Read/Write Command for Slave 2: 1 - Read Transaction; 0 - write Transaction. For a 7-bit slave address, this bit must match with the LSB of address byte for slave 2. Valid only when bit-4 of this register is set. **0** = DISABLE **1** = ENABLE
6	RW	0x0	**CMD1**: Read/Write Command for Slave 1: 1 - Read Transaction; 0 - write Transaction. Command for Slave 1: For a 7-bit slave address this bit must match with the LSB of address byte for slave1. **0** = DISABLE **1** = ENABLE
5	RW	0x0	**START**: 1 = Yes, a Start byte needs to be sent. **0** = DISABLE **1** = ENABLE
4	RW	0x0	**SLV2**: 1 = Enables a two slave transaction 0 = No command for Slave 2 present. **0** = DISABLE **1** = ENABLE
3:1	RW	0x0	**LENGTH**: The Number of bytes to be transmitted per transaction 000 = 1byte ... 111 = 8bytes; In a two slave transaction number of bytes should be programmed less than or equal to 011. I2C transfer with Slave-address byte only is not supported in the controller.
0	RW	0x0	**A_MOD**: Address mode defines whether a 7-bit or a 10-bit slave address is programmed. 1 = 10-bit device address; 0 = 7-bit device address **0** = SEVEN_BIT_DEVICE_ADDRESS **1** = TEN_BIT_DEVICE_ADDRESS

I2C_I2C_CMD_ADDR0_0

I2C Slave-1 Address. I2C_CMD_ADDR0 is programmed the 7-bit or 10-bit address of slave 1 with which the transaction is intended;

Offset: 0x4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9:0	0x0	**ADDR0**: In case of 7-Bit mode address is written in the I2C_CMD_ADDR0[7:1] and I2C_CMD_ADDR0[0] indicates the read/write transaction.I2C_CMD_ADDR0[0] bit must match with the I2C_CNFG[6]. In case of 10-Bit mode address is written in I2C_CMD_ADDR0[9:0] and I2C_CNFG[6] indicates the read/write transaction.

I2C_I2C_CMD_ADDR1_0

I2C Slave-2 Address. I2C_CMD_ADDR1 is programmed the 7 Bit or 10 Bit address of slave 2 with which the transaction is intended;

Offset: 0x8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000,0000)

Bit	Reset	Description
9:0	0x0	**ADDR1**: In case of 7-Bit mode address is written in the I2C_CMD_ADDR0[7:1] and I2C_CMD_ADDR0[0] indicates the read/write transaction.I2C_CMD_ADDR0[0] bit must match with the I2C_CNFG[7]. In case of 10-Bit mode address is written in I2C_CMD_ADDR0[9:0] and I2C_CNFG[7] indicates the read/write transaction.

I2C_I2C_CMD_DATA1_0

IC Controller Data 1: Transmit/Receive The four Least Significant Bytes of Data to be Transmitted is loaded into the register when I2C Master is in Write Mode;
The four Least Significant Bytes of Data are Read through this register when I2C Master is in Read mode.

Offset: 0xc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**DATA4**: Fourth data byte to be sent/received
23:16	0x0	**DATA3**: Third data byte to be sent/received
15:8	0x0	**DATA2**: Second data byte to be sent/received
7:0	0x0	**DATA1**: This register contains the first data byte to be sent/received.

I2C_I2C_CMD_DATA2_0

IC Controller Data 2: Transmit/Receive The four Most Significant Bytes of Data to be Transmitted is loaded into the register when I^2C Master is in Write Mode;
The four Most Significant Bytes of Data are Read through this register when I^2C Master is in Read mode.

Offset: 0x10
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0x0	**DATA8**: Eighth data byte to be sent/received
23:16	0x0	**DATA7**: Seventh data byte to be sent/received
15:8	0x0	**DATA6**: Sixth data byte to be sent/received
7:0	0x0	**DATA5**: This register contains the Fifth data byte to be sent/received.

I2C_I2C_STATUS_0

IC Controller Status (Master). I2C_STATUS gives the status of I2C Master operation

Offset: 0x1c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,0000)

Bit	Reset	Description
8	0x0	**BUSY**: **0** = NOT_BUSY **1** = BUSY
7:4	0x0	**CMD2_STAT**: Transaction for Slave2 for x byte failed. x is 'h0 to 'ha, all others invalid. **0** = SL2_XFER_SUCCESSFUL **1** = SL2_NOACK_FOR_BYTE1 **2** = SL2_NOACK_FOR_BYTE2 **3** = SL2_NOACK_FOR_BYTE3 **4** = SL2_NOACK_FOR_BYTE4 **5** = SL2_NOACK_FOR_BYTE5 **6** = SL2_NOACK_FOR_BYTE6 **7** = SL2_NOACK_FOR_BYTE7 **8** = SL2_NOACK_FOR_BYTE8 **9** = SL2_NOACK_FOR_BYTE9 **10** = SL2_NOACK_FOR_BYTE10
3:0	0x0	**CMD1_STAT**: Transaction for Slave1 for x byte failed. x is 'h0 to 'ha, all others invalid. **0** = SL1_XFER_SUCCESSFUL **1** = SL1_NOACK_FOR_BYTE1 **2** = SL1_NOACK_FOR_BYTE2 **3** = SL1_NOACK_FOR_BYTE3 **4** = SL1_NOACK_FOR_BYTE4 **5** = SL1_NOACK_FOR_BYTE5 **6** = SL1_NOACK_FOR_BYTE6 **7** = SL1_NOACK_FOR_BYTE7 **8** = SL1_NOACK_FOR_BYTE8 **9** = SL1_NOACK_FOR_BYTE9 **10** = SL1_NOACK_FOR_BYTE10

I2C_I2C_SL_CNFG_0

IC Controller Configuration (Slave). The register should be programmed when I^2C controller is configured as slave.

Offset: 0x20
Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000004 (0bxxxx,xxxx,xx00,0000,0000,0000,0000,0100)

Bit	R/W	Reset	Description
21	RW	0x0	**SLV_XFER_ERR_CLK_STRETCH_EN**: This bit is used in FIFO mode only. If a transfer resulted in SLV_PKT_XFER_ERR, enabling this bit makes the Slave to stretch the I2C SCL line after Address cycle and ACK bit of following transfer. This helps Software in cases where it is not ready to attend SLV_PKT_XFER_ERR immediately and wants to stall the Master until that time. Once SLV_PKT_XFER_ERR status is cleared by Software, SCL line will be released by the Slave. **0** = DISABLE **1** = ENABLE
20	RW	0x0	**FIFO_XFER_EN**: If this bit is disabled, data is always communicated via I2C_SL_RCVD register. If enabled, it is through FIFOs **0** = DISABLE **1** = ENABLE
19:8	RW	0x0	**BUFFER_SIZE**: Payload size in bytes. RSVD
7	RW	0x0	**ACK_LAST_BYTE_VALID**: ack the last byte valid(Write-Only) This bit qualifies ACK_LAST_BYTE field **0** = DISABLE **1** = ENABLE
6	RW	0x0	**ACK_LAST_BYTE**: ack the last byte **0** = DISABLE **1** = ENABLE
5	RW	0x0	**ACK_WITHHOLD_EN**: Ack Withhold Feature Enable **0** = DISABLE **1** = ENABLE
4	RW	0x0	**PKT_MODE_EN**: Packet Mode Enable **0** = DISABLE **1** = ENABLE
3	RW	0x0	**ENABLE_SL**: By writing zero to this field, slave can be turned off **0** = DISABLE **1** = ENABLE
2	RO	0x1	**NEWSL**: New Slave 1 - use new slave **0** = DISABLE **1** = ENABLE

Bit	R/W	Reset	Description
1	RW	0x0	**NACK**: Disable Slave Ack. 1 = slave will not ack reception of address or data byte. **0** = DISABLE **1** = ENABLE
0	RW	0x0	**RESP**: Slave response to general call address (zero address) 1 = Enable. **0** = DISABLE **1** = ENABLE

I2C_I2C_SL_RCVD_0

IC Controller Slave Receive/Transmit Data (Slave)

Offset: 0x24
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**SL_DATA**: Slave Received data

I2C_I2C_SL_STATUS_0

IC Controller Slave Status (Slave)

Offset: 0x28
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,0000,0000,0000)

Bit	R/W	Reset	Description
14:8	RO	0x0	**HW_MSTR_ADR**: Hardware master address received via general call addressing. This field is meaningful only if HW_MSTR_INT is set.

Bit	R/W	Reset	Description
7	RW	0x0	**HW_MSTR_INT**: 1 = Interrupt has been generated by slave Hardware Master Address is received after General Call Address. 1 = Received Hardware Master Address 0 = No event.
6	RW	0x0	**REPROG_SL**: 1 = Interrupt has been generated by slave By after General Call Address is 0x04. 1 = Reprogram slave address. 0 = No action.
5	RW	0x0	**RST_SL**: 1 = Interrupt has been generated by slave By after General Call Address is 0x06. 1 = Reset and reprogram slave address. 0 = No action.
4	RW	0x0	**END_TRANS**: 1 = Interrupt has been generated by slave Transaction completed as indicated by stop/repeat start condition. 1 = Transaction completed. 0 = No transaction occurred or transaction in progress.
3	RW	0x0	**SL_IRQ**: 1 = Interrupt has been generated by slave 0 = No interrupt generated **0** = UNSET **1** = SET
2	RW	0x0	**RCVD**: New Transaction Received status 1 = Transaction occurred. 0 = No transaction occurred **0** = NO_TRANSACTION_OCCURED **1** = TRANSACTION_OCCURED
1	RO	0x0	**RNW**: Slave Transaction status 0 = Write 1 = Read **0** = WRITE **1** = READ
0	RW	0x0	**ZA**: Zero Address Status 1 = Yes, slave responded 0 = No, slave did not respond **0** = NO_SLAVE_RESPONSE **1** = SLAVE_RESPONSE

I2C_I2C_SL_ADDR1_0

IC Controller Slave Address 1 Register (Slave)

Offset: 0x2c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:8	0x0	**SL_ADDR1**: For a 10-bit slave address, this field is the least significant 8 bits.
7:0	0x0	**SL_ADDR0**: For a 10-bit slave address, this field is the least significant 8 bits.

I2C_I2C_SL_ADDR2_0

IC Controller Slave Address 2 Register (Slave)

Offset: 0x30
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxx0,xxxx,x000,xxxx,x000)

Bit	Reset	Description
16	0x0	**SELECT_SLAVE**: 0 = Use slave addr0 1 = Use slave addr1
10:9	0x0	**SL1_ADDR_HI**: In 7 bit address mode these bits are don't care; In 10 bit address mode they represent the 2 MSB of the address.
8	0x0	**SL1_VLD**: 0 = 7-bit addressing. **0** = SEVEN_BIT_ADDR_MODE **1** = TEN_BIT_ADDR_MODE
2:1	0x0	**SL_ADDR_HI**: In 7-bit address mode these bits are don't care; In 10-bit address mode they represent the 2 MSB of the address.
0	0x0	**VLD**: 0 = 7-bit addressing. 1 = 10-bit addressing. **0** = SEVEN_BIT_ADDR_MODE **1** = TEN_BIT_ADDR_MODE

I2C_I2C_TLOW_SEXT_0

IC Controller SMBUS timeout thresholds

Offset: 0x34
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
27	0x0	**RST_SL_ON_TIMEOUT**: Reset Slave state machine on time-out
26	0x0	**TLOW_MEXT_EN**: Enable TLOW_MEXT counter
25	0x0	**TLOW_SEXT_EN**: Enable TLOW_SEXT counter
24	0x0	**TIMEOUT_EN**: Enable TIMEOUT counter
23:16	0x0	**TLOW_MEXT**: Cumulative clock low extend time (master device) accumulated over a byte transfer period in milli-seconds (START to ACK, ACK to ACK, or ACK to STOP).
15:8	0x0	**TLOW_SEXT**: Cumulative clock low extend time (slave device) accumulated over a complete transfer (START until STOP)
7:0	0x0	**TIMEOUT**: Clock low timeout period in milli-seconds

I2C_I2C_SL_DELAY_COUNT_0

IC Slave Controller Delay Count

Offset: 0x3c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000001e (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0001,1110)

Bit	Reset	Description
15:0	0x1e	**SL_DELAY_COUNT**: The value determines the timing between an address cycle and a subsequent data cycle or two consecutive data cycles on the bus. The I2C_SL_DELAY_COUNT is valid only when internal slave is accessed. I2C_SL_DELAY_COUNT has to be programmed such that TIMING = T * DLY where T is period of clock source selected for I2c; and DLY is I2C_SL_DELAY_COUNT; TIMING is the desired timing, A value of >= 1250 ns is advisable.

I2C_I2C_SL_INT_MASK_0

IC Controller Slave Mask (Slave)

Offset: 0x40
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000000fd (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1111,11x1)

Bit	Reset	Description
7	0x1	**HW_MSTR_INT**: **0** = DISABLE **1** = ENABLE
6	0x1	**REPROG_SL**: **0** = DISABLE **1** = ENABLE
5	0x1	**RST_SL**: **0** = DISABLE **1** = ENABLE
4	0x1	**END_TRANS**: **0** = DISABLE **1** = ENABLE
3	0x1	**SL_IRQ**: **0** = DISABLE **1** = ENABLE
2	0x1	**RCVD**: **0** = DISABLE **1** = ENABLE
0	0x1	**ZA**: **0** = DISABLE **1** = ENABLE

I2C_I2C_SL_INT_SOURCE_0

IC Controller Slave Source (Slave)

Offset: 0x44
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,00x0)

Bit	Reset	Description
7	0x0	**HW_MSTR_INT**: **0** = UNSET **1** = SET
6	0x0	**REPROG_SL**: **0** = UNSET **1** = SET
5	0x0	**RST_SL**: **0** = UNSET **1** = SET
4	0x0	**END_TRANS**: **0** = UNSET **1** = SET
3	0x0	**SL_IRQ**: **0** = UNSET **1** = SET
2	0x0	**RCVD**: **0** = UNSET **1** = SET
0	0x0	**ZA**: **0** = UNSET **1** = SET

I2C_I2C_SL_INT_SET_0

IC Controller Slave Source (Slave)

Offset: 0x48
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,00x0)

Bit	Reset	Description
7	0x0	**HW_MSTR_INT**: **0** = UNSET **1** = SET

Bit	Reset	Description
6	0x0	**REPROG_SL**: **0** = UNSET **1** = SET
5	0x0	**RST_SL**: **0** = UNSET **1** = SET
4	0x0	**END_TRANS**: **0** = UNSET **1** = SET
3	0x0	**SL_IRQ**: **0** = UNSET **1** = SET
2	0x0	**RCVD**: **0** = UNSET **1** = SET
0	0x0	**ZA**: **0** = UNSET **1** = SET

I2C_I2C_TX_PACKET_FIFO_0

Offset: 0x50
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TX_PACKET**: Software writes packets into this register.

I2C_I2C_RX_FIFO_0

Header or I2C specific header or data

Offset: 0x54
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**RD_DATA**: Software Reads data from this register, causes pop

I2C_PACKET_TRANSFER_STATUS_0

Offset: 0x58
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxx0,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
24	0x0	**TRANSFER_COMPLETE**: The packet transfer for which last packet is set has been completed **0** = UNSET **1** = SET
23:16	0x0	**TRANSFER_PKT_ID**: The current packet id for which the transaction is happening on the bus
15:4	0x0	**TRANSFER_BYTENUM**: RSVD The number of bytes transferred in the current packet
3	0x0	**NOACK_FOR_ADDR**: No ack received for the address byte **0** = UNSET **1** = SET
2	0x0	**NOACK_FOR_DATA**: No ack received for the data byte **0** = UNSET **1** = SET
1	0x0	**ARB_LOST**: Arbitration lost for the current byte **0** = UNSET **1** = SET
0	0x0	**CONTROLLER_BUSY**: 1 = Controller is busy **0** = UNSET **1** = SET

I2C_FIFO_CONTROL_0

This register is RSVD and not used
Offset: 0x5c
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:13	0x0	**SLV_TX_FIFO_TRIG**: Slave Transmit FIFO trigger level 000 = 1 word, DMA trigger is asserted when at least one word empty in the FIFO 001 = 2 word, DMA trigger is asserted when at least two words empty in the FIFO
12:10	0x0	**SLV_RX_FIFO_TRIG**: Slave Receive FIFO trigger level 000 = 1 word DMA trigger is asserted when at least one word full in the FIFO 001 = 2 word DMA trigger is asserted when at least two word full in the FIFO
9	0x0	**SLV_TX_FIFO_FLUSH**: 1= flush the Tx FIFO, cleared after FIFO is flushed. **0** = UNSET **1** = SET
8	0x0	**SLV_RX_FIFO_FLUSH**: 1= flush the Rx FIFO, cleared after FIFO is flushed. **0** = UNSET **1** = SET
7:5	0x0	**TX_FIFO_TRIG**: Transmit FIFO trigger level 000 = 1 word, DMA trigger is asserted when at least one word empty in the FIFO 001 = 2 word, DMA trigger is asserted when at least two words empty in the FIFO
4:2	0x0	**RX_FIFO_TRIG**: Receive FIFO trigger level 000 = 1 word DMA trigger is asserted when at least one word full in the FIFO 001 = 2 word DMA trigger is asserted when at least two word full in the FIFO
1	0x0	**TX_FIFO_FLUSH**: 1= flush the Tx FIFO, cleared after FIFO is flushed. **0** = UNSET **1** = SET
0	0x0	**RX_FIFO_FLUSH**: 1= flush the Rx FIFO, cleared after FIFO is flushed. **0** = UNSET **1** = SET

I2C_FIFO_STATUS_0

This register is RSVD and not used
Offset: 0x60
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00800080 (0bxxxx,xx0x,1000,0000,xxxx,xxxx,1000,0000)

Bit	Reset	Description
25	0x0	**SLV_XFER_ERR_REASON**: This bit describes the nature of the packet transfer error. It is meaningful only if PKT_XFER_ERR is set: 0 = Master terminated transaction before it was completed 1 = Master did not terminate transaction when all bytes are transferred
23:20	0x8	**SLV_TX_FIFO_EMPTY_CNT**: The number of slots that can be written to the slave Tx FIFO 0000 = tx_fifo full 0001 = 1 slot empty 0010 = 2 slots empty
19:16	0x0	**SLV_RX_FIFO_FULL_CNT**: The number of slots to be read from the Slave Rx FIFO 0000 = rx_fifo empty 0001 = 1 slot full 0010 = 2 slots full
7:4	0x8	**TX_FIFO_EMPTY_CNT**: The number of slots that can be written to the Tx FIFO 0000 = tx_fifo full 0001 = 1 slot empty 0010 = 2 slots empty
3:0	0x0	**RX_FIFO_FULL_CNT**: The number of slots to be read from the Rx FIFO 0000 = rx_fifo empty 0001 = 1 slot full 0010 = 2 slots full

I2C_INTERRUPT_MASK_REGISTER_0

Offset: 0x64
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxx0,0000,0000,xx00,xxxx,0000,0000,0000)

Bit	Reset	Description
28	0x0	**SLV_ACK_WITHHELD_INT_EN**: **0** = DISABLE **1** = ENABLE
27	0x0	**SLV_RD2WR_INT_EN**: **0** = DISABLE **1** = ENABLE
26	0x0	**SLV_WR2RD_INT_EN**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
25	0x0	**SLV_PKT_XFER_ERR_INT_EN**: **0** = DISABLE **1** = ENABLE
24	0x0	**SLV_TX_BUFFER_REQ_INT_EN**: **0** = DISABLE **1** = ENABLE
23	0x0	**SLV_RX_BUFFER_FILLED_INT_EN**: **0** = DISABLE **1** = ENABLE
22	0x0	**SLV_PACKET_XFER_COMPLETE_INT_EN**: **0** = DISABLE **1** = ENABLE
21	0x0	**SLV_TFIFO_OVF_REQ_INT_EN**: **0** = DISABLE **1** = ENABLE
20	0x0	**SLV_RFIFO_UNF_REQ_INT_EN**: **0** = DISABLE **1** = ENABLE
17	0x0	**SLV_TFIFO_DATA_REQ_INT_EN**: **0** = DISABLE **1** = ENABLE
16	0x0	**SLV_RFIFO_DATA_REQ_INT_EN**: **0** = DISABLE **1** = ENABLE
11	0x0	**BUS_CLEAR_DONE_INT_EN**: **0** = DISABLE **1** = ENABLE
10	0x0	**TLOW_MEXT_TIMEOUT_EN**: **0** = DISABLE **1** = ENABLE
9	0x0	**TLOW_SEXT_TIMEOUT_EN**: **0** = DISABLE **1** = ENABLE
8	0x0	**TIMEOUT_INT_EN**: **0** = DISABLE **1** = ENABLE
7	0x0	**PACKET_XFER_COMPLETE_INT_EN**: **0** = DISABLE **1** = ENABLE
6	0x0	**ALL_PACKETS_XFER_COMPLETE_INT_EN**: **0** = DISABLE **1** = ENABLE
5	0x0	**TFIFO_OVF_INT_EN**: **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
4	0x0	**RFIFO_UNF_INT_EN**: **0** = DISABLE **1** = ENABLE
3	0x0	**NOACK_INT_EN**: **0** = DISABLE **1** = ENABLE
2	0x0	**ARB_LOST_INT_EN**: **0** = DISABLE **1** = ENABLE
1	0x0	**TFIFO_DATA_REQ_INT_EN**: **0** = DISABLE **1** = ENABLE
0	0x0	**RFIFO_DATA_REQ_INT_EN**: **0** = DISABLE **1** = ENABLE

I2C_INTERRUPT_STATUS_REGISTER_0

This register indicates the status bit for which the interrupt is set.

Offset: 0x68
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxx0,0000,0000,xx00,xxxx,0000,0000,0000)

Bit	R/W	Reset	Description
28	RW	0x0	**SLV_ACK_WITHHELD**: ack is withheld, waiting for Software explicit info about ack. **0** = UNSET **1** = SET
27	RW	0x0	**SLV_RD2WR**: Transaction switching from read to write. **0** = UNSET **1** = SET
26	RW	0x0	**SLV_WR2RD**: Transaction switching from write to read. **0** = UNSET **1** = SET
25	RW	0x0	**SLV_PKT_XFER_ERR**: 0 = Request was successful 1 = Error has occurred during packet transfer **0** = UNSET **1** = SET

Bit	R/W	Reset	Description
24	RW	0x0	**SLV_TX_BUFFER_REQ**: Slave Tx buffer is full **0** = UNSET **1** = SET
23	RW	0x0	**SLV_RX_BUFFER_FILLED**: Slave Rx buffer is full **0** = UNSET **1** = SET
22	RW	0x0	**SLV_PACKET_XFER_COMPLETE**: Slave packet transfer complete **0** = UNSET **1** = SET
21	RW	0x0	**SLV_TFIFO_OVF**: slave Tx FIFO overflow **0** = UNSET **1** = SET
20	RW	0x0	**SLV_RFIFO_UNF**: slave Rx FIFO underflow **0** = UNSET **1** = SET
17	RO	0x0	**SLV_TFIFO_DATA_REQ**: Slave Tx FIFO data req **0** = UNSET **1** = SET
16	RO	0x0	**SLV_RFIFO_DATA_REQ**: Slave Rx FIFO data req **0** = UNSET **1** = SET
11	RW	0x0	**BUS_CLEAR_DONE**: bus clear done status **0** = UNSET **1** = SET
10	RW	0x0	**TLOW_MEXT_TIMEOUT**: SMBUS mext time-out **0** = UNSET **1** = SET
9	RW	0x0	**TLOW_SEXT_TIMEOUT**: SMBUS sext time-out **0** = UNSET **1** = SET
8	RW	0x0	**TIMEOUT**: SMBUS time-out **0** = UNSET **1** = SET
7	RW	0x0	**PACKET_XFER_COMPLETE**: A packet has been transferred successfully. TRANSFER_PKT_ID filed can be used to know the current byte under transfer. This bit can be masked by the IE field in the i2c specific header **0** = UNSET **1** = SET

Bit	R/W	Reset	Description
6	RW	0x0	**ALL_PACKETS_XFER_COMPLETE**: All the packets transferred successfully **0** = UNSET **1** = SET
5	RW	0x0	**TFIFO_OVF**: Tx FIFO overflow **0** = UNSET **1** = SET
4	RW	0x0	**RFIFO_UNF**: Rx FIFO underflow **0** = UNSET **1** = SET
3	RW	0x0	**NOACK**: No ACK from slave **0** = UNSET **1** = SET
2	RW	0x0	**ARB_LOST**: Arbitration lost **0** = UNSET **1** = SET
1	RO	0x0	**TFIFO_DATA_REQ**: Tx FIFO data req **0** = UNSET **1** = SET
0	RO	0x0	**RFIFO_DATA_REQ**: Rx FIFO data req **0** = UNSET **1** = SET

I2C_I2C_CLK_DIVISOR_REGISTER_0

The divisor values (N) must be programmed so that desired
scl freq (std/fast/fm+ modes) = ClkSourceFreq/((tlow+thigh+3)*(N+1)) for lower values of N, upto 3
scl freq (std/fast/fm+ modes) = ClkSourceFreq/((tlow+thigh+2)*(N+1)) for higher values of N, above 3
scl freq (HS mode) = ClkSourceFreq/((ths_low+ths_high+4)*(N+1)) for lower values of N, upto 4
scl freq (HS mode) = ClkSourceFreq/((ths_low+ths_high+2)*(N+1)) for higher values of N, above 4

Offset: 0x6c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00190001 (0b0000,0000,0001,1001,0000,0000,0000,0001)

Bit	Reset	Description
31:16	0x19	**I2C_CLK_DIVISOR_STD_FAST_MODE**: N = divide by n+1
15:0	0x1	**I2C_CLK_DIVISOR_HSMODE**: N = divide by n+1

I2C_I2C_INTERRUPT_SOURCE_REGISTER_0

This is a read-only register which returns the AND of Interrupt Status and Interrupt Mask registers.

Offset: 0x70
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxx0,0000,0000,xx00,xxxx,0000,0000,0000)

Bit	Reset	Description
28	0x0	**SLV_ACK_WITHHELD**: **0** = UNSET **1** = SET
27	0x0	**SLV_RD2WR**: **0** = UNSET **1** = SET
26	0x0	**SLV_WR2RD**: **0** = UNSET **1** = SET
25	0x0	**SLV_PKT_XFER_ERR**: Error occurred during slave xfer **0** = UNSET **1** = SET
24	0x0	**SLV_TX_BUFFER_REQ**: slave Tx buffer is full **0** = UNSET **1** = SET
23	0x0	**SLV_RX_BUFFER_FILLED**: slave Rx buffer is full **0** = UNSET **1** = SET
22	0x0	**SLV_PACKET_XFER_COMPLETE**: slave packet transfer complete **0** = UNSET **1** = SET
21	0x0	**SLV_TFIFO_OVF**: slave Tx FIFO overflow **0** = UNSET **1** = SET

Bit	Reset	Description
20	0x0	**SLV_RFIFO_UNF**: slave Rx FIFO underflow **0** = UNSET **1** = SET
17	0x0	**SLV_TFIFO_DATA_REQ**: slave Tx FIFO data req **0** = UNSET **1** = SET
16	0x0	**SLV_RFIFO_DATA_REQ**: slave Rx FIFO data req **0** = UNSET **1** = SET
11	0x0	**BUS_CLEAR_DONE**: bus clear done **0** = UNSET **1** = SET
10	0x0	**TLOW_MEXT_TIMEOUT**: SMBUS mext time-out **0** = UNSET **1** = SET
9	0x0	**TLOW_SEXT_TIMEOUT**: SMBUS sext time-out **0** = UNSET **1** = SET
8	0x0	**TIMEOUT**: SMBUS time-out **0** = UNSET **1** = SET
7	0x0	**PACKET_XFER_COMPLETE**: packet transferred successfully **0** = UNSET **1** = SET
6	0x0	**ALL_PACKETS_XFER_COMPLETE**: All the packets transferred successfully **0** = UNSET **1** = SET
5	0x0	**TFIFO_OVF**: Tx FIFO overflow **0** = UNSET **1** = SET
4	0x0	**RFIFO_UNF**: Rx FIFO underflow **0** = UNSET **1** = SET
3	0x0	**NOACK**: No ACK from slave **0** = UNSET **1** = SET

Bit	Reset	Description
2	0x0	**ARB_LOST**: Arbitration lost **0** = UNSET **1** = SET
1	0x0	**TFIFO_DATA_REQ**: Tx FIFO data req **0** = UNSET **1** = SET
0	0x0	**RFIFO_DATA_REQ**: Rx FIFO data req **0** = UNSET **1** = SET

I2C_I2C_INTERRUPT_SET_REGISTER_0

This is a write-only register which can be used to set the interrupt status bit. A write to this register causes the bits in status register to be set if the corresponding bit in write data is 1'b1. Read always returns 'h0.

Offset: 0x74
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxx0,0000,0000,xxxx,xxxx,0000,0000,00xx)

Bit	Reset	Description
28	0x0	**SLV_ACK_WITHHELD**: **0** = UNSET **1** = SET
27	0x0	**SLV_RD2WR**: **0** = UNSET **1** = SET
26	0x0	**SLV_WR2RD**: **0** = UNSET **1** = SET
25	0x0	**SLV_PKT_XFER_ERR**: Error occurred during slave xfer **0** = UNSET **1** = SET
24	0x0	**SLV_TX_BUFFER_REQ**: slave Tx buffer is full **0** = UNSET **1** = SET
23	0x0	**SLV_RX_BUFFER_FILLED**: slave Rx buffer is full **0** = UNSET **1** = SET

Bit	Reset	Description
22	0x0	**SLV_PACKET_XFER_COMPLETE**: slave packet transfer complete **0** = UNSET **1** = SET
21	0x0	**SLV_TFIFO_OVF**: slave Tx FIFO overflow **0** = UNSET **1** = SET
20	0x0	**SLV_RFIFO_UNF**: slave Rx FIFO underflow **0** = UNSET **1** = SET
11	0x0	**BUS_CLEAR_DONE**: bus clear done **0** = UNSET **1** = SET
10	0x0	**TLOW_MEXT_TIMEOUT**: SMBUS mext time-out **0** = UNSET **1** = SET
9	0x0	**TLOW_SEXT_TIMEOUT**: SMBUS sext time-out **0** = UNSET **1** = SET
8	0x0	**TIMEOUT**: SMBUS time-out **0** = UNSET **1** = SET
7	0x0	**PACKET_XFER_COMPLETE**: packet transferred successfully **0** = UNSET **1** = SET
6	0x0	**ALL_PACKETS_XFER_COMPLETE**: All the packets transferred successfully **0** = UNSET **1** = SET
5	0x0	**TFIFO_OVF**: Tx FIFO overflow **0** = UNSET **1** = SET
4	0x0	**RFIFO_UNF**: Rx FIFO underflow **0** = UNSET **1** = SET
3	0x0	**NOACK**: No ACK from slave **0** = UNSET **1** = SET

Bit	Reset	Description
2	0x0	**ARB_LOST**: Arbitration lost **0** = UNSET **1** = SET

I2C_I2C_SLV_TX_PACKET_FIFO_0

Offset: 0x78
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TX_PACKET**: Software writes packets into this register

I2C_I2C_SLV_RX_FIFO_0

Header or I2C specific header or data

Offset: 0x7c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**RD_DATA**: Software Reads data from this register, causes pop

I2C_I2C_SLV_PACKET_STATUS_0

Offset: 0x80
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xx00,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
25	0x0	**ACK_WITHHELD**: Indicates that ack is withheld for last byte and slave is waiting for host to explicitly command slave to ACK the last byte 0 = Bus is released 1 = ACK is withheld
24	0x0	**TRANSFER_COMPLETE**: ALL the packets have been transferred successfully
23:16	0x0	**TRANSFER_PKT_ID**: The current packet id for which the transaction is happening on the bus
15:0	0x0	**TRANSFER_BYTENUM**: The number of bytes transferred in the current packet

I2C_I2C_BUS_CLEAR_CONFIG_0

Offset: 0x84
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00090004 (0bxxxx,xxxx,0000,1001,xxxx,xxxx,xxxx,x100)

Bit	Reset	Description
23:16	0x9	**BC_SCLK_THRESHOLD**: Send the clock pulses until this threshold is met
2	0x1	**BC_STOP_COND**: **0** = NO_STOP: do not send stop condition at the end of bus clear operation **1** = STOP: send stop condition at the end of the bus clear operation
1	0x0	**BC_TERMINATE**: **0** = THRESHOLD: irrespective of SDA release status during BC, terminate the BC only after threshold is reached. **1** = IMMEDIATE: terminate the bus clear operation immediately when SDA is released or threshold count is reached whichever is earlier
0	0x0	**BC_ENABLE**: Starts bus clear operation, Hardware auto-clears this bit upon bus clear transaction completion

I2C_I2C_BUS_CLEAR_STATUS_0

Offset: 0x88
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**BC_STATUS**: **0** = NOT_CLEARED: indicates SDA is not released by slave, its status is still low. **1** = CLEARED: SDA is released

I2C_I2C_CONFIG_LOAD_0

Offset: 0x8c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**TIMEOUT_CONFIG_LOAD**: This bit loads the timeout configuration from pclk domain to the receive (i2c_slow_clk) domain. Software has to set this bit at the end after doing the required registers configuration for the logic to take the updates. Once the internal update is done, Hardware auto clears this bit. Since the Hardware would be busy with internal update, Software should not write again until this bit is cleared by Hardware. **0** = DISABLE **1** = ENABLE
1	0x0	**SLV_CONFIG_LOAD**: This bit loads the slave configuration from pclk domain to the receive (i2c_clk) domain. Software has to set this bit finally after doing the required registers configuration for the slave controller to take updates. Once the internal update is done, Hardware auto clears this bit. Since the Hardware would be busy with internal update, Software should not write again until this bit is cleared by Hardware. **0** = DISABLE **1** = ENABLE
0	0x0	**MSTR_CONFIG_LOAD**: This bit loads the master configuration from pclk domain to the receive (i2c_clk) domain. Software has to set this bit finally after doing the required registers configuration like I2C_I2C_CNFG_0 bit fields etc. Once the internal update is done, Hardware auto clears this bit. Since the Hardware would be busy with internal update, Software should not write again until this bit is cleared by Hardware. **0** = DISABLE **1** = ENABLE

I2C_I2C_CLKEN_OVERRIDE_0

Offset: 0x90
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0,00x0)

Bit	Reset	Description
4	CLK_GATED	**I2C_BUS_CLEAR_CLKEN_OVR**: Override for 2nd-level clock enable for I^2C bus clear logic **0** = CLK_GATED **1** = CLK_ALWAYS_ON
3	CLK_GATED	**I2C_SLV_HIF_CLKEN_OVR**: Override for 2nd-level clock enable for I^2C slave to host interface logic **0** = CLK_GATED **1** = CLK_ALWAYS_ON
2	CLK_GATED	**I2C_SLV_CORE_CLKEN_OVR**: Override for 2nd-level clock enable for I^2C slave core logic **0** = CLK_GATED **1** = CLK_ALWAYS_ON
0	CLK_GATED	**I2C_MST_CORE_CLKEN_OVR**: Override for 2nd-level clock enable for I^2C master core logic **0** = CLK_GATED **1** = CLK_ALWAYS_ON

I2C_I2C_INTERFACE_TIMING_0_0

Register for Standard/Fast/Fm+ mode timing
Offset: 0x94
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000204 (0bxxxx,xxxx,xxxx,xxxx,0000,0010,0000,0100)

Bit	Reset	Description
15:8	0x2	**THIGH**: High period of the SCL clock
7:0	0x4	**TLOW**: Low period of the SCL clock

I2C_I2C_INTERFACE_TIMING_1_0

Register for Standard/Fast/Fm+ mode timing
Offset: 0x98
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x04070404 (0b0000,0100,0000,0111,0000,0100,0000,0100)

Bit	Reset	Description
31:24	0x4	**TBUF**: Bus free time between STOP and START conditions
23:16	0x7	**TSU_STO**: Set-up time for STOP condition
15:8	0x4	**THD_STA**: hold time for a (repeated) START condition
7:0	0x4	**TSU_STA**: set-up time for a Repeated START condition

I2C_I2C_HS_INTERFACE_TIMING_0_0
I2C interface timing register for HS mode transfers

Offset: 0x9c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000308 (0bxxxx,xxxx,xxxx,xxxx,0000,0011,0000,1000)

Bit	Reset	Description
15:8	0x3	**HS_THIGH**: High period of the SCL clock
7:0	0x8	**HS_TLOW**: Low period of the SCL clock

I2C_I2C_HS_INTERFACE_TIMING_1_0
I2C interface timing register for HS mode transfers

Offset: 0xa0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x000b0b0b (0bxxxx,xxxx,0000,1011,0000,1011,0000,1011)

Bit	Reset	Description
23:16	0xb	**HS_TSU_STO**: Set-up time for STOP condition
15:8	0xb	**HS_THD_STA**: Hold time for a (repeated) START condition

Bit	Reset	Description
7:0	0xb	**HS_TSU_STA**: Set-up time for a Repeated START condition

I2C_I2C_MASTER_RESET_CNTRL_0

I2C Master reset control register
Offset: 0xa8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: If SOFT_RESET is set to 1, it will reset all internal state of Master logic including FIFOs. But registers configuration is retained. Clear this bit to 0 for normal operation. I2C transfers: Wait 2µs after reset assertion and also after clear, before going for any data transfers. SMBUS transfers where Timeout feature used: Wait 5ms after reset assertion and clear. **0** = CLEARED **1** = ASSERTED

I2C_I2C_SLV_RESET_CNTRL_0

I2C Slave reset control register
Offset: 0xac
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SOFT_RESET**: If SOFT_RESET is set to 1, it will reset all internal state of Slave logic including FIFOs. But registers configuration is retained. Clear this bit to 0 for normal operation. I2C transfers: Wait 2µs after reset assertion and also after clear, before going for any data transfers. SMBUS transfers where Timeout feature used: Wait 5ms after reset assertion and clear. **0** = CLEARED **1** = ASSERTED

I2C_MST_PACKET_TRANSFER_CNT_STATUS_0

Offset: 0xb0
Read/Write: RO

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**TRANSFER_BYTENUM**: The number of bytes transferred in the current packet

I2C_MST_FIFO_CONTROL_0

Offset: 0xb4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,x000,0000,xxxx,x000,0000,xx00)

Bit	Reset	Description
22:16	0x0	**TX_FIFO_TRIG**: Master Transmit FIFO trigger level. 0000000 = 1 word, trigger is asserted when at least one word empty in the FIFO. 0000001 = 2 words, trigger is asserted when at least two words empty in the FIFO..... 1111111 = 128 words, trigger is asserted when 128 words empty in the FIFO.
10:4	0x0	**RX_FIFO_TRIG**: Master Receive FIFO trigger level. 0000000 = 1 word, trigger is asserted when at least one word full in the FIFO. 0000001 = 2 words, trigger is asserted when at least two words full in the FIFO..... 1111111 = 128 words, trigger is asserted when 128 words full in the FIFO.
1	0x0	**TX_FIFO_FLUSH**: 1 = flush the TX FIFO; bit is auto-cleared by Hardware after FIFO flush is done **0** = UNSET **1** = SET
0	0x0	**RX_FIFO_FLUSH**: 1 = flush the RX FIFO; bit is auto-cleared by Hardware after FIFO flush is done **0** = UNSET **1** = SET

I2C_MST_FIFO_STATUS_0

Offset: 0xb8
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00800000 (0bxxxx,xxxx,1000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
23:16	0x80	**TX_FIFO_EMPTY_CNT**: The number of slots that can be written to the Master Tx FIFO 00000000 = tx_fifo full 00000001 = 1 slot empty 00000010 = 2 slots empty.... 10000000 = 128 slots empty
7:0	0x0	**RX_FIFO_FULL_CNT**: The number of slots to be read from the Slave Rx FIFO 00000000 = rx_fifo empty 00000001 = 1 slot full 00000010 = 2 slots full .... 10000000 = 128 slots full ....

I2C_SLV_PAYLOAD_0

Offset: 0xbc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**BUFFER_SIZE**: Payload size in bytes

I2C_SLV_FIFO_CONTROL_0

Offset: 0xc0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,x000,0000,xxxx,x000,0000,xx00)

Bit	Reset	Description
22:16	0x0	**TX_FIFO_TRIG**: Slave Transmit FIFO trigger level. 0000000 = 1 word, trigger is asserted when at least one word empty in the FIFO. 0000001 = 2 words, trigger is asserted when at least two words empty in the FIFO..... 1111111 = 128 words, trigger is asserted when 128 words empty in the FIFO.

Bit	Reset	Description
10:4	0x0	**RX_FIFO_TRIG**: Slave Receive FIFO trigger level 0000000 = 1 word, trigger is asserted when at least one word full in the FIFO 0000001 = 2 words, trigger is asserted when at least two words full in the FIFO..... 1111111 = 128 words, trigger is asserted when 128 words full in the FIFO.
1	0x0	**TX_FIFO_FLUSH**: 1 = flush the SLAVE TX FIFO; bit will be auto-cleared by Hardware after FIFO flush is done **0** = UNSET **1** = SET
0	0x0	**RX_FIFO_FLUSH**: 1 = flush the SLAVE RX FIFO; bit will be auto-cleared by Hardware after FIFO flush is done **0** = UNSET **1** = SET

I2C_SLV_FIFO_STATUS_0

Offset: 0xc4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00800000 (0bxxxx,xxx0,1000,0000,xxxx,xxxx,0000,0000)

Bit	Reset	Description
24	0x0	**SLV_XFER_ERR_REASON**: This bit describes the nature of the packet transfer error. It is meaningful only if PKT_XFER_ERR is set. 0 = Master terminated transaction before it was completed, 1 = Master did not terminate transaction when all bytes are transferred
23:16	0x80	**TX_FIFO_EMPTY_CNT**: The number of slots that can be written to the Slave Tx FIFO 00000000 = tx_fifo full 00000001 = 1 slot empty 00000010 = 2 slots empty.... 10000000 = 128 slots empty
7:0	0x0	**RX_FIFO_FULL_CNT**: The number of slots to be read from the Slave Rx FIFO 00000000 = rx_fifo empty 00000001 = 1 slot full 00000010 = 2 slots full .... 10000000 = 128 slots full ....

I2C_I2C_MST_NM_INT_STATUS_0

I2C Master Controller, Normal Mode Status register
Offset: 0xc8
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**TXN_DONE**: Status bit for Master controller transactions done in Normal mode. Sticky bit. If set, write 1 to clear it to zero. **0** = UNSET **1** = SET

I2C_I2C_MST_NM_INT_MASK_0

I2C Master Controller, Normal Mode Interrupt Mask register
Offset: 0xcc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx1)

Bit	Reset	Description
0	0x1	**TXN_DONE_INT_EN**: Interrupt enable bit for Master controller transactions done in Normal mode. **0** = DISABLE **1** = ENABLE

I2C_BUS_STATUS_0

This is a ready only register which gives the status of I2C interface
Offset: 0xd0
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000X (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
2	X	**SCL**: Gives the status of SCL (Serial Clock Line) of I^2C bus
1	X	**SDA**: Gives the status of SDA (Serial Data Line) of I^2C bus
0	X	**BUS_BUSY**: I2C Bus Busy status **0** = FREE **1** = BUSY

I2C_I2C_INTERFACE_TIMING_2_0

Register for Standard/Fast/Fm+ mode timing

Offset: 0xd4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**THD_DAT**: Data hold time in SM/FM/Fm+ speed modes

I2C_I2C_HS_INTERFACE_TIMING_2_0

I2C interface timing register for HS mode transfers

Offset: 0xd8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**HS_THD_DAT**: Data hold time in HS mode

I2C_I2C_MSTR_DATA_CAPTURE_TIMING_0

Register for I2C Master controller data capturing time.
Offset: 0xdc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0001)

Bit	Reset	Description
15:0	0x1	**TCAPTURE_DAT**: Used to tune the Master Controller Data Capture Period. Shifts the sampling point of SDA line by (Tcpature_dat+1) i2c_clk cycles from SCL rising edge. 0 = This field is not used, SDA line is sampled on SCL rising edge. Program to a value that is less than 3 or I2C_CLK_DIVISOR_STD_FAST_MODE or I2C_CLK_DIVISOR_HSMODE whichever is lower based on speed mode

I2C_I2C_SLV_DATA_CAPTURE_TIMING_0
Register for I2C Slave controller data capturing time.
Offset: 0xe0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000002 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0010)

Bit	Reset	Description
15:0	0x2	**TCAPTURE_DAT**: Used to tune the Slave Controller Data Capture Period. Shifts the sampling point of SDA line by (Tcpature_dat+1) i2c_clk cycles from SCL rising edge. 0 = This field is not used, SDA line is sampled on SCL rising edge. Program to a value that is less than half (1/2) of SCL High time in number of i2c_clk cycles.

9.7 Universal Asynchronous Receiver/Transmitter (UART)

9.7.1 Overview

There are seven Universal Asynchronous Receiver/Transmitters (UARTs) built into the System-on-Chip (SoC). Six of them are in the LSIO cluster in the SoC power domain, and the other instance is in ULP in the AON power domain. These UARTs support both 16450- and 16550-compatible modes and all are identical in feature set.

UARTs support full-duplex serial data transmission, performing synchronization and parallel-to-serial and serial-to-parallel data conversion. Synchronization is accomplished by adding start and stop bits to transmit data to form a data character. Data integrity may be accomplished by attaching an optional parity bit to the data character. The parity bit can be checked by the receiver for any transmission bit errors.

The interface supports word lengths from five to eight bits, an optional parity bit, and one or two stop bits. If enabled, parity can be odd, even, or forced to a defined state. Interrupts can be generated from any of 10 sources.

The UART controller supports both 16450 and 16550 compatible modes. The default mode is 16450 (non-FIFO mode). FIFO mode provides two independent FIFOs for transmit and receive and is selected by the FIFO control register. The controller also includes a 16-bit programmable baud rate generator and an 8-bit scratch register, up to six "modem" control lines, and two internal DMA handshake lines that are used to indicate when the FIFOs are ready to transfer data to the CPU.

The UARTs support a device clock of up to 204 MHz. Each symbol requires 16 clock cycles for proper sampling and processing of the input data stream. Thus, the maximum baud rate is 204/16 = 12.75 Mbps. Because 1 symbol = 1 bit, the data rate is 12.75 Mbps.

9.7.1.1 Features

The features supported by the UARTs are as follows:

- Synchronization for the serial data stream with start and stop bits to transmit data to and from a data character
- Data integrity by attaching a parity bit to the data character
- Support for word lengths from 5 to 8 bits, with an optional parity bit, and one or two stop bits
- Support for both 16450- and 16550-compatible modes. The default mode is 16450.
- Baud tolerance of ±2.7% (10 bits per character)
- DMA capable for both Tx and Rx
- 8-bit x 36-deep Tx FIFO
- 11-bit x 36-deep Rx FIFO
- 3 bits of 11 bits per entry logs the Rx errors in FIFO mode (break, framing, and parity errors as bits 10, 9, 8 of the FIFO entry)
- Auto sense baud detection
- Timeout interrupts to indicate if the incoming stream stopped
- Flow control support using RTS and CTS (hardware/software controlled)
- Internal loopback
- SIR encoding/decoding (3/16 or 4/16 baud pulse widths to transmit bit 0)

9.7.1.2 Security Features

There are no secure control or TrustZone protected registers in the UART register specification. Access may however be controlled from the Control Back-Bone of the SoC.

9.7.1.3 Hardware Signaling

All UARTs are implemented by a hardware block that supports modem control signals. For all UARTs, the DSRn and DCDn input signals are tied high (active); the RIn input signal is tied low; and the DTRn output signal is not connected.

Table 9.30 Serial Bus Interface Signals

Signal Name	Description
Outputs	

Signal Name	Description
TXD	Transmit Data Port
RTS	Request to Send
DTRn	Data Terminal Ready
Inputs	
RXD	Receiver Data Port
CTS	Clear to Send
DSRn	Data Set Ready
DCDn	Data Carrier Detect
RIn	Ring Indicator

Note: The DSR, DCD, and RI modem control signals do not control any UART hardware other than generating interrupts and status on change.

9.7.1.4 Clocking

The UART controller requires three clock sources:

1. APB clock: Maximum frequency is 408 MHz
2. clk_24M: Maximum device/Interface clock: 102 MHz clock
3. baud_clk: Maximum frequency of this clock is 204 MHz

9.7.1.4.1 Clock Sources

Table 9.31 Clk_24M Clock Sources

UARTA/UARTB/UARTD/UARTE/UARTF/UARTH	PLLP_OUT0, CLK_S, CLK_M
UARTC	PLLP_OUT0, PLLC_OUT0, pllAON_out, OSC_UNDIV, CLK_S

Clock Divisors

UART_FST_MIPI_CAL_CLK_DIVISOR/AON_UART_FST_MIPI_CAL_CLK_DIVISOR: X = Divide by (N/2+1)

- To achieve the required fast clock for UART controller, the following registers should be programmed:
  - CLK_RST_CONTROLLER_CLK_SOURCE_UART_FST_MIPI_CAL_0
  - CLK_RST_CONTROLLER_CLK_SOURCE_AON_UART_FST_MIPI_CAL_0.

The source of the clock can be chosen by programming [31:29] of this register and the divisor by programming [7:0]. Divisor to be programmed in [7:0] N = ((X-1)*2)

Table 9.32 Baud_clk Clock Sources

UARTA/UARTB/UARTD/UARTE/UARTF/UARTH	PLLP_OUT0, CLK_S, CLK_M
UARTC	PLLP_OUT0, PLLC_OUT0, pllAON_out, PLLC4_muxed (controlled by PLLC4_CLK_SEL), CLK_S, OSC_UNDIV

Clock Divisor Options

UART dll/dlm registers work as clk divisor as in the clock registers.

To achieve the required baud rate for a given UART controller, the CLK_RST_CONTROLLER_CLK_SOURCE_UART*_0 register should be programmed. The source of the clock can be chosen by programming [31:29] of this register and divisor by programming [15:0].

To achieve a divisor value of 'X', the register should be configured with value N=((X-1)*2).

Programming bit 24 determines if the programmed [15:0] divisor should be used or the DLM/DLL value from UART. Only one of them can be used at a time.

For example: Source clock: CLK_M -> crystal clock of 38.4 MHz

- To achieve baud rate of 115200, baud_clock needed = 1.8432 MHz
- To achieve 1.8432 MHz with 38.4 MHz crystal clock, divisor needed is (38.4/1.8432) = 20.833333
- Since CAR registers provide divisors with granularity of 0.5, the nearest divisor can be either 20.5 or 21.
- For divisors of 20.5, 21 baud frequencies that can be achieved are (38.4/20.5) = 1.873 MHz and (38.4/21) = 1.8285 MHz, respectively. Based on the error %, the nearest baud rate that can be achieved is 1.8285 MHz
- For divisor of 21, value to be programmed in [15:0] is 0x28, [31:29] = 0x7

The UART supports baud tolerance of +/-4% (10 bits per character).

The table below summarizes the divisor to achieve a 115200 baud rate with different source frequencies.

Table 9.33 Divisor for Baud Rate

Osc Clk Frequency (M)	For baud rate 115200, divisor needed D = (M *1000*1000) / (115200 *16)	Value of divisor (DR) = N/ 2+1	UARTA_CLK_DIVISOR (N)
12	6.51	6.5	11 (=0xb)

Osc Clk Frequency (M)	For baud rate 115200, divisor needed D = (M *1000*1000) / (115200 *16)	Value of divisor (DR) = N/ 2+1	UARTA_CLK_DIVISOR (N)
13	7.053	7.0	12 (=0xc)
19.2	10.417	10.5	19 (=0x13)
26	14.106	14.0	26 (=0x1a)
38.4	20.833	21.0	40 (=0x28)
48	26.042	26.0	50 (=0x32)

9.7.1.5 Use Cases

Potential use cases are listed in the following table.

Table 9.34 UART Use Cases

Interface	Use Case	ULP/AON	LSIO	UART Pins on Interface	I/O Rail to Power Up the Instance
UART1/UARTA	GPS		Yes	TXD/RXD/CTS/RTS	VDDIO_G3
UART2/UARTB	First Bluetooth (for driver)		Yes	TXD/RXD/CTS/RTS	VDDIO_G2
UART3/UARTC	Primary Debug (2-pin only)/ Boot UART – open-box debug solution.	Yes		TXD/RXD	VDDIO_AO
UART4/UARTD	Second Bluetooth (for passenger)		Yes	TXD/RXD/CTS/RTS	VDDIO_G4
UART5/UARTE	Satellite Radio (Sirius XM)		Yes	TXD/RXD/CTS/RTS	VDDIO_G2
UART6/UARTF	UART debug over DP-AUX0 (2-pin only) – closed-box debug solution		Yes	TXD/RXD	VDDIO_EDP1
UART8/UARTH	UART debug over USB2 (2-pin only) – closed box debug solution		Yes	TXD/RXD	AVDD_USB

Typically, UART use cases should enable the flow control mechanism to allow for smooth data transfers. Without flow control, data overflow/underflow may be seen due to changes in system load conditions. For example, in the debug console use case, lack of flow control may result in erroneous print on the console. For other use cases, not enabling flow controller can functionally disrupt the behavior.

9.7.1.6 Error Handling

The UART has a APB slave error port to generate a slave error (APB4.0 specification compliant). The APB slave error is generated by the UART if there is access to a register that does not exist (address hole) or if a write request is issued to a read-only register. If there are errors in the Rx data, there is no way currently to associate error information with a particular character unless post-processing is done on the characters received, which negates the benefit of DMA. The software will re-transfer a chunk if needed. This makes the controller incompatible with the standard POSIX I/O error handling.

9.7.2 Programming Guidelines

Below are typical groups of registers required in a peripheral initialization sequence/programmer model. All GROUP1 registers must be configured before the IP registers in GROUP2 are configured.

GROUP1

- CAR registers
- Pad Control registers
- Pinmux registers
- GPC DMA registers
- Interrupt Control registers

GROUP2

- IP registers

Pad control and pinmux registers are typically done once during the boot and they affect runtime performance. These registers must be properly programmed during SC7 exit context recovery. In this case, you must enable strict, for example. If boot ROM changes the pad controls to enable the boot media interface (to avoid a daisy situation due to ordering). Similarly, with SC7 exit the complete pinmux recovery should occur before the drivers start accessing the interface. The security ordering (SO) setting is part of the CACHEABLE[3:0] bits on the AXI interface and should be SET to enforce SO of the downlink APB clients. If those bits are zero, then a transaction is SO and the bridge waits for the write response from the slave. Use the BRIDGE_EARLY_ACK_DISABLE bit for debugging.

9.7.2.1 UART Register Access

UART uses an APB interface for register access with a 64 KB AMAP. The following are address offsets for different UART controllers.

Table 9.35 Address Offsets for UART Controllers

Controller	Offsets
UART1/UARTA (LSIO)	0000:0000 0000:ffff
UART2/UARTB (LSIO)	0001:0000 0001:ffff
UART3/UARTC (AON)	0028:0000 0028:ffff
UART4/UARTD (LSIO)	0003:0000 0003:ffff
UART5/UARTE (LSIO)	0004:0000 0004:ffff
UART6/UARTF (LSIO)	0005:0000 0005:ffff
UART8/UARTH (LSIO)	0007:0000 0007:ffff

Typical APB delay (request seen at UART controller interface to response) is two APB cycles.

9.7.2.2 16450 Mode Programming

1. Program pinmux settings to select a UART.
2. Enable the UART clocks.
3. Apply module reset.
4. If internal loopback is needed, program MCR[4] to 1.
5. Program FCR[0] to 0.
6. Enable interrupts in the IER register as needed.
7. Write data into the THR register.
8. Wait for a THR interrupt, if enabled, or poll for LSR[5].
9. During a receive, wait for an RDR interrupt or poll for LSR[0]/LSR[9].
10. Read the UART.LSR register to clear interrupts.

9.7.2.3 16550 Mode Programming

1. Program pinmux settings to select a UART.
2. Enable the UART clocks.
3. Apply module reset.
4. For enabling internal loopback, program MCR[4] to 1.
5. Program FCR[0] to 1.
6. Poll IIR[6] to see if FIFO_MODE is really enabled.
7. Program trigger levels as required.
8. Enable interrupts in the IER register as needed.
9. Write data into the THR register.
10. Wait for a THR interrupt, if enabled, or poll for LSR[5].

11. During a receive, wait for an RDR interrupt or poll for LSR[0]/LSR[9].
12. Read the UART.LSR register to clear interrupts.
13. GPC DMA requester numbers for UARTs A/1, B/2, C/3, D/4, E/5, F/6, G/7, and H/8 are 8, 9, 3, 19, 20, 12, 2, and 13, respectively.

Interface	DMA Flow Control Mapping
UART1	GPC DMA + CBB's AXI2APB Bridge
UART2	GPC DMA + CBB's AXI2APB Bridge
UART3	AON Cluster DMA, BPMP DMA, and GPC DMA + CBB's AXI2APB Bridge
UART4	GPC DMA + CBB's AXI2APB Bridge
UART5	GPC DMA + CBB's AXI2APB Bridge
UART6	AON Cluster DMA and GPC DMA + CBB's AXI2APB Bridge
UART8	AON Cluster DMA and GPC DMA + CBB's AXI2APB Bridge

9.7.2.3.1 UART Instances to GPCDMA Mapping

LSIO instances:

- UART1/A – 8
- UART2/B – 9
- UART4/D – 19
- UART5/E – 20
- UART6/F – 12
- UART8/H – 13

AON UART instances:

- UART3/C – 3

9.7.2.4 Transmitter and Receiver Holding Registers

The serial transmitter section consists of a Transmit Hold Register (THR) and Transmit Shift Register (TSR). The status of transmit hold register is provided in the Line Status Register (LSR). Writing to this register (THR) transfers the contents of data bus (D [7:0]) to the Transmit Holding Register whenever the Transmit Holding Register or Transmit Shift Register is empty. The Transmit Holding Register empty flag is set to 1 when the transmitter is empty or data is transferred to the Transmit Shift Register. Note that a write operation is performed when the Transmit Holding Register empty flag is set.

The serial receiver section also contains an 8-bit Receiver Holding/Buffer Register (RBR). Receive data is removed from the UART and received by the processor by reading the RBR.

The receiver contains a mechanism for preventing false starts as follows: On the falling edge of the start bit, the receiver internal counter starts to count 7.5 clocks (16x clock), which is the center of the start bit. If the input goes high before counter reaches mid-point sampling, RX_FSM is moved back to the IDLE state and that is treated as a start bit glitch. Verifying the start bit prevents the receiver from assembling a false data character due to a low going noise spike on the Rx input. Receiver status codes are posted in the Line Status Register.

9.7.2.5 FIFO Interrupt Mode Operation

When the receive FIFO (FCR bit 0 = 1) and receive interrupts (IER bit 0 = 1) are enabled, a receiver interrupt occurs as follows:

- The receive data available interrupts are issued to the CPU when the FIFO has reached its programmed trigger level; it is cleared as soon as the FIFO drops below its programmed trigger level.
- The ISR receive data available indication also occurs when the FIFO trigger level is reached, and like the interrupt, it is cleared when the FIFO drops below the trigger level.
- The data ready bit (LSR bit 0) is set as soon as a character is transferred from the shift register to the receiver FIFO. It is reset when the FIFO is empty.

9.7.2.6 FIFO Polled Mode Operation

When FCR bit 0 = 1; clearing IER bits [3:0] to zero puts the UART in the FIFO polled mode of operation. Since the receiver and transmitter are controlled separately, either one or both can be in the polled mode operation by using the line status register as follows:

- LSR bit 0 is set when the number of entries is greater than or equal to trigger level programmed
- LST bits [4:1] specifies which error(s) has occurred
- LSR bit 5 indicates when the transmit FIFO is empty
- LSR bit 6 indicates when both transmit FIFO and transmit shift registers are empty
- LSR bit 7 indicates when there are any errors in the receive FIFO
- LSR bit 8 indicates when the transmit FIFO is full
- LSR bit 9 indicates Rx FIFO empty status

The UART requires a two-step FIFO enable operation in order to enable receive trigger levels.

9.7.2.7 Programmable Baud Rate Generator

The UART contains a programmable baud rate generator that is capable of taking the UART clock input and dividing it by any divisor from 1 to 2^{16}-1 The UART clocks are defined in the Clocking chapter. The output frequency of baudout is equal to 16X the transmission baud rate (Baudout=16 X baud rate). Customized baud rates are achieved by selecting proper divisor values for the MSB and LSB bits of the baud rate generator.

9.7.2.8 Enable Register

There is an Interrupt Enable Register (IER) for each UART. The Interrupt Enable Register(s) masks the incoming interrupts from the receiver ready, transmitter empty, and line status.

9.7.2.9 Interrupt Identification Register

The UART provides four level prioritized interrupt conditions to minimize software overhead during data character transfers. The Interrupt Identification Register (IIR) provides the source of the interrupt in a prioritized manner.

During the read cycle, the 16550 provides the highest interrupt level to be serviced by the CPU. No other interrupts are acknowledged until the particular interrupt is serviced. The prioritized interrupt levels are shown in the table below. The Receive Data Time-out mode is enabled when the UART is operating in the FIFO mode.

Receive time-out does not occur if the receive FIFO is empty. The time-out counter is reset at the center of each stop bit received or each time the receive holding register is read. The actual time out value is:

- T (Time out length in bits) = 4xP (Programmed work length) + 12

To convert the time-out value to a character value, divide this number to its complete word length + parity (if used) + number of stop bits and start bit.

Example

If you program the word length = 7, no parity, and one stop bit, the time-out is:

- T = 4x7 (programmed word length) + 12 = 40 bits.
- Character time = 40/9
- (Programmed word length = 7) + (stop bit = 1) + (start bit = 1) =4.4 characters

Table 9.36 Prioritized Interrupt Levels

Priority	D3	D2	D1	D0	Interrupt Source Description
1	0	1	1	0	LSR (Receiver Line Status Register)
2	0	1	0	0	RXRDY (Received Data Ready)
2	1	1	0	0	RXRDY (Received Data Timeout)
3	0	0	1	0	TXRDY (Transmitter Holding Register)
4	0	0	0	0	MSR (Modem Status Register)

9.7.2.10 FIFO Control Register

9.7.2.10.1 FIFO Control Register Modes

The FIFO control register (FCR) is used to enable the FIFOs, clear the FIFOs, set the receiver FIFO trigger level, and select the type of DMA signaling.

9.7.2.10.2 FIFO Mode (CR bit 0 = 1)

FCR bit 0 should be programmed to '1' to operate the UART controller in FIFO (16450) mode. When FIFO mode is enabled, allow minimum of three baud cycles duration before writing TX_DATA into the THR_DLAB register.

Design requirement: It takes three baud cycles for the fifo_mode enable signal to synchronize to the baud_clk domain and ~fifo_mode_bclk is also a flush condition for Tx and Rx FIFOs. When FIFO mode is enabled through a register write in the pclk domain and if there is not enough delay before servicing THRE interrupt/NTXRDY requests, the written data is lost. To ensure proper data transfer, after fifo_mode is enabled for first time through register write, give a delay of minimum of three baud cycles before writing TX_DATA into the THR register.

The operation of the FCR in the four DMA modes is given below.

9.7.2.10.3 Transmit Operation in DMA Mode 0 or Mode 1

When the UART is in the 16450 mode (FCR bit 0 = 0) or in the FIFO mode (FCR bit 0 = 1, FCR bit 3 = 0) and when there are no characters in the transmit FIFO or Transmit Holding Register, the TXRDY* pin goes low. Once active, the TXRDY* pin goes high (inactive) after the first character is loaded into the Transmit Holding Register.

9.7.2.10.4 Receive Operation in DMA Mode 0

When the UART is in 16450 mode (FCR bit 0 = 0) or in the FIFO mode (FCR bit 0 = 1, FCR bit 3 = 0) and there is at least 1 character in the receive FIFO, the RXRDY* pin goes low. Once active, the RXRDY* pin goes high (inactive) when there are no more characters in the receiver.

9.7.2.10.5 Receive Operation in DMA Mode 1

When the UART is in the FIFO mode (FCR bit 0 = 1, FCR bit 3 = 1) and the trigger level or the timeout has been reached, the RXRDY* pin goes low. Once it is activated, it goes high (inactive) when there are no more characters in the FIFO.

9.7.2.11 Line Control Register

The Line Control Register (LCR) is used to specify the asynchronous data communication format. The word length, stop bits, and parity can be selected by writing appropriate bits in this register.

9.7.2.12 Modem Control Register

The Modem Control Register (MCR) controls the interface with the modem or a peripheral device (RS232).

9.7.2.12.1 Loopback Mode

If MCR[4] = 1, the loopback mode is enabled, and the following occurs: The receiver input (Rx), CTS, DSR, CD, and RI are disabled. Internally, the transmitter output is connected to the receiver input and DTR, RTS, OP1, and OP2 are connected to modem control inputs. In this mode, the receiver and transmitter interrupts are fully operational, but the interrupt sources are now the lower four bits of the Modem Control Register instead of the four modem control inputs. The interrupts are controlled by the IER register.

9.7.2.13 Line Status Register

The Line Status Register (LSR) provides the status of data transfer to the CPU.

9.7.2.14 Modem Status Register

The Modem Status Register (MSR) provides the current state of the control lines from the modem or peripheral to the CPU. Four bits of this register are used to indicate the changed information. These bits are set to "1" whenever a control input from the modem changes state. They are set to "0" whenever the CPU reads the register.

9.7.2.15 Scratchpad Register

Eight bits of information can be stored in the Scratchpad Register (SR). Information in this register does not affect the operation of the device in any way.

9.7.2.16 Auto-Baud Sense Register

The UART can automatically determine the correct baud divisor by using the auto-baud sense register (UART offsets 0x3C). The most significant bit of this register is the valid flag. A write to this register clears the valid flag and enable the auto-baud process. When the first Rx edge occurs, a counter running at 24 MHz starts counting. When another rx_edge occurs, the "complete" flag is set, the value is frozen, and the auto-baud sense value register is updated with the count value. The low 20 bits of the Auto-Baud Sense Register (ASR) give the number of clocks within a single bit. Because the UART uses 16x oversampling, the resulting value needs to be adjusted by shifting right 4 bits, then loading the resulting count in the divisor latch of the UART. (In the code snippet below, the lower 4 bits are rounded to give slightly greater accuracy.)

Because the speed determination is made by measuring the start bit, special characters must be sent by the transmitting UART to guarantee that the next character after the start bit is a 1. Since bit 0 (rightmost bit) is sent first, the ASCII Carriage Return character (CR) is sufficient to enable proper speed sense.

The following code snippet returns the values for DLH and DLL in r9, r8:

```
MOV  r0, #1  ; dummy write data
 STRB  r0, \[r3, #U_ASR\]; Start autobaud sense (r3 as uart_base)
; now poll the autobaud sense register MSB until Valid is true
wait4valid
  LDR  r2, \[r3, #U_ASR\]; Read ASR
  TST  r2, #0x80000000 ; the Valid bit (active high)
  BEQ  wait4valid
; autobaud sense check complete...
; r2 as number of 1x clocks in one bit time
; representing the number of 24MHz clocks in the start bit
; Since this represents 16 of the baud (16x) clocks, we
; will be dividing by 16 to get baud divisor, but first
; round to nearest by adding 8 before the divide:
  ADD  r4, r2, #8 ; add 1/2 resolution
  MOV  r6, r4, LSR #4 ; divide by 16... R6 will have total divisor
  AND  r8, r6, #0xFF  ; copy DLL to r8
  MOV  r9, r6, LSR #8 ; copy DLM to r9
  AND  r9, r9, #0xff ; mask upper bytes
```

9.7.2.17 Baud Rate Generator

The following table is a divisor table for the baud rate, assuming the oscillator is 24.000 MHz.

Divisor = (osc clock/(baud rate*16))*10^6

Table 9.37 Baud Rate Generator Programming

Baud Rate	Divisor
300	5000
1200	1250
2400	625
4800	312
9600	156
19.2K	78
38.4K	39
57.6K	26
115.2K	13
460.8K	3

9.7.2.18 SIR Pulse Encoder/Decoder

The UART transmit (Tx) data is passed through the SIR Pulse Encoder/Decoder module prior to being muxed out to the IR transmitter. This module converts transmitted zeros into a 3/16 Return-To-Zero (RZ) pulse. Similarly, received (Rx) data is bit-synchronized using the 16x baud clock and the original serial data recovered from the IR bit stream. This data is then sent to the UART for reception.

The signal generated in SIR is as follows:

- On logic '1' the LED is off.
- On logic '0' a pulse is created starting the center of the bit time and lasting 3/16 of bit time period or 4/16 of bit time period depending on the current settings.

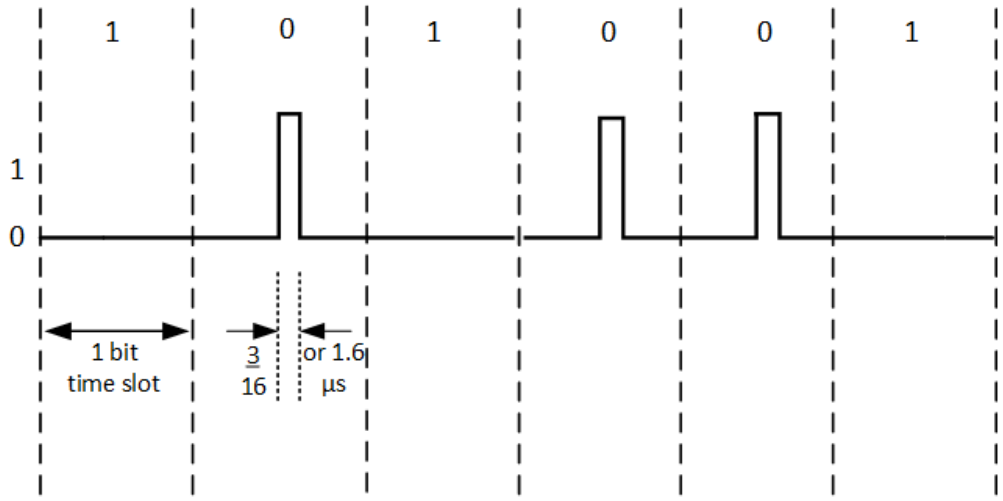
The figure below displays the output for the input 101001 (in sending order) in SIR encoding.

If SIR encoding/decoding has to be used in loopback mode, the following programming guideline/sequence is required:

1. Program SIR enable

2. Wait for a minimum of (4*baudclk)+ (4clk24m)
3. Program loopback enable

Figure 9.28 Output in Sending Order in SIR Encoding



9.7.2.19 UART Interrupts and Interrupt Service Routine

There are multiple conditions on which a UART interrupt is generated. The following are interrupts available for the UART controller. Each of the interrupts is routed to an interrupt controller, if the corresponding Interrupt enable bit in the IER register is set.

1. EORD: This indicates an end of received data interrupt. This interrupt is set when UART Rx detects there is no data being received for more than four character times. This is useful to determine that the transmitter is done with sending all its data. In FIFO mode, this interrupt is not generated if data is stopped because of handshaking.
2. RXT: RX_TIMEOUT interrupt occurs when data has been sitting in RX_FIFO for more than four character times due to some reason. One reason could be because there is not enough data to reach the trigger level. This interrupt needs to be handled by software to have data reception in a timely manner.
3. RXS: This interrupt indicates there are errors detected in the received data. The error could be break error/overrun error/parity error or framing error. A Break error occurs when a break condition (all zeros) has been detected on an Rx line. An Overrun error indicates a new character reception from an external device when the Rx FIFO is already full. Parity error occurs because of parity mismatch and framing error occurs when stop bit of a character received is '0' instead of '1'.
4. Handling framing errors:
   When a framing error is detected at UART Rx, the reference point for further sampling is lost. UART Rx needs the line to be continuous HIGH for at least one character time to recover the sampling point. All subsequent data after framing error detection needs to be discarded by software (clear IER[0] before flushing the FIFO to avoid any spurious interrupt).

Retransmission is required on framing error cases. This needs some protocol at a higher level than the UART. When there is framing error seen by software, the UART line needs to be maintained HIGH for at least one character time to let Rx recover. This has to be completely handled by software.

For a 4-pin UART, RTS can be used to stop further reception of data until Rx recovers.

5. Force RTS to let Tx stop sending further data
6. Rx given enough time to recover
7. Deassert RTS
8. Start receiving further data
   a. THR: This interrupt indicates the Transmit Holding Register is empty. In FIFO mode, this is set when there is enough space in the FIFO based on trigger level setting so that software can start writing Tx data into the THR register.
   b. RDR: Receive data ready interrupt is asserted when number of entries in RX_FIFO has reached the trigger level programmed when in FIFO mode. This bit is an indicator to start reading the contents of RX_FIFO. While servicing RDR, disable the RXT interrupt and enable it after RDR is serviced.

When uart_intr is asserted, software reads the IIR register to know which interrupt occurred. The interrupts in the UART are prioritized and hence IIR always shows the current highest priority interrupt. If more than one interrupt is pending to be serviced, the one with the highest priority is shown in IIR. When all interrupts have been serviced, IIR shows no_interrupt. The interrupt priority is defined per the encoding below (the priorities are listed highest to lowest):

IIR[3:0] priority level:

- 0001: no interrupt pending
- 0110: Overrun Error, Parity Error, Framing Error, Break
- 0100: Receiver Data Available
- 1100: rx_timeout_intr
- 1000: eord_timeout_intr
- 0010: Transmitter Holding Register empty
- 0000: modem_status interrupt

The interrupt service routine (ISR) for UART interrupts is not same for all and varies depending on type of interrupt. This section summarizes the ISR expected for clearing each of the interrupt condition.

- RXS interrupt: LSR read clears this interrupt. RXS interrupt conditions include parity/break/overrun or framing error. Parity/break/framing errors also clear on Rx FIFO pop operation.
- RDR/RX_TIMEOUT interrupt: These interrupts are cleared on reading the Read buffer register
- EORD interrupt: IER[5] needs to be disabled to clear this interrupt. The bit can be re-enabled again to receive further interrupts.

- THR interrupt: IIR read or write to buffer data register clears this interrupt. The interrupt keeps occurring until writes are performed to buffer the write register to fill the FIFO and reach the trigger level.
- Disabling an interrupt_enable bit for any of the following interrupts lead to clearing of uart_intr.

Bit	Reset	Description
3	0x0	IE_MSI: Interrupt Enable for Modem Status Interrupt 0 = DISABLE 1 = ENABLE
2	0x0	IE_RXS: Interrupt Enable for Receiver Line Status Interrupt. 0 = DISABLE 1 = ENABLE
1	0x0	IE_THR: Interrupt Enable for Transmitter Holding Register Empty interrupt 0 = DISABLE 1 = ENABLE
0	0x0	IE_RHR: Interrupt Enable for Received Data Interrupt 0 = DISABLE 1 = ENABLE

Notes:

1. Interrupt enable for RXS interrupt (IE_RXS) should be disabled whenever LSR is polled/read to avoid spurious interrupts because of errors detected in receive data. LSR read is an interrupt clearing mechanism for RXS errors.
2. In the UART controller, status registers are also masked with corresponding IE bits unlike conventional controller's implementation.
3. Disable the RXT interrupt before issuing an Rx FIFO flush request. Enable RXT after the flush is done.

Interrupt mapping of different UART instances is listed in the table below.

Interface	Interrupt Mapping
UART1	LIC
UART2	LIC
UART3	LIC, BPMP, SPE
UART4	LIC
UART5	LIC
UART6	LIC, SPE
UART8	LIC, SPE

9.7.2.20 FIFO Flushing Guidelines

The UART controller provides TX_CLR and RX_CLR bits to flush the FIFOs. The FIFOs can be flushed independently or simultaneously by following the programming guidelines below:

9.7.2.20.1 Non-loopback Node

Simultaneous flush (TX_FLUSH and RX_FLUSH in single register write)

1. Make sure there is no new data being written to TX_buffer.
2. Program rts_enable to 0 (MCR[6]).
3. Make sure MCR[1] is 0.
4. Wait for 1 character time.
5. Issue flush requests - Program TX_CLR to 1 and RX_CLR to 1 (UART_IIR_FCR_0[2], UART_IIR_FCR_0[1]).
6. Poll for TMTY to be '1' (UART_LSR_0[6]).
7. Poll for RDR to be '0' (UART_LSR_0[0]).
8. Re-enable rts_enable MCR[6].
9. Perform new transfers.

9.7.2.20.2 Independent Flush Operations (Tx/Rx separately)

Tx flush:

1. Make sure there is no new data being written to TX_buffer.
2. Program TX_CLR to 1 (UART_IIR_FCR_0[2] - TX_FLUSH request).
3. Poll for TMTY to be '1' (UART_LSR_0[6]).
4. Perform new transfers.

Rx flush:

1. Program rts_enable to 0 (MCR[6]).
2. Make sure MCR[1] is 0.
3. Wait for 1 character time.
4. Program RX_CLR to 1 (UART_IIR_FCR_0[1] - RX_FLUSH request).
5. Poll for RDR to be '0' (UART_LSR_0[0]).
6. Re-enable rts_enable MCR[6].
7. Perform new transfers.

9.7.2.20.3 Loopback Mode

Simultaneous flush operations do not work (TX_FLUSH and RX_FLUSH in a single register write). The only option is to disable the FIFO mode bit.

Two independent flushes to be issued as mentioned below to flush both FIFOs:

1. Program rts_enable to 0 (MCR[6]).
2. Make sure MCR[1] is 0.
3. Program TX_CLR to 1 (UART_IIR_FCR_0[2] - TX_FLUSH request).
4. Poll for TMTY to be '1' (UART_LSR_0[6]).
5. Program RX_CLR to 1 (UART_IIR_FCR_0[1] - RX_FLUSH request).
6. Poll for RDR to be '0' (UART_LSR_0[0]).
7. Re-enable rts_enable MCR[6].
8. Perform new transfers.

9.7.2.21 UART Boot

UART3 is used for boot. The access path is as follows:

- BPMP-R5 → BPMP-NOC → CBB-NOC → AON-NOC → UART

Clock configuration that can be used for UARTC fast boot (@192 MHz baud clock = 12 MHz baud rate) as follows.

Clock source options available for UARTC baud clock are:

- PLLP_OUT0, PLLC_OUT0, pllAON_out, PLLC4_muxed (controlled by PLLC4_CLK_SEL), CLK_S, CLK_M

To achieve a baud rate as high as 12 MHz, PLLC4 is the right source.

PLLC4 has three output options, and you can choose PLLC4_OUT2 by programming CLK_RST_CONTROLLER_PLLC4_MISC1_0[16:15] = 0x1 (see the Clock Controller chapter for the details of the CLK_RST_CONTROLLER_PLLC4_MISC1_0 register).

With an oscillator frequency at 38.4 MHz, a 192 MHz baud clock can be achieved using PLLC4 option as follows:

- Crystal frequency = 38.4 MHz, N = 50, M = 2
- PLLC4_OUT0 = 38.4*(50/2) = 960 MHz (Normal Frequency mode)
- PLLC4_OUT2 = VCO/DIV5 = 960/5 = 192 MHz

Choosing divisor = 1 (CLK_RST_CONTROLLER_CLK_SOURCE_UARTC_0.UARTC_CLK_DIVISOR = 0), baud frequency = 192/1 = 192 MHz can be achieved.

- Set CLK_RST_CONTROLLER_PLLC4_MISC1_0. PLLC4_CLK_SEL[16:15] = 0x1 [To select PLLC4_OUT2 which is PLLC4-VCO/5 as the PLLC4_MUXED clock selection]
- Set CLK_RST_CONTROLLER_CLK_SOURCE_UARTC_0. UARTC_CLK_SRC[31:29] = 0x1 [To select PLLC4_MUXED]
- Set CLK_RST_CONTROLLER_CLK_SOURCE_UARTC_0.UARTC_CLK_DIVISOR = 0x0 [To select the UART final post-divider value '1,' i.e., the equation is PLL_C4-VCO/5]
- Set CLK_RST_CONTROLLER_CLK_SOURCE_UARTC_0.UARTC_DIV_ENB=0x1 [To select UART divisor programmed in CAR register]

Notes:

1. With PLLC4_OUT0 = 998.4 MHz, PLLC4_OUT1 = 332.8 MHz, PLLC4_OUT2=199.68 – normal EMMC mode selection.
2. For Fast boot @ 192 MHz (12 MHz baud), nearest possible value that can be achieved using CLK_DIV=0. This gives nearest baud_clk of 199.68 MHz
3. Baud clock = PLLC4_OUT2/DIVISOR = (199.68/((0/2)+1)) = 199.68/1 = 199.68 MHz

Optimized PIO mode programming model to meet fast boot requirement @ 192 MHz baud clock is as below:

RDR generation in the SoC is dependent on trigger setting. So to avoid two reads for each register access as was being done in Boot ROM, use one RDR for every 16-byte read (assuming trigger level set is 16)

1. Set Rx trigger level to 16
2. Check for RDR
3. Issue 16 read data requests to UART
4. Go back to step 2
5. For any residue bytes which are a non-multiple of 16, read the residue bytes using the RX_FIFO_EMPTY register available in LSR register.

9.7.2.22 IRDA_CSR Register (UART_IRDA_CSR_0)

The IRDA_CSR register provides configuration bits to enable SIR mode, and pulse width and invert polarity options for CTS/RTS/TXD and RXD pins. Default polarity of TXD/RXD/CTS/RTS pins is active low and usage of invert polarity option reverses polarity to active high.

Use the below programming model for the invert_RXD option in case of 4-pin UART:

1. Get UART controller out of reset
2. Force RTS from the SoC so that data from external device is not sent out
3. Program invert_rxd=1 in the SoC
4. Wait for 1 character time (character sampling may be in progress in the SoC with false start bit interpreted)
5. Flush Rx FIFO

6. Release RTS
7. Start characters reception from Tx

For the no-flow control case:

1. Get UART controller out of reset
2. Program invert_rxd=1 in the SoC
3. Wait for 1 character time (character sampling may be in progress in the SoC with false start bit interpreted)
4. Flush Rx FIFO
5. Remove external device out of reset
6. Start characters reception from Tx

The same guidelines are applicable when switching the invert_rxd option through register configuration.

9.7.2.23 Pad and Pinmux Information

The UART uses active-low protocol by default (unless invert rts/cts/txd/rxd options are used). To facilitate sharing with other pads, some of the pins may have pulldown by default. For correct operation, the pins should be pulled up using pad control programming before removing the UART controller from reset.

9.7.2.23.1 UART1/UARTA

UARTA has one pinmux option as the primary interface on UART1 Tx/Rx/RTS/CTS pins.

PIN	REGISTER	PADCTL	I/O PAD
CTS	PADCTL_CAM_UART1_CTS_0	PADCTL_CAM_CFG2TMC_UART1_CTS_0	BDPGLP_JFCR90_VDVXP1P1P1
RTS	PADCTL_CAM_UART1_RTS_0	PADCTL_CAM_CFG2TMC_UART1_RTS_0	BDPGLP_JFCR90_VXVDP1P1P1
RXD	PADCTL_CAM_UART1_RX_0	PADCTL_CAM_CFG2TMC_UART1_RX_0	BDPGLPHVIN_EFCR90_VXVDP1P1P1
TXD	PADCTL_CAM_UART1_TX_0	PADCTL_CAM_CFG2TMC_UART1_TX_0	BDPGLPHVIN_EFCR90_VXVDP1P1P1

9.7.2.23.2 UART2/UARTB

UARTB has one pinmux option as the primary interface on UART2 Tx/Rx/RTS/CTS pins.

PIN	REGISTER	PADCTL	I/O PAD
CTS	PADCTL_UART_UART2_CTS_0	PADCTL_UART_CFG2TMC_UART2_CTS_0	BDPGLP_JFCR90_VDVXP1P1P1
RTS	PADCTL_UART_UART2_RTS_0	PADCTL_UART_CFG2TMC_UART2_RTS_0	BDPGLP_JFCR90_VXVDP1P1P1
RXD	PADCTL_UART_UART2_RX_0	PADCTL_UART_CFG2TMC_UART2_RX_0	BDPGLPHVIN_EFCR90_VXVDP1P1P1
TXD	PADCTL_UART_UART2_TX_0	PADCTL_UART_CFG2TMC_UART2_TX_0	BDPGLPHVIN_EFCR90_VXVDP1P1P1

9.7.2.23.3 UART3/UARTC

UARTC has one pinmux option as the primary interface on UART3 Tx/Rx pins.

PIN	REGISTER	PADCTL	I/O PAD
RXD	PADCTL_AO_UART3_RX_0	PADCTL_AO_CFG2TMC_UART3_RX_0	BDPGLPHVIN_EFCR90_VDP1P1P1
TXD	PADCTL_AO_UART3_TX_0	PADCTL_AO_CFG2TMC_UART3_TX_0	BDPGLPHVIN_EFCR90_VDP1P1P1

9.7.2.23.4 UART4/UARTD

UARTD has one pinmux option as the primary interface on UART4 Tx/Rx/RTS/CTS pins.

PIN	REGISTER	PADCTL	I/O PAD
CTS	PADCTL_CONN_UART4_CTS_0	PADCTL_CONN_CFG2TMC_UART4_CTS_0	BDPGLP_JFCR90_VDVXP1P1P1
RTS	PADCTL_CONN_UART4_RTS_0	PADCTL_CONN_CFG2TMC_UART4_RTS_0	BDPGLP_JFCR90_VXVDP1P1P1
RXD	PADCTL_CONN_UART4_RX_0	PADCTL_CONN_CFG2TMC_UART4_RX_0	BDPGLP_JFCR90_VXVDP1P1P1
TXD	PADCTL_CONN_UART4_TX_0	PADCTL_CONN_CFG2TMC_UART4_TX_0	BDPGLP_JFCR90_VXVDP1P1P1

9.7.2.23.5 UART5/UARTE

UARTE has one pinmux option as the primary interface on UART5 Tx/Rx/RTS/CTS pins.

PIN	REGISTER	PADCTL	I/O PAD
CTS	PADCTL_UART_UART5_CTS_0	PADCTL_UART_CFG2TMC_UART5_CTS_0	BDPGLP_JFCR90_VDVXP1P1P1

PIN	REGISTER	PADCTL	I/O PAD
RTS	PADCTL_UART_UART5_RTS_0	PADCTL_UART_CFG2TMC_UART5_RTS_0	BDPGLP_JFCR90_VXVDP1P1P1
RXD	PADCTL_UART_UART5_RX_0	PADCTL_UART_CFG2TMC_UART5_RX_0	BDPGLP_JFCR90_VXVDP1P1P1
TXD	PADCTL_UART_UART5_TX_0	PADCTL_UART_CFG2TMC_UART5_TX_0	BDPGLP_JFCR90_VXVDP1P1P1

9.7.2.23.6 UART6/UARTF

I/O pad: BDDPAUX_F3FCR90_VD181818NC

UART6 is muxed internally in LSIO with I2C6 that operates on DP-AUX-CH0. Mux Sel is configured in DPAUX_PINMUX_CFG_0 register. I2C6 would be used with DP Aux port and pinmux doesn't exist for this. I^2C vs Aux channel mode selection happens from a register bit field in DP registers. Interface has DP_AUX_CH0_P and DP_AUX_CH0_N pins. UART6 would use I^2C mode of the DPAUX pad.

DPAUX_HYBRID_PADCTL_0

I/O Pad: BDDPAUX_F3FCR90_VD181818NC

0	AUX	MODE: 0 = AUX 1 = I2C

DPAUX_PINMUX_CFG_0

PINMUX_CFG

This pinmux is implemented in LSIO cluster. It determines the interfaces for I2C6 is connected to real I2C6 or UART6.

The pinmux configurations are as follows:

- 0: DPAUX (Functional / Default)
- 1: UART6 (Debug)

UART is sharing I^2C pins internal to LSIO; UART uses I^2C mode of pad.

The signal generated by this register is output of DPAUX and input to LSIO cluster

Refer to the Display Interfaces: HDMI and DisplayPort chapter for more information on the DPAUX_HYBRID_PADCTL_0 and DPAUX_PINMUX_CFG_0 registers.

9.7.2.23.7 UART8/UARTH

UART8 is multiplexed over USB2. Pinmux does not exist for this.

UART8 <-> XUSB Pad Macro <-> USB PADLET <-> USB PAD.

For UART8 to be operational, set either of the following in XUSB_PADCTL_USB2_PAD_MUX_0 register.

- USB2_OTG_PAD_PORT0 = UART
  (or)
- USB2_OTG_PAD_PORT1 = UART

No other PAD programming sequence is needed. Above mentioned settings would configure the PAD in test mode as far as UART over USB2 pins is concerned. Note that fields USB2_OTG_PAD_PORT2 and USB2_OTG_PAD_PORT3 are NA.

9.7.3 UART Registers

Refer to "Reading Register Tables" in the Introduction chapter for the register table protocol as well as recommendations for accessing registers.

The UART is based on the 16550 industry standard Universal Asynchronous Receiver/Transmitter (UART), with enhancements to support auto-baud detection and End-of-Received Data timeout detection.

The THR, RBR, and DLL registers all occupy the same address space.

- The Transmitter Holding Register (THR) is Write-Only, and can be accessed if the LSR.DLAB bit is clear.
- The Receiver Buffer Register (RBR) is Read-Only, and can be accessed if the LSR.DLAB bit is clear.
- The Divisor Latch LSByte Register (DLL) is Read/Write and can be accessed if the LSR.DLAB bit is set.

UART_THR_DLAB_0_0

UART Transmit Holding Register

Offset: 0x0
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	R/W	Reset	Description
7:0	WO	0x0	THR_A: Transmit holding register, holds the character to be transmitted by the UART. In FIFO mode, a write to this FIFO places the data at the end of the FIFO.
7:0	RO	X	RBR_A: Receive Buffer Register. Rx Data read from here.
7:0	RW	0x0	DLL_A: Divisor Latch LSB (low 8 bits of 16-bit Baud Divisor)

UART_IER_DLAB_0_0

The IER and DLH registers occupy the same address space, selected by the LSR.DLAB bit. The Interrupt Enable Register (IER) is selected if the LSR.DLAB bit is clear. The Divisor Latch MSByte register (DLM) is selected if the LSR.DLAB bit is set. The DLM register holds the upper 8 bits of the 16-bit Baud Divisor (16x).

Offset: 0x4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5	0x0	IE_EORD: Interrupt Enable for End of Received Data (EORD). 0 = DISABLE 1 = ENABLE
4	0x0	IE_RX_TIMEOUT: Interrupt Enable for Rx FIFO timeout. RX_TIMEOUT 0 = DISABLE 1 = ENABLE
3	0x0	IE_MSI: Interrupt Enable for Modem Status Interrupt 0 = DISABLE 1 = ENABLE
2	0x0	IE_RXS: Interrupt Enable for Receiver Line Status Interrupt 0 = DISABLE 1 = ENABLE
1	0x0	IE_THR: Interrupt Enable for Transmitter Holding Register Empty interrupt 0 = DISABLE 1 = ENABLE
0	0x0	IE_RHR: Interrupt Enable for Received Data Interrupt 0 = DISABLE 1 = ENABLE

UART_IIR_FCR_0

The FIFO Control Register (FCR) and Interrupt Identification Registers (IIR) occupy the same address space. The FCR is write-only and the IIR is read-only.

UART FIFO Control Register POR = 0x00000000

The DMA can run in one of two modes:

1. If Mode0 is selected (NO_CHANGE), the Rx DMA request goes active whenever the Rx FIFO is not empty. Only single byte transactions can be performed in this mode. If the FIFO is not enabled, Mode0 MUST be used.
2. If Mode1 is selected (CHANGE), the Rx DMA request goes active whenever the Rx FIFO trigger level is reached.

For best performance, always enable the FIFO and select DMA Mode 1.

For RX_TRIG, the FIFO_COUNT references the number of bytes in the receive FIFO.

For TX_TRIG, the FIFO_COUNT references the number of empty bytes in the transmit FIFO. For example, FIFO_COUNT_GREATER_16 means that there are at least 16 empty byte slots in the TX_FIFO.

UART Interrupt ID Register POR = 0x00000001

The Interrupt ID field indicates the current highest priority interrupt. If more than one interrupt is pending the one with the highest priority will be shown.

The table below shows the encoding. Priority flows from top (highest) to bottom (lowest).

IIR[3:0]	{IS_PRI2, IS_PRI1,IS_PRI0,IS_STA} Priority Level
0001	No interrupt (IS_STA=1, No pending interrupt)
0110	Overrun Error, Parity Error, Framing Error, Break (IS_STA=0, overrun/parity/framing/break error interrupt is pending)
0100	Receiver Data Available (IS_STA=0, receiver data available interrupt is pending)
1100	rx_timeout_intr (IS_STA=0, Rx timeout interrupt is pending)
1000	eord_timeout_intr (IS_STA=0, end of received data interrupt is pending)
0010	Transmitter Holding Register empty (IS_STA=0, transmitter holding register empty interrupt is pending)
0000	modem_status interrupt (IS_STA=0, modem interrupt is pending)

UART FIFO Control and Interrupt Identification Registers

Offset: 0x8
Read/Write: See table below
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	R/W	Reset	Description
7:6	RO	X	EN_FIFO: FIFO Mode Status. Bit 7 and bit 6 are the same 0x0 = 16450 mode (no FIFO) 0x3 = 16550 mode (FIFO) 1 = MODE_16550 0 = MODE_16450
7:6	WO	0x0	RX_TRIG: 0 = FIFO_COUNT_GREATER_1 1 = FIFO_COUNT_GREATER_4 2 = FIFO_COUNT_GREATER_8 3 = FIFO_COUNT_GREATER_16
5:4	WO	0x0	TX_TRIG: 0 = FIFO_COUNT_GREATER_16 1 = FIFO_COUNT_GREATER_8 2 = FIFO_COUNT_GREATER_4 3 = FIFO_COUNT_GREATER_1
3	RO	X	IS_PRI2: Encoded Interrupt ID. Refer to the IIR[3:0] table above. 0 = DISABLE 1 = ENABLE
3	WO	0x0	DMA: 0: DMA_MODE_0 1: DMA_MODE_1 0 = NO_CHANGE 1 = CHANGE
2	RO	X	IS_PRI1: Encoded Interrupt ID. Refer to the IIR[3:0] table above. 0 = DISABLE 1 = ENABLE
2	WO	0x0	TX_CLR: 1 = Clears the contents of the transmit FIFO and resets its counter logic to 0 (the transmit shift register is not cleared or altered). This bit returns to 0 after clearing the FIFOs. 0 = NO_CLEAR 1 = CLEAR
1	RO	X	IS_PRI0: Encoded Interrupt ID. Refer to IIR[3:0] table above. 0 = DISABLE 1 = ENABLE
1	WO	0x0	RX_CLR: 1 = Clears the contents of the receive FIFO and resets its counter logic to 0 (the receive shift register is not cleared or altered). This bit returns to 0 after clearing the FIFOs. 0 = NO_CLEAR 1 = CLEAR
0	RO	X	IS_STA: Interrupt Pending if ZERO 0 = INTR_PEND 1 = NO_INTR_PEND
0	WO	0x0	FCR_EN_FIFO: 1 = Enable the transmit and receive FIFOs. This bit should be enabled 0 = DISABLE 1 = ENABLE

UART_LCR_0

UART Line Control Register

Offset: 0xc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7	0x0	DLAB: Divisor Latch Access Bit (set to allow programming of the DLH, DLM Divisors) 0 = DISABLE 1 = ENABLE
6	0x0	SET_B: Set BREAK condition. Transmitter sends all zeros to indicate BREAK 0 = NO_BREAK 1 = BREAK
5	0x0	SET_P: Set (force) parity to value in LCR [4] 0 = NO_PARITY 1 = PARITY
4	0x0	EVEN: Even parity format. There will always be an even number of 1s in the binary representation (PAR = 1) 0 = DISABLE 1 = ENABLE
3	0x0	PAR: 0 = No parity sent 0 = NO_PARITY 1 = PARITY
2	0x0	STOP: 0 = Transmit 1 stop bit 1 = Transmit 2 stop bits (receiver always checks for 1 stop bit) 0 = DISABLE 1 = ENABLE
1:0	0x0	WD_SIZE: 3 = Word length of 8 0 = WORD_LENGTH_5 1 = WORD_LENGTH_6 2 = WORD_LENGTH_7 3 = WORD_LENGTH_8

UART_MCR_0

The Request-To-Send (RTS) and Clear-To-Send (CTS) pins are used for hardware handshaking with an external serial device. RTS informs the device that the UART is ready to accept data. CTS comes from the RTS of the external device and indicates that data can be sent.

The RTS pin can be controlled manually with the RTS bit in the MCR or automatically by setting the RTS_EN bit.

If RTS_EN is set, the RTS pin automatically removes the Request-To-Send when the FIFO reaches the trigger level.

If the CTS_EN bit is set, the UART transmitter stops transmitting when the Clear-To-Send input is taken away.

Offset: 0x10
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,x000,0000,0000)

Bit	Reset	Description
10	0x0	OLD_SIR_DECODE: Diagnostics bit to use the required SIR decode path. 0 = new SIR decode path 1 = old SIR decode path 0 = DISABLE 1 = ENABLE
9:8	0x0	RI_POLARITY: Polarity selection bit for RI pin toggling to generate modem status interrupt. 0x0 - interrupt is generated when RI pin toggles from low to high 0x1 - interrupt is generated when RI pin toggles from high to low 0x2 - interrupt is generated on RI delta change detection 0x3 - Reserved 0 = LOW_TO_HIGH 1 = HIGH_TO_LOW 2 = BOTH_EDGES 3 = RSVD
7	0x0	DEL_QUAL_CTS_EN: Diagnostics bit to use the old qualified CTS in Tx state machine 1 = use old qualified_cts 0 = use the new qualified_cts in Tx state machine 0 = DISABLE 1 = ENABLE
6	0x0	RTS_EN: 1 = Enable RTS Hardware Flow Control 0 = DISABLE 1 = ENABLE
5	0x0	CTS_EN: 1 = Enable CTS Hardware Flow Control 0 = DISABLE 1 = ENABLE
4	0x0	LOOPBK: 1 = Enable internal loop back of Serial Out to In 0 = DISABLE 1 = ENABLE
3	0x0	OUT2: nOUT2 (Not Used) 0 = DISABLE 1 = ENABLE
2	0x0	OUT1: nOUT1 (Not Used) 0 = DISABLE 1 = ENABLE
1	0x0	RTS: 0 = Force RTS to high if RTS hardware flow control not enabled 0 = FORCE_RTS_HI 1 = FORCE_RTS_LOW
0	0x0	DTR: 1 = Force DTR to Low 0 = FORCE_DTR_HI 1 = FORCE_DTR_LOW

UART_LSR_0

UART Line Status Register

Offset: 0x14
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000260 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xx10,0110,0000)

Bit	Reset	Description
9	X	RX_FIFO_EMPTY: 1 = RX FIFO is empty 0 = RX FIFO has at least one entry 0 = NOT_EMPTY 1 = EMPTY
8	X	TX_FIFO_FULL: Transmitter FIFO full status 0 = NOT_FULL 1 = FULL
7	X	FIFOE: 1 = Receive FIFO Error 0 = NO_ERR 1 = ERR
6	X	TMTY: Transmit Shift Register empty status 0 = NO_EMPTY 1 = EMPTY
5	X	THRE: 1 = Transmit Holding Register is Empty – OK to write data 0 = FULL 1 = EMPTY
4	X	BRK: 1 = BREAK condition detected on line 0 = NO_BREAK 1 = BREAK
3	X	FERR: 1 = Framing Error 0 = NO_FRAME_ERR 1 = FRAME_ERR
2	X	PERR: 1 = Parity Error 0 = NO_PARITY_ERR 1 = PARITY_ERR
1	X	OVRF: 1 = Receiver Overrun Error 0 = NO_OVERRUN_ERROR 1 = OVERRUN_ERROR
0	X	RDR: 1 = Receiver Data Ready (Data available to read) 0 = NO_DATA_IN_FIFO 1 = DATA_IN_FIFO

UART_MSR_0

UART Modem Status Register

Offset: 0x18
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7	0x0	CD: State of Carrier detect pin 0 = DISABLE 1 = ENABLE
6	0x0	RI: State of Ring Indicator pin 0 = DISABLE 1 = ENABLE
5	0x0	DSR: State of Data set ready pin 0 = DISABLE 1 = ENABLE
4	0x0	CTS: State of Clear to send pin 0 = DISABLE 1 = ENABLE
3	0x0	DCD: Change (Delta) in CD state detected 0 = DISABLE 1 = ENABLE
2	0x0	DRI: Change (Delta) in RI state detected 0 = DISABLE 1 = ENABLE
1	0x0	DDSR: Change (Delta) in DSR state detected 0 = DISABLE 1 = ENABLE
0	0x0	DCTS: Change (Delta) in CTS state detected 0 = DISABLE 1 = ENABLE

UART_SPR_0

UART Scratchpad Register

Offset: 0x1c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	SPR_A: Scratchpad register (not used internally)

UART_IRDA_CSR_0

UART IrDA Pulse Coding CSR Register

Bits [7:6] control the infrared (SIR) encoding. If enabled, each zero bit is transmitted as a short IR pulse. No pulse is sent during idle or '1' data bits. The IR pulse can be either 3/16 or 4/16 of a normal bit time. The low 4 bits control signal polarity and apply whether or not SIR coding is enabled.

Offset: 0x20
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,00xx,0000)

Bit	Reset	Description
7	0x0	SIR_A: 1 = Enable SIR coder 0 = Disable SIR coder 0 = DISABLE 1 = ENABLE
6	0x0	PWT_A: 0 = 3/16th Baud Pulse 1 = 4/16th Baud Pulse 0 = BAUD_PULSE_3_14 1 = BAUD_PULSE_4_14
3	0x0	INVERT_RTS: Inverts the normally inactive high nRTS pin 0 = DISABLE 1 = ENABLE
2	0x0	INVERT_CTS: Inverts the normally inactive high nCTS pin 0 = DISABLE 1 = ENABLE
1	0x0	INVERT_TXD: Inverts the normally inactive high TXD pin 0 = DISABLE 1 = ENABLE
0	0x0	INVERT_RXD: Inverts the normally inactive high RXD pin 0 = DISABLE 1 = ENABLE

UART_RX_FIFO_CFG_0

Offset: 0x24
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0x00,0001)

Bit	Reset	Description
7	0x0	EN_RX_FIFO_TRIG: Enable use of RX_FIFO_TRIG count (obsoletes RX_TRIG when enabled) 0 = DISABLE 1 = ENABLE
5:0	0x1	RX_FIFO_TRIG: Set RX_FIFO trigger level (any value 1 through 32)

UART_MIE_0

UART Modem Interrupt Enable Register

Offset: 0x28
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000f (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,1111)

Bit	Reset	Description
3	0x1	DCD_INT_EN: Interrupt Enable for Change (Delta) in CD state detected 0 = DISABLE 1 = ENABLE
2	0x1	DRI_INT_EN: Interrupt Enable for Change (Delta) in RI state detected 0 = DISABLE 1 = ENABLE
1	0x1	DDSR_INT_EN: Interrupt Enable for Change (Delta) in DSR state detected 0 = DISABLE 1 = ENABLE
0	0x1	DCTS_INT_EN: Interrupt Enable for Change (Delta) in CTS state detected 0 = DISABLE 1 = ENABLE

UART_VENDOR_STATUS_0_0

UART Controller Status Register

Offset: 0x2c
Read/Write: RO
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000003 (0bxx00,0000,xx00,0000,xxxx,xxxx,xxxx,0011)

Bit	Reset	Description
29:24	X	TX_FIFO_COUNTER: The entry in this field reflects the number of current entries in the Tx FIFO
21:16	X	RX_FIFO_COUNTER: The entry in this field reflects the number of current entries in the Rx FIFO
3	X	TX_OVERRUN: This bit is set to 1 when write data is issued to the Tx FIFO when it is already full and gets cleared on register read (sticky bit until read) 0 = NO_OVERRUN 1 = OVERRUN
2	X	RX_UNDERRUN: This bit is set to 1 when a read is issued to an empty FIFO and gets cleared on register read (sticky bit until read) 0 = NO_UNDERRUN 1 = UNDERRUN
1	X	Reserved
0	X	Reserved

UART_ASR_0

UART Auto Sense Baud Register

The UART has autobaud sensing logic that can measure the width of the start bit of incoming data to determine baud rate. For this to work, the incoming character must have its LSB set. A <cr> works well (0x0d). The logic uses the UART device clock to count how wide the start pulse is, and the BUSY bit is set when the sensing is complete. The value in {RX_RATE_SENSE_H, RX_RATE_SENSE_L} should be divided by 16 with Round-to-Nearest, and the resulting value loaded into the DLM, DLL register pair to set the Baud Clock to 16x the Baud Rate.

Offset: 0x3c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b00xx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
31	0x0	VALID: This bit is set when the controller finishes counting the clocks between two successive clock edges after there is a write to ASR with don't care data. 0 = UN_SET 1 = SET

Bit	Reset	Description
30	0x0	BUSY: This bit is set when there is a write to ASR and is reset when the controller finishes counting the clock edges between two successive clock edges. 0 = NO_BUSY 1 = BUSY
15:8	0x0	RX_RATE_SENSE_H: Shows bits [15:8] of the count of clock edges between two successive clock edges.
7:0	0x0	RX_RATE_SENSE_L: Shows bits [7:0] of the count of clock edges between two successive clock edges.

9.8 Server Base System Architecture (SBSA) UART

9.8.1 Overview

The SBSA UART is a standard UART defined by Arm® as part of the Arm Server Base System Architecture (SBSA). It also supports Advanced Configuration and Power Interface (ACPI) for upstream kernel and standard OS boot.

9.8.1.1 SBSA

Server Based System Architecture (SBSA) is a hardware-system architecture based on Arm 64-bit architecture. The primary goal of the SBSA is to specify a system architecture to allow a suitably built single OS image to run on all hardware compliant with this specification. For system development and bring up, the base server system includes SBSA UART.

The Arm Server Base System Architecture specification is available from Arm's website. At the time of writing it is located at: https://developer.arm.com/documentation/den0029/c/.

9.8.1.2 UART

The Universal Asynchronous Receiver/Transmitter (UART) supports one-to-one communication. The UART controller provides serial data synchronization and data conversion (parallel-to-serial and serial-to-parallel) for both receiver and transmitter sections. UARTs transmit data asynchronously, so no clock signal is sent. Synchronization is done by adding start and stop bits to the data packet being transferred.

Data integrity is accomplished by attaching a parity bit to the data character. The parity bit can be checked by the receiver for any transmission bit errors. The interface supports word lengths from five to eight bits, an optional parity bit, and one or two stop bits

The UART controller supports both FIFO and non-FIFO modes of operation. FIFO mode provides two independent FIFOs for transmit and receive, and it is selected by the FIFO control register. The controller also includes a 16-bit programmable integer baud rate register, a 6-bit fractional programmable baud rate register, and two DMA handshake lines that are used to indicate when the FIFOs are ready for data transfers.

The UARTs require a device clock from CAR with 16 clock cycles per symbol for proper sampling and processing of the input data stream.

9.8.1.2.1 UART Use Cases

There are three SBSA UARTs built into Orin. One is in the Control Backbone Cluster, one is in the AON Cluster Fabric, and one is in the FSI.

Table 9.38 UART Allocation

Interface	Target Application	Max Baud Rate	Logical Placement	UART Pins on Interface
UARTI	Standard kernel boot and compliance to UEFI/ACPI. This instance is used explicitly by CCPLEX for kernel boot post early boot. Regular debug prints and early boot would be through UART-C, if this UART is used explicitly by CCPLEX.	Max - 4M Typical - 115200	TOP Plugged to CBB	TXD/RXD (no flow control) Pin-muxed with UART_E
UARTJ	Standard kernel boot and compliance to UEFI/ACPI. This instance can be either used/accessed by AON-R5 for early boot or by CCPLEX for standard kernel boot.	Max - 12M Typical - 115200	AON Plugged to AON Fabric	TXD/RXD (no flow control) Pin-muxed with UART_C
FSI_UART	Debug console support for FSI Cluster	Typical - 115200	FSI Plugged to FSI Fabric	TXD/RXD (no flow control) No Pin-Mux

Typical Baud Rates and its applications:

- Debug console: 115200 Baud/460800 Baud (4x115200)/921600 Baud
- Fast boot/ATE: 12M (192 MHz baud frequency)

Possible Baud rates supported - 9600/57600/115200/460800/3M/4M/12M

9.8.1.3 Feature List

The SBSA-UART Feature List table captures the supported features, showing both what the UART itself can support, and what the implementation in Orin can support.

Table 9.39 SBSA-UART Feature List

S. No	Feature	Sub-feature	Details	Orin Support
1	Frame Format	Data bits	5 data bits support	Y
		Data bits	6 data bits support	Y
		Data bits	7 data bits support	Y
		Data bits	8 data bits support	Y
		Data bits	9 data bits support	N
		Parity Bits	Even Parity support	Y
		Parity Bits	Odd Parity support	Y
		Stop bits	1 stop bit support	Y
		Stop bits	1.5 stop bit support	N
		Stop bits	2 stop bit support	Y
2	RX error detection	Rx FIFO	Overrun error	Y
		Character	Parity error	Y
		Character	Break error	Y
		Character	Framing error	Y
3	Baud rates	Baud Rates supported	9600/57600/115200/460800/3M/4M/12M	Y
		Baud tolerance	support +/- tolerance range	+-2%
4	Mode	FIFO mode (32-character FIFO's)		Y
		non-FIFO mode		Y
		DMA mode		Y
5	Flow control	CTS	Flow control in Tx path	N
		RTS	Flow control in Rx path	N
6	Flush	Tx path	Tx Flush	Y
		Rx path	Rx Flush	Y
7	SW Based Force conditions		Set Break condition - Tx to send continuous zeroes to indicate BREAK	Y
			Force Parity	Y
			Break during data transmission	Y
			Force RTS	Y

S. No	Feature	Sub-feature	Details	Orin Support
8	FIFO level trigger	RX_FIFO trigger	different trigger levels	Y
		TX_FIFO trigger	different trigger levels	Y
9	Loopback	Internal loopback	Internal loopback between Tx and Rx paths	Y
10	Compliance	ACPI/SBSA/Standard OS boot		Y
11	Interrupts	Tx Interrupt		Y
		Rx interrupt		Y
		Rx Timeout interrupt		Y
		Overrun error interrupt		Y
		Parity error interrupt		Y
		Break error interrupt		Y
		Framing error interrupt		Y
		EORD interrupt (end of receive data interrupt)		N
12	Modem support	Modem related pins support	DCD/DRI/DSR/DCTS/RI	N
13	Polarity		invert_TXD	N
			invert_RXD	N
			invert_CTS	N
			invert_RTS	N
14	SIR encoder/decoder	Pulse width	PWT=3/16th Baud Pulse	N
		Pulse width	PWT=4/16th Baud Pulse	N

9.8.1.4 Interrupt and Status Generation

The interrupts below are generated in the SBSA UART. These interrupts have mask controllability so software can mask the appropriate interrupt without getting asserted.

These interrupts have RAW interrupt status without considering the mask register bits, and masked interrupt status by considering the mask register bits. (Refer to the SBSA UART register section.)

1. OVERRUN INTERRUPT (OERIS) - set when there is overflow of Rx FIFO
2. BREAK INTERRUPT (BERIS) - set when there is break in the reception. Break condition is detected in Rx if input data line is held low for at least one full frame period.

3. PARITY INTERRUPT (PERIS) - set when there is parity error in the reception.
4. FRAME ERROR INTERRUPT (FERIS) - set when there is frame error in the reception.
5. RECEIVE TIMEOUT INTERRUPT (RTRIS) - set when there is no data transmission on the input line for at least a time period of four full data frames.
6. TX INTERRUPT (TXRIS) - set when the Tx FIFO data count is equal to or lower than the programmed trigger level. This can be cleared directly by software, or by writing data until the Tx FIFO data count is greater than the trigger level.
   Note that the interrupt is set is when the data count is transitioning to lower than trigger level. For example, if the trigger level is set to 16, interrupt is set when data count is crossing from 16 -> 15. If the UART is enabled, data count of Tx FIFO would be zero initially and does not trigger TXRIS interrupt.
   a. In non-FIFO mode, the interrupt is set if there is no data in FIFO. It is cleared by performing a single write to the transmit FIFO, or by clearing the interrupt.
7. RX INTERRUPT (RXRIS): This is set when the Rx FIFO data count is equal to or greater than programmed trigger level. This can be cleared directly by software, or by reading data until the Rx FIFO data count is below than trigger level.
   a. In non-FIFO mode, the interrupt is set if there is at least one frame of data to read. It is cleared by performing a single read to the Rx FIFO, or by clearing the interrupt.

All interrupts are combined to generate a final interrupt (sbsa_uart_intr) from the SBSA UART to interrupt controller.

9.8.1.5 UART DMA Interface

The UART provides an interface to connect to a DMA controller. The DMA operation of the UART is controlled using the *DMA Control Register, UARTDMACR*. The DMA interface includes the following signals:

DMA_TX_req generation:

The UART controller generates an APB DMA Tx ready request (sbsa_uart_dma_tx_req) to indicate its readiness to accept data from DMA. Whenever fifo_mode is enabled and FIFO data count is less than trigger level, tx_fifo_dma_ready is generated.

DMA_RX_req generation:

The DMA Rx request port is asserted when UART has sufficient data in its FIFO, as programmed by the watermark level, and so is ready to be read by DMA. For example, at 1/8 water mark level, the Rx request to DMA is asserted at least if there are four locations filled in Rx FIFO. It gets de-asserted when the FIFO has less than four locations. In this case, the burst length by DMA should be 4.

Table 9.40 DMA Trigger Points for Transmit and Receive FIFOs

Watermark Level	Transmit	Receive
	(number of empty locations)	(number of filled locations)
1/8th	28	4
1/4th	24	8
1/2nd	16	16
3/4th	8	24
7/8th	4	28

Configuration of Parity Signals:

- PEN – Parity Enable
- EPS – Even parity Select
- SPS – Sticky parity Select

These parity signals from SBSA_UART_LCR_H_0 register can be configured as shown below.

Table 9.41 Parity Signals of SBSA

SBSA_UARTLCR_H_0 PEN	SBSA_UARTLCR_H_0 EPS	SBSA_UARTLCR_H_0 SPS	Parity bit (transmitted or checked)
0	X	x	Not transmitted or checked
1	1	0	Even parity
1	0	0	Odd parity
1	0	1	1
1	1	1	0

9.8.1.6 Error Handling

The UART has an APB slave error port to generate slave error (APB 3.0 spec compliant) when there is an access to register that does not exist (address hole), or when a write request is issued to a read-only register.

There are four errors associated to every frame of read data that is read from the FIFO:

- SBSA_UARTDR_0[11:8] - these are RO status bits and provide the error bits along-side the data read from Rx FIFO.
- SBSA_UARTRSR_0[3:0] - also captures the same information, but a write to this register clears this content.

SBSA_UARTRSR_0:

This register captures the Frame/parity/Break/overrun errors associated with data that is present with them in SBSA_UARTDR_0. This register gets overwritten with new error status associated to the data currently present in SBSA_UARTDR_0.

Overrun error: This bit is set to 1 if data is received and the Rx FIFO is full. This bit is set along with data that is written into Rx FIFO after it is ready to accept the data after overflow is done. This bit is cleared to 0 by a write to SBSA_UARTRSR_0.

Break error: This bit is set to 1 if a break condition was detected, indicating that the received data input was held LOW for longer than a full-word transmission time (defined as start, data, parity, and stop bits).

In FIFO mode, this error is associated with the character at the top of the FIFO. When a break occurs, only one character of '0' is loaded into the FIFO. The next character is only enabled after the receive data input goes to a 1 (marking state) and the next valid start bit is received.

Parity error: When set to 1, it indicates that the parity of the received data character does not match the parity that the EPS and SPS bits in the Line Control Register, SBSA_UARTLCR_H.

This bit is cleared to 0 by a write to UARTRCR. In FIFO mode, this error is associated with the character at the top of the FIFO.

Framing error: This bit is set when a received character did not have a valid stop bit (a valid stop bit is 1). This bit is cleared to 0 by a write to SBSA_UARTRSR. In FIFO mode, this error is associated with the character at the top of the FIFO.

Programming Guidelines:

While software reads the data from the Rx FIFO, these errors are monitored by software by reading the error status information from registers. Interrupts associated to these errors and software can also rely on the interrupts to find if any error has occurred.

9.8.1.7 Clock and Reset

Clock Requirements

The UART controller requires two clocks:

- APB clock
- baud_clock - Max frequency of this clock is 204 MHz

9.8.1.7.1 baud_clk

Clock Sources

1. UARTI: PLLP_OUT0, CLK_S, CLK_M
   a. Possible frequencies supported for UART-I (in MHz) = 0.1536, 1.84, 7.37, 48, 62.76, 68, 204
2. UARTJ: PLLP_OUT0, PLLC_OUT0, PLLAON_OUT, PLLC4_muxed (controlled by PLLC4_CLK_SEL), CLK_S, OSC_UNDIV
   a. Possible frequencies supported for UART-J (in MHz) = 0.1536, 1.84, 7.37, 48, 62.76, 68, 192, 204

To achieve the required baud rate for a given UART controller, the CLK_RST_CONTROLLER_CLK_SOURCE_UART[I-J] _0 register is programmed. The source of clock can be chosen by programming [31:29] of this register and divisor by programming [15:0].

To achieve divisor value of 'X', the register is configured with value N=((X-1) *2)

Programming of bit 24 determines if the divider value is picked from

- CLK_RST_CONTROLLER_CLK_SOURCE_UART[I-J] _0[15:0]
- or SBSA_UARTIBRD_0_ BAUD_DIVINT [15:0]

But only one of them can be used at a time.

For example, source clock: CLK_M -> crystal clock of 38.4 MHz

- To achieve baud rate of 115200, baud_clock needed = 1.8432 MHz
- To achieve 1.8432 MHz with 38.4 MHz crystal clock, divisor needed is (38.4/1.8432) = 20.833333
- Since CAR registers provide divisors with granularity of 0.5, the nearest divisor can be either 20.5 or 21.
- For divisors of 20.5, 21 baud frequencies that can be achieved are (38.4/20.5) = 1.873 MHz and (38.4/21) = 1.8285 MHz, respectively. Based on the error %, the nearest baud rate that can be achieved is 1.8285 MHz
- For divisor of 21, value to be programmed in [15:0] is 0x28, [31:29] = 0x7

The following table summarizes the divisor to achieve 115200 baud rates with different source frequencies.

Osc Clk Frequency (M)	**For baud rate 115200, divisor needed D = (M *1000*1000) / (115200 *16)**	**Value of divisor (DR) = N/ 2+1**	**UART[i-J] CLK_DIVISOR (N)**
12	6.510	6.5	11 (=0xb)
13	7.053	7.0	12 (=0xc)
19.2	10.417	10.5	19 (=0x13)
26	14.106	14.0	26 (=0x1a)
38.4	20.833	21.0	40 (=0x28)

Osc Clk Frequency (M)	For baud rate 115200, divisor needed D = (M *1000*1000) / (115200 *16)	Value of divisor (DR) = N/ 2+1	UART[i-J] CLK_DIVISOR (N)
48	26.042	26.0	50 (=0x32)

9.8.2 Programming Guidelines

9.8.2.1 Initial Sequence (Clock, Reset, and Pinmux Programming)

1. Reset the controller by programming CLK_RST_CONTROLLER_RST_DEV_UART[I-J] _0_SWR_UART[I-J]_ST to 0x1.
2. Program the clock source and clock divider to achieve the required frequency.
3. Enable the clock to UART by programming CLK_RST_CONTROLLER_CLK_OUT_ENB_UART[I-J] _0
4. Program the pinmux settings to select the appropriate UART.
5. Remove the controller from reset by programming CLK_RST_CONTROLLER_RST_DEV_UART[I-J] _CLR_0 to 0x1.

9.8.2.2 Basic Guideline for Tx

1. Configure the SBSA_UARTLCH_0 and SBSA_UARTCR_0 registers according to the use case.
2. Enable UART after configuring the SBSA_UARTLCH_0 and SBSA_UARTCR_0 registers.
3. Write the data into SBSA_UARTDR_0 register if:

FIFO MODE	DMA	Write Data
0 (Non-FIFO)	0 (Non-DMA)	Write data to SBSA_UARTDR_0 once TX INTERRUPT is received or UARTDR is empty (indicated in UARTFR register)
1 (FIFO)	0 (Non-DMA)	Write the 8-bit FIFO data to SBSA_UARTDR_0 once TX INTERRUPT is received
1 (FIFO)	1 (DMA)	Set the trigger level in SBSA_UARTIFLS_0 register and configure the same the burst size equal to trigger level in DMA IP for data transfer. To write the data into Tx FIFO DMA Tx request is used which is generated by DMA flow control

4. Check for interrupts (as mentioned in the sequences below) for triggering next transaction depending on FIFO/NON_FIFO DMA/NON-DMA modes
5. Disable UART according to the sequence of steps mentioned in each of above combinations

9.8.2.3 Basic Guidelines for Rx

1. Configure the SBSA_UARTLCH_0 and SBSA_UARTCR_0 registers according to the use case

2. Enable UART after configuring the SBSA_UARTLCH_0 and SBSA_UARTCR_0 registers.
3. Read the data into SBSA_UARTDR_0 register if:

FIFO MODE	DMA	Read Data
0 (Non-FIFO)	X	Read data from SBSA_UARTDR_0 on checking RX INTERRUPT
1 (FIFO)	0 (Non-DMA)	Read data from SBSA_UARTDR_0 on checking RX INTERRUPT
1 (FIFO)	1 (DMA)	Set the trigger level in SBSA_UARTIFLS_0 register and configure the same the burst size equal to trigger level in DMA IP for data transfer. To transfer the data DMA Rx request is generated by DMA flow control.

4. Check for interrupts (as mentioned in the sequences below) for triggering next transaction depending on FIFO/NON_FIFO DMA/NON-DMA modes.
5. Disable UART according to the sequence of steps mentioned in each of above combinations.

9.8.2.4 Tx: Non-FIFO Mode Programming

1. Write to SBSA_UARTLCR_H_0 register as shown below.
   a. FEN: Set this bit to zero
   b. SPS: Sticky parity select (Configure this according to requirement)
   c. WLEN: Configure word length (2'b00 -> 5 bits; 2'b01-> 6 bits; 2'b10 -> 7 bits; 2'b11 -> 8 bits)
   d. STP2: Configure stop bit length (zero -> 1 stop bit; one -> 2 stop bits)
   e. EPS: Configure Even parity Select (zero for ODD parity; one for EVEN parity)
   f. PEN: Configure Parity Enable
   g. BRK: Configure Send Break
2. Write to SBSA_UARTCR_0 as per the configuration required.
   a. CTSEn: Clear to send Enable
   b. TXE: Configure UART Tx Enable (Set this to one to enable transmission)
   c. RXE: Configure UART Rx Enable by setting to zero
   d. LBE: Loop back enable (Set to zero, unless loopback is required)
   e. SIRLP: SIR low power mode (Must be set to zero)
   f. SIREN: SIR enable (Must be set to zero)
   g. UARTEN: UART enable (Do not set this bit in this step)
3. Set SBSA_UARTCR_0[0] = 1 UARTEN (UART enable)
4. Write data into SBSA_UARTDR_0[7:0] (one character)
5. Check for UART Busy signal from register SBSA_UARTFR_0[3] (UART BUSY).
6. If SBSA_UARTFR_0[3] =1 (TXFE indicates FIFO is empty), go to STEP 7
7. Else WAIT

   a. UART Busy: The BUSY signal goes HIGH as soon as data is written to the transmit FIFO (that is, the FIFO is non-empty) and remains asserted
8. Check for TX INTERRUPT which is indicated by TXRIS = 1 (SBSA_UARTRIS_0[5]) (or)
   a. Check for SBSA_UARTFR_0[7] = 1 (indicates Transmit FIFO is empty TXFE)
   b. Clear the interrupt by setting clear in appropriate interrupt clear bit.
9. Repeat the process from step 5
10. Disabling UARTEN
   a. UARTEN (SBSA_UARTCR_0[0] = 0) can be disabled in during transmission
   b. Wait for UART_BUSY = 0 for entire Tx to get disabled.

9.8.2.5 Tx: FIFO Mode Programming (Non-DMA)

1. Write to SBSA_UARTLCR_H_0 register as shown below.
   a. FEN: FIFO ENABLE (set this for FIFO mode)
   b. SPS: Sticky parity select (Configure this according to requirement)
   c. WLEN: Configure word length (2'b00 -> 5 bits; 2'b01-> 6 bits; 2'b10 -> 7 bits; 2'b11 -> 8 bits)
   d. STP2: Configure stop bit length (zero -> 1 stop bit; one -> 2 stop bits)
   e. EPS: Configure Even parity Select (zero for ODD parity; one for EVEN parity)
   f. PEN: Configure Parity Enable
   g. BRK: Configure Send Break
2. Write to SBSA_UARTCR_0 as per the configuration required.
   a. CTSEn: Clear to send Enable
   b. TXE: Configure UART Tx Enable
   c. RXE: Configure UART Rx Enable by writing zero
   d. LBE: Loop back enable (Set to zero, unless loopback is required)
   e. SIRLP: SIR low power mode (Must be set to zero)
   f. SIREN: SIR enable (Must be set to zero)
   g. UARTEN: UART enable (Do not set this bit in this step)
3. Write data into SBSA_UARTIFLS_0 to select the interrupt level. [2:0] TXIFLSEL Transmit interrupt FIFO level select. The trigger points for the transmit interrupt are as follows:
   - b000 = Transmit FIFO becomes ≤ 1/8 full
   - b001 = Transmit FIFO becomes ≤ 1/4 full
   - b010 = Transmit FIFO becomes ≤ 1/2 full
   - b011 = Transmit FIFO becomes ≤ 3/4 full
   - b100 = Transmit FIFO becomes ≤ 7/8 full
4. Write data into SBSA_UARTDR_0 one by one to reach the FIFO Trigger/water mark level. For example, if TXIFLSEL b100 trigger level is set for DMA, it means that a total of 28-byte data can be written into the UART by HOST or until the entire FIFO can be filled.

   a. Wait for UART BUSY to become one to indicate that data is written to FIFO and has become non-empty.
5. Set SBSA_UARTCR_0[0] = 1 (UARTEN)
6. Wait for Tx INTERRUPT, *TXRIS (SBSA_UARTRIS_0[5]),* indicating that FIFO data count has come down the water mark level.
7. Write data into SBSA_UARTDR_0 until the FIFO reaches above the water mark level and repeat step 7 and 8.
8. To disable the UART:
   a. Clear the UARTEN (SBSA_UARTCR_0[0] = 0) which can be done while Tx in progress and Wait for UART_BUSY = 0 for entire Tx to get disabled.
   b. FIFO Flushing (if required)
      i. Configure FEN = 0 (SBSA_UARTLCR_H_0[4]) if TX FIFO data is required to be flushed out. To retain the data, do not change FEN = 1 (SBSA_UARTLCR_H_0[4])
      ii. Wait for UART_BUSY = 0 for entire Tx to get disabled.

9.8.2.6 Tx: FIFO Mode Programming (DMA Mode)

1. Write to SBSA_UARTLCR_H_0 register as per the configuration required to fields below.
   a. FEN: FIFO ENABLE (set this for FIFO mode)
   b. SPS: Sticky parity select (Configure this according to requirement)
   c. WLEN: Configure word length (2'b00 -> 5 bits; 2'b01-> 6 bits; 2'b10 -> 7 bits; 2'b11 -> 8 bits)
   d. STP2: Configure stop bit length (zero -> 1 stop bit; one -> 2 stop bits)
   e. EPS: Configure Even parity Select (zero for ODD parity; one for EVEN parity)
   f. PEN: Configure Parity Enable
   g. BRK: Configure Send Break
2. Configure SBSA_UARTDMACR_0 by writing one into the TXDMAE field to enable Tx DMA.
3. Write to SBSA_UARTCR_0 as per the configuration required.
   a. CTSEn: Clear to send Enable
   b. TXE: Configure UART Tx Enable
   c. RXE: Configure UART Rx Enable by writing zero
   d. LBE: Loop back enable (Set to zero, unless loopback is required)
   e. SIRLP: SIR low power mode (Must be set to zero)
   f. SIREN: SIR enable (Must be set to zero)
   g. UARTEN: UART enable (Do not set this bit in this step)
4. Write data into SBSA_UARTIFLS_0 to select the interrupt level. [2:0] TXIFLSEL Transmit interrupt FIFO level select. The trigger points for the transmit interrupt are as follows:
   - b000 = Transmit FIFO becomes ≤ 1/8 full
   - b001 = Transmit FIFO becomes ≤ 1/4 full
   - b010 = Transmit FIFO becomes ≤ 1/2 full

- b011 = Transmit FIFO becomes ≤ 3/4 full
- b100 = Transmit FIFO becomes ≤ 7/8 full

5. Program DMA IP with required configurations for data transfer to/from UART.
   a. Set the burst size equal to trigger level
6. Set SBSA_UARTCR_0[0] = 1 (UARTEN)
   a. DMA flow control is based DMA_tx_req sideband signaling from the UART.
7. To disable:
   a. Disable the DMA first
   b. Clear the UARTEN (SBSA_UARTCR_0[0] = 0) which can be done while Tx in progress and Wait for UART_BUSY = 0 for entire Tx to be disabled.
   c. FIFO Flushing
      i. Configure FEN = 0 (SBSA_UARTLCR_H_0[4]) if TX FIFO data is required to be flushed out. If we want to retain the data, do not change FEN = 1 (SBSA_UARTLCR_H_0[4])
      ii. Wait for UART_BUSY = 0 for entire Tx to get disabled.

Table 9.42 DMA Water Mark Levels for Tx/Rx FIFOs

Watermark Level	Transmit	Receive
	(number of empty locations)	(number of filled locations)
1/8th	28	4
1/4th	24	8
1/2nd	16	16
3/4th	8	24
7/8th	4	28

9.8.2.7 Rx: Non-FIFO Mode Programming

1. Write to SBSA_UARTLCR_H_0 register as per the configuration required to fields below.
   a. FEN: Set this bit to zero
   b. SPS: Sticky parity select (Configure this according to requirement)
   c. WLEN: Configure word length (2'b00 -> 5 bits; 2'b01-> 6 bits; 2'b10 -> 7 bits; 2'b11 -> 8 bits)
   d. STP2: Configure stop bit length (zero -> 1 stop bit; one -> 2 stop bits)
   e. EPS: Configure Even parity Select (zero for ODD parity; one for EVEN parity)
   f. PEN: Configure Parity Enable
   g. BRK: Configure Send Break
2. Write to SBSA_UARTCR_0 as per the configuration required.
   a. CTSEn: Clear to send Enable

b. RXE: Configure UART Rx Enable by writing one
c. TXE: Configure UART Tx Enable by writing zero
d. LBE: Loop back enable (Set to zero, unless loopback is required)
e. SIRLP: SIR low power mode (Must be set to zero)
f. SIREN: SIR enable (Must be set to zero)
g. UARTEN: UART enable (Do not set this bit in this step)

3. Set SBSA_UARTCR_0[0] = 1 UARTEN (UART enable)
4. Check for RX INTERRUPT which is indicated by RXRIS = 1 (SBSA_UARTRIS_0[4]) (or) Check for SBSA_UARTFR_0[7] = 1 (indicates Rx FIFO is non-empty RXFE)
5. Read SBSA_UARTDR_0[7:0] to read the data from UART
   a. In case of any error interrupts, or if the Read data is associated with any Error flags, as described above under "Error Handling", software needs to take appropriate action.
6. Repeat the process from step 4
7. To disable the UART:
   a. Clear UARTEN (SBSA_UARTCR_0[0] = 0) which can be disabled in between transmission.
   b. Wait for 1 frame duration depending on baud rate programmed.
   c. Keep reading the SBSA_UARTDR_0[7:0] until Rx FIFO empty (sbsa_uartfr_0_rxfe) = zero

9.8.2.8 Rx: FIFO Mode Programming (Non-DMA)

1. Write to SBSA_UARTLCR_H_0 register as per the configuration required to fields below.
   a. FEN: FIFO ENABLE (set this for FIFO mode)
   b. SPS: Sticky parity select (Configure this according to requirement)
   c. WLEN: Configure word length (2'b00 -> 5 bits; 2'b01-> 6 bits; 2'b10 -> 7 bits; 2'b11 -> 8 bits)
   d. STP2: Configure stop bit length (zero -> 1 stop bit; one -> 2 stop bits)
   e. EPS: Configure Even parity Select (zero for ODD parity; one for EVEN parity)
   f. PEN: Configure Parity Enable
   g. BRK: Configure Send Break
2. Write to SBSA_UARTCR_0 as per the configuration required.
   a. CTSEn: Clear to send Enable
   b. RXE: Configure UART Rx Enable by writing one
   c. TXE: Configure UART Tx Enable by writing zero
   d. LBE: Loop back enable (Set to zero, unless loopback is required)
   e. SIRLP: SIR low power mode (Must be set to zero)
   f. SIREN: SIR enable (Must be set to zero)
   g. UARTEN: UART enable (Do not set this bit in this step)
3. Write data into SBSA_UARTIFLS_0 to select the interrupt level. [2:0] RXIFLSEL Receive interrupt FIFO level select. The trigger points for the Rx interrupt are as follows:

- b000 = Receive FIFO becomes ≥ 1/8 full
- b001 = Receive FIFO becomes ≥ 1/4 full
- b010 = Receive FIFO becomes ≥ 1/2 full
- b011 = Receive FIFO becomes ≥ 3/4 full
- b100 = Receive FIFO becomes ≥ 7/8 full

4. b101-b111 = reserved
5. Set SBSA_UARTCR_0[0] = 1 UARTEN (UART enable)
6. Check and wait for RX INTERRUPT which is indicated by RXRIS = 1 (SBSA_UARTRIS_0[4]).
7. Read SBSA_UARTDR_0[7:0] to read the data from UART one by one till the complete data (as per trigger level) is read out from Rx FIFO.
   a. In case of any error interrupts, or if the Read data is associated with any Error flags, as described above under "Error Handling," software needs to take appropriate action.
8. Repeat 5 and 6
9. To disable the UART:
   a. Clear the UARTEN (SBSA_UARTCR_0[0] = 0) which can be done while Rx in progress.
   b. For disabling normally,
      i. Wait for the 1 Frame time interval
      ii. Keep reading the SBSA_UARTDR_0[7:0] until Rx FIFO empty (sbsa_uartfr_0_rxfe) = zero
   c. For abrupt disabling by FIFO Flushing
      i. Configure FEN = 0 (SBSA_UARTLCR_H_0[4]) if Rx FIFO data is required to be flushed out.
      ii. Wait for Rx FIFO empty (sbsa_uartfr_0_rxfe) for entire Rx path to get disabled.

9.8.2.9 Rx: FIFO Mode Programming (DMA Mode)

1. Write to SBSA_UARTLCR_H_0 register as per the configuration required to fields below.
   a. FEN: FIFO ENABLE (set this for FIFO mode)
   b. SPS: Sticky parity select (Configure this according to requirement)
   c. WLEN: Configure word length (2'b00 -> 5 bits; 2'b01-> 6 bits; 2'b10 -> 7 bits; 2'b11 -> 8 bits)
   d. STP2: Configure stop bit length (zero -> 1 stop bit; one -> 2 stop bits)
   e. EPS: Configure Even parity Select (zero for ODD parity; one for EVEN parity)
   f. PEN: Configure Parity Enable
   g. BRK: Configure Send Break
2. Configure SBSA_UARTDMACR_0 by writing one into RXDMAE field to enable RX DMA.
3. Write to SBSA_UARTCR_0 as per the configuration required.
   a. CTSEn: Clear to send Enable
   b. RXE: Configure UART Rx Enable by writing one
   c. TXE: Configure UART Tx Enable by writing zero
   d. LBE: Loop back enable (Set to zero, unless loopback is required)

e. SIRLP: SIR low power mode (Must be set to zero)
f. SIREN: SIR enable (Must be set to zero)
g. UARTEN: UART enable (Do not set this bit in this step)

4. Write data into SBSA_UARTIFLS_0 to select the interrupt level. *[2:0] RXIFLSEL Receive interrupt FIFO level select.* The trigger points for the transmit interrupt are as follows:
   - b000 = Receive FIFO becomes ≥ 1/8 full
   - b001 = Receive FIFO becomes ≥ 1/4 full
   - b010 = Receive FIFO becomes ≥ 1/2 full
   - b011 = Receive FIFO becomes ≥ 3/4 full
   - b100 = Receive FIFO becomes ≥ 7/8 full
   - b101-b111 = reserved
5. Set SBSA_UARTCR_0[0] = 1 UARTEN (UART enable)
6. Program DMA IP with required configuration for data transfer to/from UART.
   a. Set the Burst size equal to trigger level
7. Set SBSA_UARTCR_0[0] = 1 (UARTEN)
   a. DMA takes care of flow control depending on DMA_rx_req sideband signaling from UART IP and does the transfer accordingly.
8. In case of any error's interrupts set as described in section Interrupt and Status Generation, or if the Read data is associated with any Error flags, software needs to take appropriate action based on the use case. One of the options is disabling the UART or dropping that character which is in error.
9. To disable the UART:
   a. Clear the UARTEN (SBSA_UARTCR_0[0] = 0) which can be done while Rx in progress.
   b. Then
      i. Wait for the 1 Frame time interval.
      ii. Continue reading the Rx FIFO as long as threshold number of characters are present. If there is any residual remaining, software needs to either read them in PIO mode or by doing a FIFO flush to make the Rx FIFO empty.
   c. For FIFO Flushing
      i. Configure FEN=0 (SBSA_UARTLCR_H_0[4]) if Rx FIFO data is required to be flushed out.
      ii. Wait for Rx FIFO empty (sbsa_uartfr_0_rxfe) for entire Rx path to get disabled.

9.8.2.10 BREAK Programming (Non-FIFO Mode) in Tx

//Assert sequence

1. To set break, disable the UART (UARTCR_0[0] = 0)
2. SET the break (UARTLCR_H_0[0] = 1)
3. Enable UART (UARTCR_0[0] = 0)

//De-assert sequence

1. Wait for two frames time duration.
2. Disable UARTEN = 0
3. Wait for UART Busy to become low (UARTFR[3] = 0)
4. Clear BRK (UARTLCR_H_0[0] = 0)
5. Enable UART. Continue transmitting the data

9.8.2.11 BREAK Programming by Not Flushing FIFO Data (FIFO mode) in Tx

//Assert sequence

1. To set break, disable the UART (UARTCR_0[0] = 0)
2. SET the break (UARTLCR_H_0[0] = 1)
3. Enable UART (UARTCR_0[0] = 0)

//De-assert sequence

1. Wait for two frames time duration.
2. Disable UARTEN = 0
3. Clear BRK (UARTLCR_H_0[0] = 0)
4. Enable UART.

**Note: UART_BUSY is always high throughout the sequence of step as we do not flush FIFO unless there is no data in FIFO

9.8.2.12 BREAK Programming Flushing FIFO Data (FIFO mode) in Tx

//Assert sequence

1. To set break, disable the UART (UARTCR_0[0] = 0) can be done anytime during ongoing transaction
2. Configure FEN=0 (SBSA_UARTLCR_H_0[4]) if Tx FIFO data is required to be flushed out
3. Wait for UART BUSY(UARTFR[3] = 0) to become low
   a. UART BUSY: signal indicating the end of current transaction and data non-availability in FIFO).
4. SET the break BRK(UARTLCR_H_0[0] = 1)
5. Enable UART (UARTCR_0[0] = 0)

//De-assert sequence

1. Wait for two frames
2. Disable UARTEN = 0

3. Wait for UART Busy to become low (UARTFR[3] = 0)
4. Clear BRK (UARTLCR_H_0[0] = 0)
5. Enable UART. Continue writing data to FIFO

9.8.2.13 Loopback

In loopback mode, serial output (SOUT) from Tx is loop-backed to Rx side serial input (SIN).

To configure the Loopback mode, all the programming guidelines are as described above except that the Loopback bit should be set while configuring SBSA_UARTCR_0 register.

9.8.3 SBSA UART Registers

SBSA_UARTDR_0

Data Register

A write to [7:0] bits acts as Tx Data for transmission and gets written into Tx FIFO of controller.
A write to Tx FIFO only happens when UARTCR[TXE] is set to 1.
A Read to [7:0] returns the received data from Rx FIFO which is at top of the FIFO; [11:8] is status flags associated to the character read through [7:0]

Offset: 0x0
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0000,0000)

Bit	R/W	Reset	Description
11	RO	0x0	**OE**: Overrun Error
10	RO	0x0	**BE**: Break Error
9	RO	0x0	**PE**: Parity Error
8	RO	0x0	**FE**: Framing Error
7:0	RW	0x0	**DATA**: Transmit/Received Data character

SBSA_UARTRSR_0

UART Receive Status Register/Error Clear Register. A write to this register clears all the four bits

Offset: 0x4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000)

Bit	Reset	Description
3	0x0	**OE**: Overrun Error
2	0x0	**BE**: Break Error
1	0x0	**PE**: Parity Error
0	0x0	**FE**: Framing Error

SBSA_UARTFR_0

UART Flag Register

1: indicates that UART is busy in transmitting. This bit remains set until the complete byte, including all the stop bits, has been sent from the shift register.
0: indicates that UART is in IDLE and no data to transmit. Note: UARTFR[2:0] - DCD, DSR, CTS, and UARTFR[8] - RI fields are reserved as these features are not supported

Offset: 0x18
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000090 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,1001,0000)

Bit	Reset	Description
8	0x0	**RI**: State of Ring Indicator. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
7	0x1	**TXFE**: Transmit FIFO empty. **0** = NOT_EMPTY **1** = EMPTY
6	0x0	**RXFF**: Receive FIFO full. **0** = NOT_FULL **1** = FULL
5	0x0	**TXFF**: Transmit FIFO full. **0** = NOT_FULL **1** = FULL
4	0x1	**RXFE**: Receive FIFO empty. **0** = NOT_EMPTY **1** = EMPTY
3	0x0	**BUSY**: **0** = NOT_BUSY **1** = BUSY
2	0x0	**DCD**: Data carrier detect. This bit is the complement of the UART data carrier detect nUARTDCD, modem status input. **0** = DISABLE **1** = ENABLE
1	0x0	**DSR**: Data set ready. This bit is the complement of the UART data set ready, nUARTDSR, modem status input. That is, the bit is 1 when nUARTDSR is LOW. **0** = DISABLE **1** = ENABLE
0	0x0	**CTS**: Clear to send. This bit is the complement of the UART clear to send, nUARTCTS, modem status input. That is, the bit is 1 when nUARTCTS is LOW. **0** = DISABLE **1** = ENABLE

SBSA_UARTILPR_0

IrDA Low-Power Counter Register (NOT USED)

Offset: 0x20
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**ILPDVSR**: 8-bit low-power divisor value. These bits are cleared to 0 at reset

SBSA_UARTIBRD_0

Integer Baud Rate Register

Offset: 0x24
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**BAUD_DIVINT**: The integer baud rate divisor

SBSA_UARTFBRD_0

Fractional Baud Rate Register (NOT USED)

Offset: 0x28
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00,0000)

Bit	Reset	Description
5:0	0x0	**BAUD_DIVFRAC**: The fractional baud rate divisor

SBSA_UARTLCR_H_0

UART Line Control Register

Offset: 0x2c
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7	0x0	**SPS**: Sticky Parity Bit (set to allow programming of the DLH, DLM Divisors) **0** = DISABLE **1** = ENABLE
6:5	0x0	**WLEN**: 3 = Word length of 8, 2 = Word length of 7, 1 = Word length of 6, 0 = Word length of 5. **0** = WORD_LENGTH_5 **1** = WORD_LENGTH_6 **2** = WORD_LENGTH_7 **3** = WORD_LENGTH_8
4	0x0	**FEN**: This bit is to select FIFO/non-FIFO mode. FIFO mode is disabled by default. **0** = DISABLE **1** = ENABLE
3	0x0	**STP2**: 1 = Two stop bits, 0 = One stop bit. **0** = ONE **1** = TWO
2	0x0	**EPS**: 0 = Odd parity select, 1 = Even parity select. **0** = ODD **1** = EVEN
1	0x0	**PEN**: 0 = Parity Disabled, 1 = Parity Enabled. **0** = NO_PARITY **1** = PARITY
0	0x0	**BRK**: If this bit is set to 1, a low-level is continually output on the UARTTXD output, after completing transmission of the current character. For the proper execution of the break command, the software must set this bit for at least two complete frames. For normal use, this bit must be cleared to 0. **0** = NO_BREAK **1** = BREAK

SBSA_UARTCR_0

Control Register
The RTS and CTS pins are used for hardware handshaking with an external serial device.

- RTS (Request-To-Send) informs the device that the UART is ready to accept data.
- CTS (Clear-To-Send) comes from the RTS of the external device and informs us that it is OK to send data.

Bits [6:3] of UARTCR are reserved writing '1' into TXE field enables the Tx data writing into Tx FIFO even if UARTEN is not set; Tx data transmission starts from Tx FIFO, when UARTEN is set. In Receive path, both RXE and UARTEN should be set in order to start receiving the data and store in RXFIFO.

Note: bits UARTCR[13:12] - Out2, Out1; UARTCR[10] - DTR; UARTCR[2:1] - SIREN, SIRLP are reserved and are not supported in RTL.

Offset: 0x30
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000300 (0bxxxx,xxxx,xxxx,xxxx,0000,0011,0xxx,x000)

Bit	Reset	Description
15	0x0	**CTS_EN**: 1 = Enable CTS Hardware Flow Control **0** = DISABLE **1** = ENABLE
14	0x0	**RTS_EN**: 1 = Enable RTS Hardware Flow Control **0** = DISABLE **1** = ENABLE
13	0x0	**Out2**: This bit is the complement of the UARTOut2 (nUARTOut2) modem status output. For DTE this can be used as Ring Indicator (RI). **0** = DISABLE **1** = ENABLE
12	0x0	**Out1**: This bit is the complement of the UARTOut1 (nUARTOut1) modem status output. For DTE this can be used as Data Carrier Detect (DCD). **0** = DISABLE **1** = ENABLE
11	0x0	**RTS**: RTS = 1, Request to send the data. **0** = RTS_LOW **1** = RTS_HIGH
10	0x0	**DTR**: DTR = 1, Data is ready to transmit. **0** = DTR_LOW **0** = DTR_HIGH
9	0x1	**RXE**: 1 = Receive Enable. **0** = RX_DISABLE **1** = RX_ENABLE
8	0x1	**TXE**: 1 = Transmit Enable. **0** = TX_DISABLE **1** = TX_ENABLE
7	0x0	**LBE**: Disable Loopback by default. **0** = LPBCK_DISABLE **1** = LPBCK_ENABLE

Bit	Reset	Description
2	0x0	**SIRLP**: SIR low-power IrDA mode is set. **0** = NOT_SIRLOWP **1** = SIRLOWP
1	0x0	**SIREN**: 1 = IrDA SIR ENDEC is enabled; 0 = Regular TXD/RXD data transmission. **0** = SIR_DISABLE **1** = SIR_ENABLE
0	0x0	**UARTEN**: 0 = UART is disabled. If the UART is disabled in the middle of transmission or reception, it completes the current character before stopping. 1 = the UART is enabled. Data transmission and reception occurs for either UART signals or SIR signals depending on the setting of the SIREN bit. **0** = UART_DISABLE **1** = UART_ENABLE

SBSA_UARTIFLS_0

Interrupt FIFO Level Select Register
Transmit interrupt FIFO level select. Tx interrupt is raised when FIFO data count comes down to programmable threshold/trigger level; for example of 000, Tx interrupt is asserted when Tx FIFO data count comes down to four locations (1/8 full).
Note: The transmit interrupt is based on a transition through a level, rather than on the level itself.

- b000 = Transmit FIFO becomes <= 1/8 full
- b001 = Transmit FIFO becomes <= 1/4 full
- b010 = Transmit FIFO becomes <= 1/2 full
- b011 = Transmit FIFO becomes <= 3/4 full
- b100 = Transmit FIFO becomes <= 7/8 full
- b101-b111 = reserved

Receive interrupt FIFO level select. The trigger points for the receive interrupt are as follows:

- b000 = Receive FIFO becomes >= 1/8 full
- b001 = Receive FIFO becomes >= 1/4 full
- b010 = Receive FIFO becomes >= 1/2 full
- b011 = Receive FIFO becomes >= 3/4 full
- b100 = Receive FIFO becomes >= 7/8 full
- b101-b111 = reserved

Offset: 0x34
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000012 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx01,0010)

Bit	Reset	Description
5:3	0x2	**RXIFLSEL**
2:0	0x2	**TXIFLSEL**

SBSA_UARTIMSC_0

Interrupt Mask Set/Clear
Writing 1 into a particular field will enable respective interrupt to get asserted to LIC.
Writing 0 into a particular field will disable/mask respective interrupt to LIC

Offset: 0x38
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,x000,0000,0000)

Bit	Reset	Description
10	0x0	**OEIM**: Overrun error interrupt mask. **0** = OVR_INTR_DISABLED_MASKED **1** = OVR_INTR_ENABLED
9	0x0	**BEIM**: Break error interrupt mask. **0** = BRK_INTR_DISABLED_MASKED **1** = BRK_INTR_ENABLED
8	0x0	**PEIM**: Parity error interrupt mask. **0** = PTY_INTR_DISABLED_MASKED **1** = PTY_INTR_ENABLED
7	0x0	**FEIM**: Framing error interrupt mask. **0** = FRM_INTR_DISABLED_MASKED **1** = FRM_INTR_ENABLED
6	0x0	**RTIM**: Receive timeout interrupt mask. **0** = TIMEOUT_INTR_DISABLED_MASKED **1** = TIMEOUT_INTR_ENABLED
5	0x0	**TXIM**: Transmit interrupt mask. **0** = TX_INTR_DISABLED_MASKED **1** = TX_INTR_ENABLED

Bit	Reset	Description
4	0x0	**RXIM**: Receive interrupt mask. **0** = RX_INTR_DISABLED_MASKED **1** = RX_INTR_ENABLED
3	0x0	**DSRMIM**: nUARTDSR modem interrupt mask. **0** = DSR_INTR_DISABLED_MASKED **1** = DSR_INTR_ENABLED
2	0x0	**DCDMIM**: nUARTDCD modem interrupt mask. **0** = DCD_INTR_DISABLED_MASKED **1** = DCD_INTR_ENABLED
1	0x0	**CTSMIM**: nUARTCTS modem interrupt mask. **0** = CTS_INTR_DISABLED_MASKED **1** = CTS_INTR_ENABLED
0	0x0	**RIMIM**: nUARTRI modem interrupt mask. **0** = RI_INTR_DISABLED_MASKED **1** = RI_INTR_ENABLED

SBSA_UARTRIS_0

Raw Interrupt Status Register
1 = indicates that respective interrupt is set.
0 = indicates there is no interrupt. Note: The Raw Interrupt Status (RIS) register is not impacted/ changed irrespective of value of UARTIMSC

Offset: 0x3c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,x000,0000,0000)

Bit	Reset	Description
10	0x0	**OERIS**: Overrun error interrupt **0** = NO_OVR_ERR **1** = OVR_ERR
9	0x0	**BERIS**: Break error interrupt **0** = NO_BRK_ERR **1** = BRK_ERR
8	0x0	**PERIS**: Parity error interrupt **0** = NO_PTY_ERR **1** = PTY_ERR

Bit	Reset	Description
7	0x0	**FERIS**: Framing error interrupt **0** = NO_FRM_ERR **1** = FRM_ERR
6	0x0	**RTRIS**: Receive timeout interrupt **0** = NO_TIMEOUT **1** = TIMEOUT
5	0x0	**TXRIS**: Transmit interrupt **0** = NO_TX_INTR **1** = TX_INTR
4	0x0	**RXRIS**: Receive interrupt **0** = NO_RX_INTR **1** = RX_INTR
3	0x0	**DSRRMIS**: nUARTDSR modem interrupt **0** = NO_DSR_INTR **1** = DSR_INTR
2	0x0	**DCDRMIS**: nUARTDCD modem interrupt **0** = NO_DCD_INTR **1** = DCD_INTR
1	0x0	**CTSRMIS**: nUARTCTS modem interrupt **0** = NO_CTS_INTR **1** = CTS_INTR
0	0x0	**RIRMIMS**: nUARTRI modem interrupt **0** = NO_RI_INTR **1** = RI_INTR

SBSA_UARTMIS_0

Masked Interrupt Status Register

1: indicates that respective interrupt is set/active.
0: indicates there is no interrupt. Note: this register status is updated/changed based on UARTIMSC register configuration

Offset: 0x40
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,x000,0000,0000)

Bit	Reset	Description
10	0x0	**OEMIS**: Overrun error interrupt mask status **0** = OVR_INTR_INACTIVE **1** = OVR_INTR_ACTIVE
9	0x0	**BEMIS**: Break error interrupt mask status **0** = BRK_INTR_INACTIVE **1** = BRK_INTR_ACTIVE
8	0x0	**PEMIS**: Parity error interrupt mask status **0** = PTY_INTR_INACTIVE **1** = PTY_INTR_ACTIVE
7	0x0	**FEMIS**: Framing error interrupt mask status **0** = FRM_INTR_INACTIVE **1** = FRM_INTR_ACTIVE
6	0x0	**RTMIS**: Receive timeout interrupt mask status **0** = TIMEOUT_INTR_INACTIVE **1** = TIMEOUT_INTR_ACTIVE
5	0x0	**TXMIS**: Transmit interrupt mask status **0** = TX_INTR_INACTIVE **1** = TX_INTR_ACTIVE
4	0x0	**RXMIS**: Receive interrupt mask status **0** = RX_INTR_INACTIVE **1** = RX_INTR_ACTIVE
3	0x0	**DSRMMIS**: nUARTDSR modem interrupt mask status **0** = DSR_INTR_INACTIVE **1** = DSR_INTR_ACTIVE
2	0x0	**DCDMMIS**: nUARTDCD modem interrupt mask status **0** = DCD_INTR_INACTIVE **1** = DCD_INTR_ACTIVE
1	0x0	**CTSMMIS**: nUARTCTS modem interrupt mask status **0** = CTS_INTR_INACTIVE **1** = CTS_INTR_ACTIVE
0	0x0	**RIMMIS**: nUARTRI modem interrupt mask status **0** = RI_INTR_INACTIVE **1** = RI_INTR_ACTIVE

SBSA_UARTICR_0

Interrupt clear Register
Writing a 1 into particular field clears respective interrupt in both UARTMIS/UARTRIS and eventual to LIC.
Writing a 0 has no effect/impact

Offset: 0x44
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,x000,0000,0000)

Bit	Reset	Description
10	0x0	**OEIC**: Overrun error interrupt clear **0** = NO_OVR_ERR_CLR **1** = OVR_ERR_CLR
9	0x0	**BEIC**: Break error interrupt clear **0** = NO_BRK_ERR_CLR **1** = BRK_ERR_CLR
8	0x0	**PEIC**: Parity error interrupt clear **0** = NO_PTY_ERR_CLR **1** = PTY_ERR_CLR
7	0x0	**FEIC**: Framing error interrupt clear **0** = NO_FRM_ERR_CLR **1** = FRM_ERR_CLR
6	0x0	**RTIC**: Receive timeout interrupt clear **0** = NO_TIMEOUT_CLR **1** = TIMEOUT_CLR
5	0x0	**TXIC**: Transmit interrupt clear **0** = NO_TX_INTR_CLR **1** = TX_INTR_CLR
4	0x0	**RXIC**: Receive interrupt clear **0** = NO_RX_INTR_CLR **1** = RX_INTR_CLR
3	0x0	**DSRMIC**: nUARTDSR modem interrupt clear **0** = NO_DSR_INTR_CLR **1** = DSR_INTR_CLR
2	0x0	**DCDMIC**: nUARTDCD modem interrupt clear **0** = NO_DCD_INTR_CLR **1** = DCD_INTR_CLR

Bit	Reset	Description
1	0x0	**CTSMIC**: nUARTCTS modem interrupt clear **0** = NO_CTS_INTR_CLR **1** = CTS_INTR_CLR
0	0x0	**RIMIC**: nUARTRI modem interrupt clear **0** = NO_RI_INTR_CLR **1** = RI_INTR_CLR

SBSA_UARTDMACR_0

DMA Control Register

Offset: 0x48
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**DMAONERR**: DMA on error. If this bit is set to 1, the DMA receive request outputs **0** = DISABLE **1** = ENABLE
1	0x0	**TXDMAE**: 1 = DMA for the transmit FIFO is enabled **0** = DISABLE **1** = ENABLE
0	0x0	**RXDMAE**: 1 = DMA for the receive FIFO is enabled **0** = DISABLE **1** = ENABLE

9.9 Serial Peripheral Interface (SPI)

9.9.1 Overview

The Serial Peripheral Interface (SPI) controller in the System-on-Chip (SoC) is a serial communications link between the processor and on/off-chip serial peripheral devices such as sensors, ADC/DAC devices, and Flash controllers.

The SPI controller supports both Master and Slave modes of SPI operation on this interface. The SoC embeds five SPI controllers, each of which works independently of others. In the context of the I/O pins and Controller registers, these five SPI controllers are referred to as SPI1, SPI2, SPI3, SPI4, and SPI5.

9.9.1.1 Standard and Compatibility

The Synchronous Serial Communication Interface, (Serial Peripheral Interface bus) works with both Master and Slave Mode. The SPI controller works on the Serial protocol (Serial Peripheral Interface) and has the following compatible features:

- Master/Slave
- Clock Polarity and Phase
- Chip Select Polarity

9.9.1.2 Glossary

Note that different names may end up in the same acronym when the entire TRM is put in perspective. The acronyms listed here are within the context of the SPI chapter.

Term	Definition
Word	A 32-bit data in the Tx or Rx FIFO.
Packet	The fixed-length data that is transmitted or received. ▪ All transmitting/receiving must be in Packets. ▪ Packet size can 4, 8, 16, or 32 bits for Packed Tx/Rx or 4 ~ 32 bits for Unpacked Tx/Rx
Transfer	Transmitting (Tx) or Receiving (Rx) of Packets.
Transaction	A sequence of Packets in Tx or Rx direction.
WRITE	Writing to the device the SPI controller interfaces with. ▪ WRITE operations are Tx operations for the SPI controller.
READ	Reading from the device the SPI controller interfaces with. ▪ READ operations are Rx operations for the SPI controller.
DMA	DMA operations accomplished using the DMA engines, GPC_DMA or BPMP_DMA, as discussed in the General-Purpose Direct Memory Access (GP-DMA) chapter.

9.9.1.2.1 Relevant Chapters in the TRM

- Address Map
- Clock Controller and Reset (CAR)
- General-Purpose Direct Memory Access (GP-DMA)
- Multi-Purpose I/O Pins and Pin Multiplexing (PinMux)

9.9.1.3 Features

- Master Mode Operation
  - All transfer modes (Mode 0, Mode 1, Mode 2, Mode 3) supported for both Tx and Rx transactions
- Slave Mode Operation
  - Slave Tx: Mode 1 and Mode 3 supported
  - Slave Rx: All transfer modes (Mode 0, Mode 1, Mode 2, Mode 3) supported
- Independent Rx and Tx FIFO
- FIFO Size: 64 x 32 bits
- Programmable packet sizes of 4 to 32 bits
- Packed and Unpacked Mode
  - Four Packed Packet Sizes:
    - Master: 4, 8, 16, 32 bits
    - Slave: 8, 16, 32 bits
  - Unpacked Packet Size
    - Master: 4 ~ 32 bits
    - Slave: 8 ~ 32 bits
- PIO or DMA Mode depending on total transfer sizes and packet size
  - PIO Mode: transfer sizes <= 64 words (32-bit)
  - DMA Mode: transfer sizes > 64 words (32-bit); limited to 64 Ki words (32-bit) per transfer
- Continuous mode
- Programmable Clock Phase and Polarity
- Programmable Delay between Consecutive Transfers
- Chip select (CS) Controllable by Software or Generated by Hardware on Packet Boundaries
- Maximum 4-chip Support with Programmable CS Polarity for Each Chip Select
- Maximum Data Rate: 81.6 MHz in Master mode and 51 MHz in Slave mode

9.9.2 Functional Description

The SPI Controller works as a Master/Slave on the SPI bus. It has independent Tx and Rx FIFOs of 64 x 32 bits each. Software can program the controller to generate transactions of a required number of packets of specific packet size on the SPI bus, where a transaction is a sequence of packets in either direction.

It can be operated in two modes:

1. The PIO Mode requires software to read and write FIFOs for handling data transfers between system memory and FIFO.

2. The DMA Mode uses a channel of the GP-DMA controller (outside of SPI) to transfer data between system memory and the FIFOs. Refer to the General-Purpose Direct Memory Access (GP-DMA) chapter.

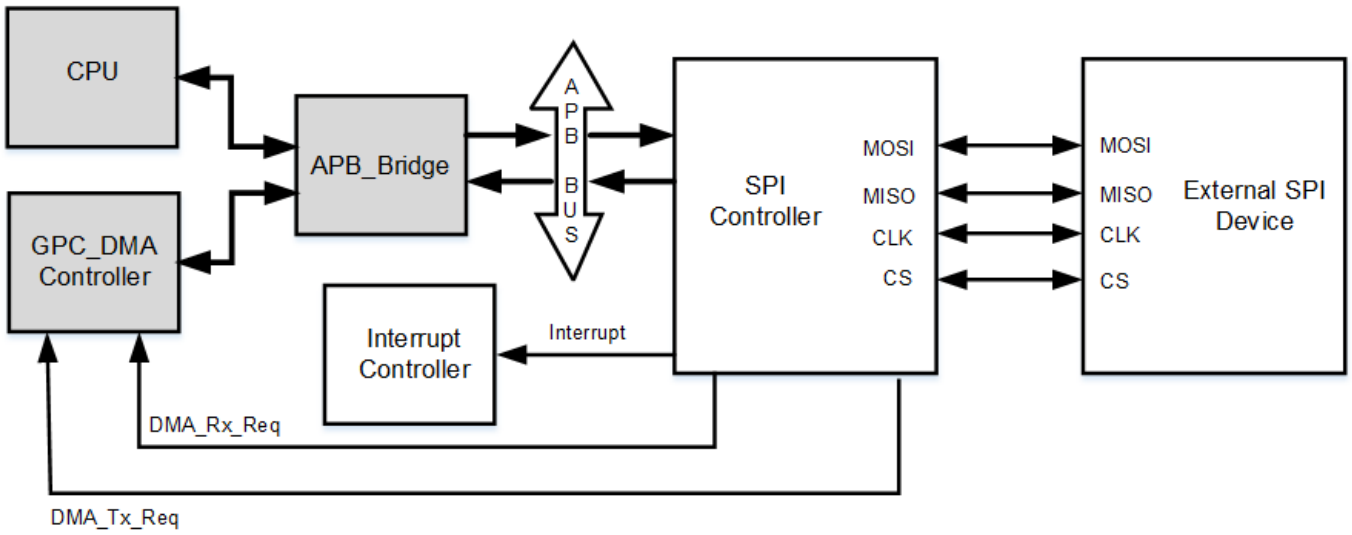
At the end of each transaction, an Interrupt is generated, if enabled. Software uses Tx and Rx operations in combination with Chip Select (CS) control to generate commands on the SPI bus.

The SPI Interface consists of four signals defined in the table below and illustrated in the figure after.

Table 9.43 SPI Interface Signals

Signal Name	Type	Description
SCK	Output	Serial Clock
CS	Output	Chip Select/Slave Select Typically active Low; programmable by SPI_COMMAND_0.CS_POL_INACTIVE0.
MOSI	Output / Bi-directional	Master Data Out; Slave Data In
MISO	Input / Bi-directional	Master Data In; Slave Data Out

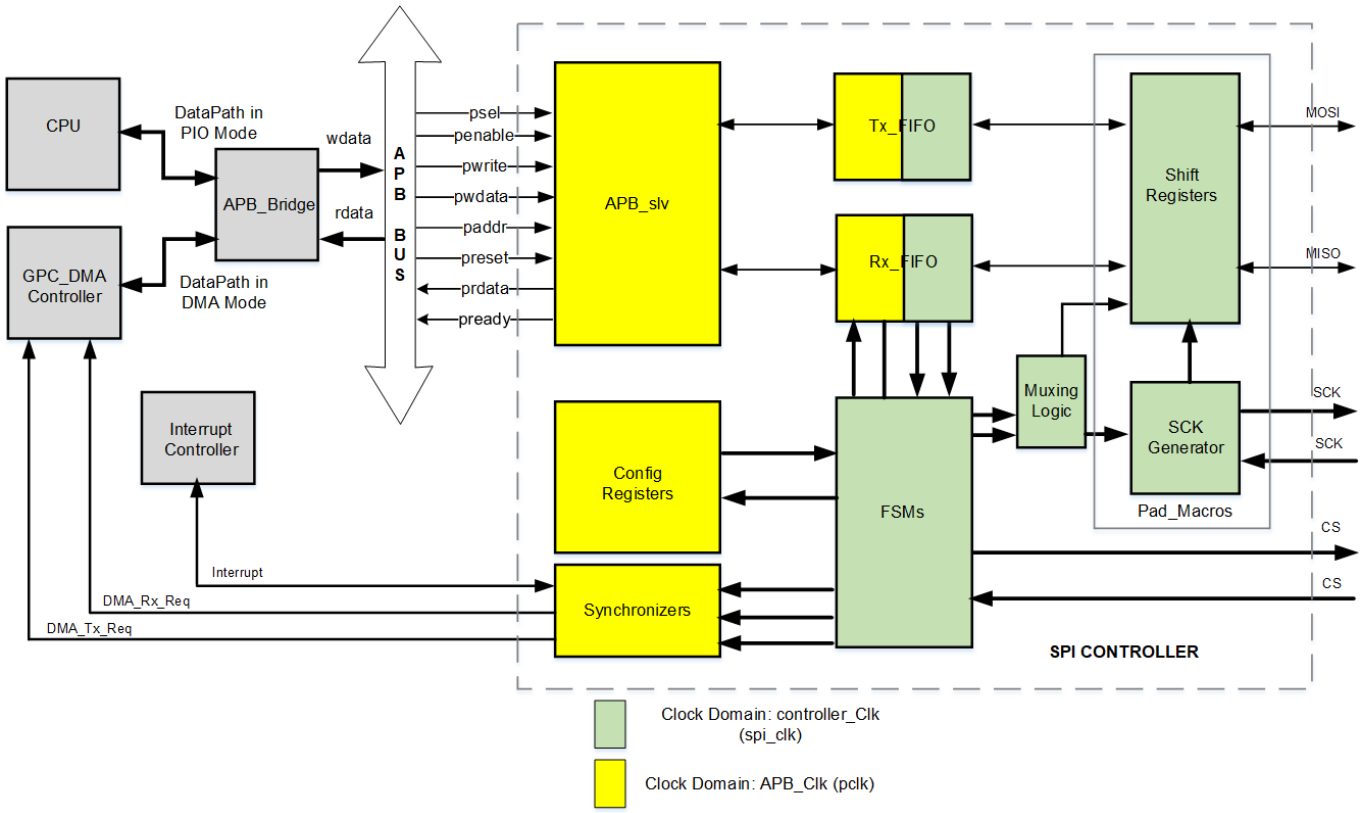
Figure 9.29 SPI Controller System Diagram



Note: Directions of the signals in the above figure vary depending on how the SPI Controller is configured as a Master or Slave.

The figure below shows the block diagram of the SPI controller. The color code indicates the clock domains of the blocks and the clock domain crossings (CDCs).

Figure 9.30 SPI Controller Block Diagram – Clock Domain Crossings



9.9.2.1 Transmission Format

Using the MODE field in the SPI_COMMAND_0 Register, software sets one of the four modes in which the SPI controller works. The two bits of the Mode field specify the idle (inactive) state of SCK before the chip select is asserted and the data transmitting/receiving edges of SCK. Mode [1] determines the SCK polarity. If the polarity is "0" (Low), then the SCK signal is "0" (Low) when idle and transitions to "1" (High) when data transfer starts. If the clock polarity is "1" (High), then the SCK signal is "1" (High) when idle and transitions to "0" when data transfer starts. Mode [0] is the SCK phase control bit. If the clock phase is "0," then the receiver latches the data on the first transition of SCK from the idle state. If the clock phase is "1," then the receiver latches the data on the second transition of SCK.

Modes supported by the SPI controller are:

- Master
  - All modes for Tx and Rx
- Slave
  - Mode 1 and Mode 3 for Tx
  - All modes for Rx

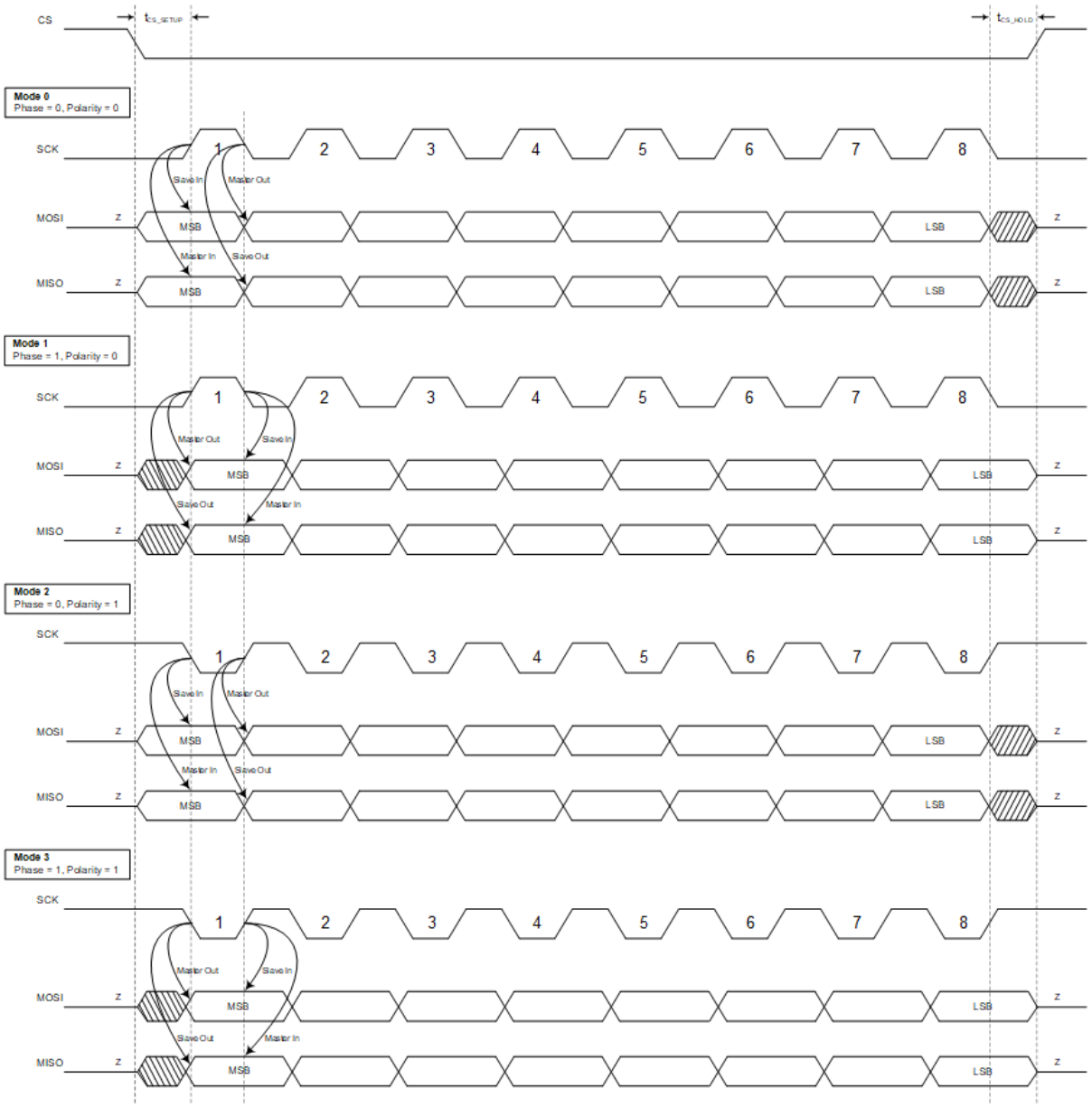
The following table defines the four modes for the SPI protocol.

Table 9.44 SPI Modes with Clock Polarity and Phase

SPI Mode	Clock Polarity	Clock Phase	SCK Inactive	Data Latch IN	Data Latch OUT
0	0	0	Low	on positive edges of clock	on negative edges of clock
1	0	1	Low	on negative edges of clock	on positive edges of clock
2	1	0	High	on negative edges of clock	on positive edges of clock
3	1	1	High	on positive edges of clock	on negative edges of clock

The figure below shows the SPI CS, MOSI, and MISO data lines and SCK for different modes.

Figure 9.31 SPI Timing of Mode 0, 1, 2, 3



9.9.2.2 Modes of Operation

9.9.2.2.1 Master and Slave Modes

The SPI controller can be configured to operate as a Master or Slave. A "1" in the M_S bit of the SPI_COMMAND_0 Register puts the SPI controller in Master Mode whereas a "0" puts the controller in Slave Mode. Only in Master Mode, the SPI controller can initiate a transaction.

In Master Mode, data from the Tx FIFO is transmitted on the MOSI pin, whereas data from the Slave is received on the MISO pin and sent to the Rx FIFO. The Master can simultaneously transmit and receive on MOSI and MISO in Full-Duplex Mode.

In Slave mode, once software enables the SPI controller by setting the PIO bit, it waits for the Master to initiate a transaction. Before the transaction begins, the Slave logic continuously polls the CS input. When the Master asserts CS and SCK is transmitted to the Slave, the Slave data is transmitted from the Tx FIFO on MISO and data from MOSI is received in the Rx FIFO. The Slave can also simultaneously transmit and receive on MOSI and MISO in Full-Duplex Mode.

Table 9.45 Master Mode Port Configuration

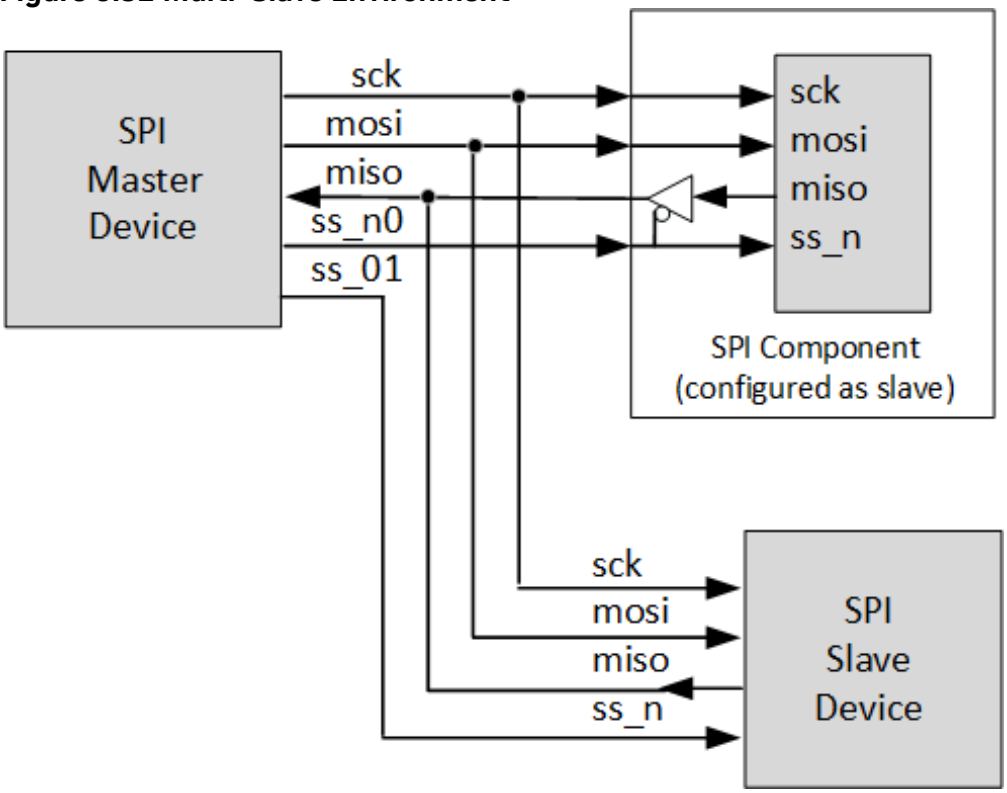
Name	Direction	Description
MOSI	Bi-directional	Output in Full-Duplex/Tx Mode. Input in Both_Enable Rx Mode.
MISO	Bi-directional	Input in Full-Duplex/Rx Mode. Output in Both_Enable Tx Mode.
SCK	Output	Synchronization clock to all Slaves.
CS	Output	Slave select signal to x, where x is any of {0, 1, 2, 3}.

Table 9.46 Slave Mode Port Configuration

Name	Direction	Description
MISO	Bi-directional	Output in Full-Duplex/Tx Mode. Input in Both_Enable Rx Mode.
MOSI	Bi-directional	Input in Full-Duplex/Rx Mode. Output in Both_Enable Tx Mode.
SCK	Input	Synchronization clock received from the Master.
CS	Input	Select signal.

When the SPI controller is configured to interface with multiple Slaves, the controller has one CS signal for each Slave, up to a maximum of four Slaves. During a transfer, the Master asserts CS specified in the CS_SEL field in SPI_COMMAND_0 Register. At any given time, there can be only one Slave transmitting data. The figure below shows an example of an SPI controller in Slave Mode interfacing in a two-Slave environment with another SPI controller in Master Mode. In the multi-slave environment, all Slaves not selected by the Master through proper CS assertion, tristate their MISO lines.

Figure 9.32 Multi-Slave Environment



9.9.2.2.2 Packed and Unpacked Modes

SPI transactions can be carried out in packets. In particular, multiple packets (i.e., Packed) or a single packet (i.e., Unpacked) can be put in a Tx/Rx FIFO word in transferring.

In the SPI_COMMAND_0 Register, the PACKED field specifies whether the packets are packed or unpacked, and the BIT_LENGTH field specifies the packet size, i.e., how many bits in a packet.

The Packed and Unpacked Mode are available for both PIO and DMA Mode.

Packed Mode (SPI_COMMAND_0.PACKED == 1):

- If (BIT_LENGTH == 03h), the packet size is 4 bits; and each Tx/Rx FIFO word can hold up to eight such packets.
- If (BIT_LENGTH == 07h), the packet size is 8 bits; and each Tx/Rx FIFO word can hold up to four such packets.
- If (BIT_LENGTH == 0fh), the packet size is 16 bits; and each Tx/Rx FIFO word can hold up to two such packets.
- If (BIT_LENGTH == 1fh), the packet size is 32 bits; and each Tx/Rx FIFO word holds exactly one such packet.

In the Packed Mode, if the packet size is 4 bits and the number of packets is not a multiple of 8, or if the packet size is 8 bits and the number of packets is not a multiple of 4, or if the packet size is 16 bits and the number of packets is not a multiple of 2, then the last word in the Tx FIFO contains only the residual number (after dividing by 8, 4, or 2) of packets together with some number (28/24/20/16/12/8/4 if packet size is 4, 24/16/8 if Packet size is 8, or 16 if packet size is 16) of irrelevant random bits originally in the last Tx FIFO word. These irrelevant random bits will be ignored and not transmitted by the controller. In other words, the controller stops transmitting after finishing the last packet (of the not fully packed Tx FIFO word). On the receiver side, receiving also stops after the last packet, thus leaving the remaining number (28/24/20/16/12/8/4 if Packet size is 4, 24/16/8 if packet size is 8, or 16 if packet size is 16) of bits in the last Rx FIFO word untouched or irrelevant for the received packet.

Packets are placed in the Tx/Rx FIFO word Least-Significant-Bit-justified.

Unpacked Mode (SPI_COMMAND_0.PACKED == 0):

- Packet size is (SPI_COMMAND_0.BIT_LENGTH + 1)

In the Unpacked Mode, there is no packing of packets in the Tx/Rx FIFO word. It uses exactly one Tx/Rx FIFO word for one packet transmitting/receiving regardless of Packet sizes, thus leaving some bits unused. Unpacked Mode is always used for transmitting/receiving of all transfer sizes smaller than 32 bits.

Packets are placed in the Tx/Rx FIFO word Least-Significant-Bit-justified.

Byte and Bit Endianness

A packet can be transmitted or received in Little Endian or Big Endian format at both the Byte and Bit level as defined by the En_LE_Byte and En_LE_Bit field in the SPI_COMMAND_0 Register.

If (En_LE_Byte == 1), the bytes in a Packet are arranged in Big Endian; otherwise, Little Endian.
If (En_LE_Bit == 1), the bits in a byte are arranged in Little Endian; otherwise, Big Endian.

The table below explains how a 48-bit data ([47:0]) in three 16-bit Packets ({P2[15:0], P1[15:0], P0[15:0]}) is arranged according to the settings of En_LE_Byte and En_LE_Bit.

Table 9.47 Byte/Bit-Endian Arrangement

En_LE_Byte	En_LE_Bit	Byte/Bit-Endian Arrangement
0	0	P2([8:15],[0:7]), P1([8:15],[0:7]), P0([8:15],[0:7])
0	1	P2([15:8],[7:0]), P1([15:8],[7:0]), P0([15:8],[7:0])
1	0	P2([0:7],[8:15]), P1([0:7],[8:15]), P0([0:7],[8:15])
1	1	P2([7:0],[15:8]), P1([7:0],[15:8]), P0([7:0],[15:8])

The following examples explain the effects of En_LE_Byte and En-LE_Bit with Unpacked and Packed Mode.

Unpacked Mode
Consider a 24-bit data ([23:0]) as three 8-bit Packets ({B2[7:0], B1[7:0], B0[7:0]} for Tx/Rx.

- Tx: The 24-bit data ([23:0]) is stored in a Tx FIFO word (Column C) according to the settings of En_LE_Byte and En_LE_Bit, then transmitted on the MOSI line Least Significant Bit first (Column D).

Table 9.48 Transmission of Unpacked Data

A	B	C	D
En_LE_Byte	En_LE_Bit	Byte/Bit-Endian-Arranged Data in Tx FIFO Word for Tx	Data Bit Sequence Tx'd on MOSI Line (Right-most Bit First)
0	0	{8'bx,[16:23],[8:15],[0:7]}	{[16:23],[8:15],[0:7]}
0	1	{8'bx,[23:16],[15:8],[7:0]}	{[23:16],[15:8],[7:0]}
1	0	{8'bx,[0:7],[8:15],[16:23]}	{[0:7],[8:15],[16:23]}
1	1	{8'bx,[7:0],[15:8],[23:16]}	{[7:0],[15:8],[23:16]}

- Rx: Bits are received in the same sequence they were transmitted (Column C), then placed in a Rx FIFO word Least-Significant-Bit-justified (Column D)

Table 9.49 Receiving of Unpacked Data

A	B	C	D
En_LE_Byte	En_LE_Bit	Data Bit Sequence Rx'd from MISO Line (Right-most Bit First)	Data Arranged in Rx FIFO Word after Being Rx'd
0	0	{[16:23],[8:15],[0:7]}	{8'bx,[16:23],[8:15],[0:7]}
0	1	{[23:16],[15:8],[7:0]}	{8'bx,[23:16],[15:8],[7:0]}
1	0	{[0:7],[8:15],[16:23]}	{8'bx,[0:7],[8:15],[16:23]}
1	1	{[7:0],[15:8],[23:16]}	{8'bx,[7:0],[15:8],[23:16]}

Packed Mode
Consider a 48-bit data ([47:0]) as 3 16-bit Packets ({P2[15:0], P1[15:0], P0[15:0]}) for Tx/Rx.

- Tx: The 48-bit data ([47:0] is stored in two Tx FIFO words (Column C) according to the settings of En_LE_Byte and En_LE_Bit, then transmitted on the MOSI line Least Significant Bit first (Column D).

Table 9.50 Transmission of Packed Data

A	B	C	D
En_LE_Byte	**En_LE_Bit**	**Byte/Bit-Endian-Arranged Data in Tx FIFO Words for Tx**	**Data Bit Sequence Tx'd on MOSI Line (Right-most Bit First)**
0	0	{16'bx,[40:47],[32:39]}, {[24:31],[16:23],[8:15],[0:7]}	{[40:47],[32:39],[24:31],[16:23],[8:15],[0:7]}
0	1	{16'bx,[47:40],[39:32]}, {[31:24],[23:16],[15:8],[7:0]}	{[47:40],[39:32],[31:24],[23:16],[15:8],[7:0]}
1	0	{16'bx,[32:39],[40:47]}, {[16:23],[24:31],[0:7],[8:15]}	{[32:39],[40:47],[16:23],[24:31],[0:7],[8:15]}
1	1	{16'bx,[39:32],[47:40]}, {[23:16],[31:24],[7:0],[15:8]}	{[39:32],[47:40],[23:16],[31:24],[7:0],[15:8]}

- Rx: Bits are received in the same sequence they were transmitted (Column C), then placed in two Rx FIFO words Least-Significant-Bit-justified (Column D)

Table 9.51 Receiving of Packed Data

A	B	C	D
En_LE_Byte	**En_LE_Bit**	**Data Bit Sequence Rx'd from MISO Line (Right-most Bit First)**	**Data Arranged in Rx FIFO Words after Being Rx'd**
0	0	{[40:47],[32:39],[24:31],[16:23],[8:15],[0:7]}	{16'bx,[40:47],[32:39]}, {[24:31],[16:23],[8:15],[0:7]}
0	1	{[47:40],[39:32],[31;24],[23:16],[15:8],[7:0]}	{16'bx,[47:40],[39:32]}, {[31:24],[23:16],[15:8],[7:0]}
1	0	{[32:39],[40:47],[16:23],[24:31],[0:7],[8:15]}	{16'bx,[32:39],[40:47]}, {[16:23],[24:31],[0:7],[8:15]}
1	1	{[39:32],[47:40],[23:16],[31:24],[7:0],[15:8]}	{16'bx,[39:32],[47:40]}, {[23:16],[31:24],[7:0],[15:8]}

9.9.2.2.3 PIO and DMA Modes

PIO and DMA are the two main operational Modes of the SPI controller. When the number of FIFO words needed to hold the total number of packets for transmitting/receiving is less than or equal to 64 (depth of the Tx/Rx FIFO) the PIO Mode is used. Otherwise, the DMA Mode is used. In other words, the transfer size is limited to 64 (the Tx/Rx FIFO depth) 32-bit words worth of Packed/Unpacked Packets for the PIO Mode, and 2^{16} (64 Ki, as indicated by SPI_DMA_BLK_SIZE_0.DMA_BLOCK_SIZE) 32-bit words, or 256 KiB worth of Packed/Unpacked Packets for the DMA Mode. For data blocks with sizes larger than the limits (for PIO or DMA Mode), multiple (PIO or DMA) transfers will be used to complete transferring such data blocks as in Continuous Mode. In either PIO or DMA Mode, the SPI controller sends an Interrupt to the processor at the end of each transfer.

For details of the DMA operations, refer to the General-Purpose Direct Memory Access (GP-DMA) chapter.

Continuous Mode

Continuous Mode means:

- In Master Mode, the SPI Controller can transfer data continuously with an external device, unless Interrupted by software.
- In Slave Mode, the SPI Controller can transfer data continuously with an external device, unless Interrupted by software or an external Master deasserts chip select (CS). In the CS deassert case, the SPI controller can generate an Interrupt based on Interrupt mask bit.
- This mode can be enabled only in DMA mode, not in PIO mode. The SPI controller disregards the DMA_BLOCK_SIZE Register in this mode.

SPI Controller Configured as Master

When the SPI Controller is configured as a Master, it can transfer any amount of data unless software disables the SPI Controller by clearing (write "1" to clear) the DMA_EN bit in the SPI_DMA_CTL_0 Register. The SPI Controller generates CS and SCK accordingly throughout the data transfer. Along with the SPI Controller, the GPC DMA also needs to be configured in Continuous Mode. Refer to the General Purpose DMA chapter of this TRM for more information.

- Errors and Interrupt generation:
  In Master Continuous Mode, the SPI controller will not generate Tx FIFO Underflow or Rx FIFO Overflow errors. These errors generally occur in Slave mode when the GPC DMA does not write data into the Tx FIFO or does not read data from the Rx FIFO fast enough due to system bandwidth/arbitration issues. In such cases, the SPI Controller keeps waiting for the GPC DMA to write data into the Tx FIFO or read data from the Rx FIFO. During such waiting periods (Idle periods), the SPI Controller stops generating SCK but keeps CS asserted. Once data is available in the Tx FIFO or enough space is available in Rx FIFO, the SPI Controller resumes the data transfer and generates SCK, continuing the CS in the asserted state.
- Disabling the SPI Controller in Master Continuous Mode:
  In Master Continuous Mode, the SPI Controller can be disabled only by clearing (write "1" to clear) the DMA_EN bit by software. The SPI Controller will not be disabled in any other way by hardware.
- Generally, software is recommended to clear (write "1" to clear) the DMA_EN bit when the Controller is in the Idle state.
- If software clears the DMA_EN bit during the Active state (when a transaction is in process, i.e., CS is asserted and SCK is present), the SPI Controller writes the last received partial/full word (whatever is present in the shift Register) up to the point before disabling the DMA_EN bit into the Rx FIFO and increments the BLK_CNT counter in the SPI_TRANSFER_STATUS_0 Register.

SPI Controller Configured as Slave

When the SPI Controller is configured as a Slave, it can transfer any amount of data unless:

- Software disables the SPI Controller by clearing the DMA_EN bit (write "1" to clear) (or)
- External SPI Master Device deasserts Chip Select.
- External SPI Master stops sending SCK for SLV_IDLE_COUNT number of clocks.

9.9.2.2.4 Errors and Interrupt Generation

In Slave Continuous Mode, the SPI controller will generate Tx FIFO Underflow or Rx FIFO Overflow errors. These errors generally occur when GPC DMA does not write data into the Tx FIFO or does not read data from Rx FIFO fast enough due to system bandwidth/arbitration issues. In such cases, the SPI Controller generates an error condition, sets the TX_FIFO_UNF or RX_FIFO_OVF bits, and generates an Interrupt. If an Overflow happens, data in the Rx FIFO will not be corrupted.

9.9.2.2.5 Disabling the SPI Controller in Slave Continuous Mode

In Slave Continuous Mode, the SPI Controller can be disabled by software clearing the DMA_EN bit.

Generally, software is recommended to clear the DMA_EN bit when the SPI Controller is in the Idle state (the Idle state is defined as when there is no input SCK to the SPI Slave from the External Master device and/or CS is inactive).

If software clears the DMA_EN bit during the Active state (the Active state is defined as when there is an input SCK to the SPI Slave from the External Master device and CS is active), the SPI Controller will write the last received partial/full word (whatever is present in shift Register) into the Rx FIFO and increments BLK_CNT counter in the SPI_TRANSFER_STATUS_0 Register.

9.9.2.2.6 Disabling the SPI Controller by an External Master Device

When the SPI Controller is configured in Slave Continuous Mode, the External Master can start a data transfer by generating CS and SCK to the Slave SPI. Once the External Master finishes the data transfer (Tx, Rx, or both), it will stop generating SCK and then deassert CS. The SPI Slave treats this as an end of transaction and generates an Interrupt along with clearing the DMA_EN bit. The SPI Controller will write the last received partial/full word (whatever is present in the shift Register before CS is deasserted) into the Rx FIFO and increments the BLK_CNT counter in the SPI_TRANSFER_STATUS_0 Register. This will be true even if CS is deasserted in an unaligned word boundary (i.e. CS deassertion can happen only at a packet boundary).

In Slave Error cases (RX_FIFO_OVF), the BLK_CNT counter in the SPI_TRANSFER_STATUS_0 Register is incremented first and then the Rx FIFO is written. So when RX_FIFO_OVF happens, the BLK_CNT counter in the SPI_TRANSFER_STATUS_0 Register also shows 1 word extra compared to what is actually present in the Rx FIFO.

In Non-continuous Mode, the SPI Controller does not generate a CS_DEASSERT Interrupt at the very end when the actual programmed blk_size is done. That time CS is deasserted by the Master normally. Software will get a completion Interrupt in this case.

CS_INACTIVE Bit in SPI FIFO Control/Status Register (SPI_FIFO_STATUS_0)

In the CS deassert case, the SPI controller can generate Interrupts, provided the Interrupt Mask bit is set Low (Interrupt enabled) by software and (SPI_FIFO_STATUS_0.CS_INACTIVE == 1) indicating CS is deasserted by an External Master. Software needs to clear this bit while addressing the Interrupt. On the other hand, if the Interrupt Mask bit is set High (Interrupt not enabled), SPI_FIFO_STATUS_0.CS_INACTIVE may still be set High by the SPI Controller but will not generate an Interrupt.

- If the CS deassert Interrupt mask is set Low (i.e., the Interrupt is enabled), the SPI Slave Controller generates an Interrupt when all the data in the Rx shift Register is written into the Rx FIFO.
- Whenever external master deasserts CS, SPI slave will always set the CS_INACTIVE status bit and generate interrupt based on interrupt mask. Pause is a feature which when set will not terminate the transfer in case of CS deassertion by external master.
- In both the above cases, CS deassertion Interrupt generation is independent of the Clock provided by the External Master.

FRAME_END in SPI FIFO Control/Status Register

In Slave Continuous Mode, if SCK is not received for the number of clocks specified in the SLV_IDLE_COUNT field and CS is asserted, then the continuous mode is terminated and this status bit is set. Continuous Mode is terminated if the condition is met. In this case, an Interrupt is generated.

Software has two use cases:

1. In Slave Continuous mode, software mainly will use the pause feature (set PAUSE bit to "1"), because the use case is such that there will be continuous audio streaming and it is not desired for the Slave to be stopped based on CS deassert.
2. In Slave non-continuous mode, where the use case is based on variable length data coming in from the external Master, software will use the non-pause feature (enable the CS deassert Interrupt) so that software can read the received chunk of data based on the CS deassert Interrupt with frame_end remaining deasserted.

In the Non-Pause case, continuous mode termination can happen with CS deasserted or DMA disabled (DMA_EN = 0).In the Pause case, continuous mode termination can happen only with DMA disable. Until the DMA is disabled, it is the same continuous mode transaction and continues forever until the DMA is disabled.

9.9.2.3 Error Conditions

When the SPI controller is configured as a Master, the following error scenarios can happen:

- Tx FIFO Overflow

  Tx FIFO overflow occurs when the DMA writes into the Tx FIFO when it is full. In this case, the SPI controller terminates the transaction by setting the PIO/DMA_EN bit to "0," and flags the error by setting the TX_FIFO_OVF bit in the SPI FIFO Status Register to "1." The SPI controller generates an Interrupt if the TX_FIFO_OVF_INTR_MASK bit in the SPI_INTR_MASK_0 register is turned off (thus, allowing the Interrupt). Software must remove the cause of Tx FIFO Overflow and clear the TX_FIFO_OVF bit by writing "1" to it. To start a new transaction, software must flush the Tx FIFO by writing "1" to the TX_FIFO_FLUSH bit in the SPI FIFO Status Register.

- Rx FIFO Overflow

  Rx FIFO overflow occurs when the DMA writes into the Rx FIFO when it is full. In this case, the SPI controller terminates the transaction by setting the PIO/DMA_EN bit to "0," and flags the error by setting the RX_FIFO_OVF bit in the SPI FIFO Status Register to "1." The SPI controller generates an Interrupt if the RX_FIFO_OVF_INTR_MASK bit in the SPI_INTR_MASK_0 register is turned off (thus, allowing the Interrupt). Software must remove the cause of Rx FIFO Overflow and clear the RX_FIFO_OVF bit by writing "1" to it. To start a new transaction, software must flush the Rx FIFO by writing "1" to the RX_FIFO_FLUSH bit in the SPI FIFO Status Register.

- Tx FIFO Underflow

  Tx FIFO Underflow occurs when the DMA reads from the Tx FIFO when it is empty. In this case, the SPI controller terminates the transaction by setting the PIO/DMA_EN bit to "0," and flags the error by setting the TX_FIFO_UNF bit in the SPI FIFO Status Register to "1." The SPI controller generates an Interrupt if the TX_FIFO_UNF_INTR_MASK bit in the SPI_INTR_MASK_0 register is turned off (thus, allowing the Interrupt). Software must remove the cause of Rx FIFO Overflow and clear the TX_FIFO_UNF bit by writing "1" to it before a new transaction can start.

- Rx FIFO Underflow

  Rx FIFO Underflow occurs when the DMA reads from the Rx FIFO when it is empty. In this case, the SPI controller terminates the transaction by setting the PIO/DMA_EN bit to "0," and flags the error by setting the RX_FIFO_UNF bit in the SPI FIFO Status Register to "1." The SPI controller generates an Interrupt if the RX_FIFO_UNF_INTR_MASK bit in the SPI_INTR_MASK_0 register is turned off (thus, allowing the Interrupt). Software must remove the cause of Rx FIFO Overflow and clear the TX_FIFO_UNF bit by writing "1" to it before a new transaction can start.

Note: In Master Mode, all the errors above can happen only when the CPU/GPC_DMA reads/writes an empty/full FIFO. The SPI controller will never write/read a full/empty FIFO. Instead the controller

will pause (stops sending clocks to the Slave with chip select being in an active state) whenever the FIFOs are full/empty.

When the SPI controller is configured as a Slave, the following error scenarios can happen:

- Tx FIFO Overflow

  Tx FIFO Overflow happens when the CPU/GPC_DMA or the SPI controller writes into the Tx FIFO when it is full. In this case, the SPI controller ends the transaction by setting the PIO/DMA_EN bit to "0," and flag the error bit along with TX_FIFO_OVF bit in the status Register. The SPI controller generates an Interrupt if Interrupts are enabled. Software has to clear the error bit and TX_FIFO_OVF bit. To start a new transaction, software has to disable the Master from sending clocks and then flush the FIFOs by writing to the TX_FIFO_FLUSH/RX_FIFO_FLUSH bits in the SPI FIFO Control/Status Register.

- Rx FIFO Overflow

  Rx FIFO Overflow happens when the CPU/GPC_DMA or the SPI controller writes into Rx FIFO when it is full in BOTH_EN Mode. In this case, the SPI controller ends the transaction by setting the PIO/DMA_EN bit to "0," and flag the error bit along with RX_FIFO_OVF bit in the status Register. The SPI controller generates an Interrupt if Interrupts are enabled. Software has to clear the error bit and RX_FIFO_OVF bit along with the Interrupt bit. To start a new transaction, software has to disable the Master from sending clocks and flush the FIFOs by writing to the TX_FIFO_FLUSH/RX_FIFO_FLUSH bits in the SPI FIFO Control/Status Register.

- Tx FIFO Underflow

  Tx FIFO Underflow happens when the CPU/GPC_DMA or the SPI controller reads from Tx FIFO when it is empty. In this case, the SPI controller ends the transaction by setting the PIO/DMA_EN bit to "0," and flag the error bit along with the TX_FIFO_UNF bit in the SPI FIFO Control/Status Register. The SPI controller generates an Interrupt if Interrupts are enabled. Software has to clear the error bit and TX_FIFO_UNF bit along with the Interrupt bit. To start a new transaction, software has to disable the Master from sending clocks and flush the FIFOs by writing to the TX_FIFO_FLUSH/RX_FIFO_FLUSH bits in the SPI FIFO Control/Status Register.

- Rx FIFO Underflow

  Rx FIFO Underflow happens when the CPU/GPC_DMA or the SPI controller reads from the Rx FIFO in BOTH_EN Mode. In this case, the SPI controller ends the transaction by setting the PIO/DMA_EN bit to "0," and flag the error bit along with the RX_FIFO_UNF bit in the SPI FIFO Control/Status Register. The SPI controller generates an Interrupt if Interrupts are enabled. Software has to clear the error bit and RX_FIFO_UNF bit along with the Interrupt bit. To start a new transaction, software has to disable the Master from sending clocks and flush the FIFOs by writing to the TX_FIFO_FLUSH/RX_FIFO_FLUSH bits in the SPI FIFO Control/Status Register.

- CS_inactive

When the SPI controller is configured as a Slave and CS_ACTIVE_BETWEEN_PACKETS_n in the SPI_TIMING_REG2_0 Register is set, if an external Master deasserts the chip select in the middle of a transaction, the SPI controller ends the transaction by setting the PIO/DMA_EN bit to "0," and sets the CS_INACTIVE bit in the SPI FIFO Control/Status Register. The SPI controller generates an Interrupt if Interrupts are enabled. Software has to clear the CS_INACTIVE bit. To start a new transaction, software has to disable the Master from sending clocks and flush the FIFOs by writing to the TX_FIFO_FLUSH/RX_FIFO_FLUSH bits in the SPI FIFO Control/Status Register.

It is software's responsibility to read the data that is left out of the Rx FIFO for CS inactive. For example, if the Interrupt trigger is set for four words and external Master has taken the CS off in between the transfer, and at that point of time the Rx FIFO has only two words, it is software's responsibility to read out those two words or flush the FIFO.

- Frame_End

  When the SPI controller is configured as a Slave and DMA-continuous and CS_ACTIVE_BETWEEN_PACKETS_n in the SPI_TIMING_REG2_0 Register is set, if an external Master stops sending clocks for more than Slave_idle_clock_count programmed in the Register, the SPI controller ends the transaction by setting the DMA bit to "0," and sets the FRAME_END bit in the SPI FIFO Control/Status Register. The SPI controller generates an Interrupt if Interrupts are enabled. Software has to clear the FRAME_END bit along with the Interrupt bit. To start a new transaction, software has to disable the Master from sending clocks and flush the FIFOs by writing to the TX_FIFO_FLUSH/RX_FIFO_FLUSH bits in the SPI FIFO Control/Status Register. If the Master resumes the clk within "Slave_idle_clock_count", the SPI controller will also pause and resume the transaction whenever ext_clk is available.

9.9.2.4 Programming Guidelines

There are two basic modes of operation: DMA and PIO modes. It is required that software sets up all parameters in the following registers before enabling the SPI_COMMAND_0.PIO bit for any of these modes.

- SPI_COMMAND_0
- SPI_COMMAND2_0
- SPI_TIMING_REG1_0
- SPI_TIMING_REG2_0
- SPI_TRANSFER_STATUS_0
- SPI_FIFO_STATUS_0
- SPI_DMA_CTL_0
- SPI_DMA_BLK_SIZE_0

9.9.2.4.1 PIO Mode

This mode is enabled by writing "1" to the PIO bit in the SPI Command1 Register. In this mode, the SPI controller transmits/receives as many packets as configured by software in the SPI Block Size Register.

PIO Mode has the same features as DMA Mode. The difference between PIO Mode and DMA Mode is that in PIO Mode, the maximum number of packets that can be transmitted or received is less than or equal to 64.

9.9.2.4.2 DMA Mode

This mode is enabled by writing "1" to the DMA bit in the SPI DMA Control Register. In this mode, the SPI controller transmits or receives the number of packets as indicated by the BLOCK_SIZE field in the SPI Block Size Register.

If the PACKED bit is set and BIT_LEN is set to 7, then all FIFO words contain four packets to transfer (transmit or receive). Packets are transferred as per the En_LE_Bit and En_LE_Byte bit configurations (see the En_LE_Bit and En_LE_Byte Modes section), with packet 0 in byte 0 of the FIFO and packet 3 in byte 3 of the FIFO.

In Unpacked Mode, if BIT_LEN is set to N, each packet will consist of (N + 1) bits. These bits will be transmitted/received in the Tx FIFO/Rx FIFO as per the En_LE_Bit and En_LE_Byte bit configurations (see the En_LE_Bit and En_LE_Byte Modes section). Any remaining bits in the FIFO will be ignored by the hardware. The maximum packet length is 32, which can be selected by setting BIT_LEN to "31". In this case, all data bits in the FIFO contain valid packet data.

A DMA request will be generated to GPC_DMA in this mode depending on the setting of Tx_TRIG and Rx_TRIG. If transmits are enabled, setting Tx_TRIG to "00" generates a Tx DMA request whenever the Tx FIFO has one word of space available (if not full). Setting Tx_TRIG to "01" generates a Tx DMA request whenever the Tx FIFO has four words of space available. If receives are enabled, setting Rx_TRIG to "00" will generate an Rx DMA request whenever the Rx FIFO has one word of data available (is not empty). Setting Rx_TRIG to "01" generates an Rx DMA request whenever the Rx FIFO has four words of data available.

9.9.2.4.3 Interrupt Generation

The SPI controller generates an Interrupt to the processor at the end of a transfer in PIO Mode or when an error is detected in PIO or DMA Mode.

An Interrupt is generated whenever RDY or one of the FIFO status bits in the SPI FIFO Control/ Status Register is set by the hardware.

During transmits, when the SPI controller is configured as a Slave, if software/GPC DMA cannot fill the transmit FIFO fast enough, hardware will set the TX_FIFO_UNF bit and an Underflow condition

is generated. If software tries to write to a full Tx FIFO, hardware sets the TX_FIFO_OVF bit as an indication that software attempted to Overflow the Tx FIFO. Hardware makes sure that the Overflowing data is never written to the Tx FIFO.

During receives, when the SPI controller is configured as a Slave, if software/GPC DMA cannot read the receive FIFO fast enough, hardware will set the RX_FIFO_OVF bit and an Overflow condition is generated. However, if software tries to read from an empty Rx FIFO, hardware sets the RXF_UNF bit as an indication that software attempted to Underflow the Rx FIFO.

The Interrupt can be cleared by writing a "1" to the source of the Interrupt. If the Interrupt is generated by assertion of RDY, then writing a "1" to the RDY bit clears the Interrupt. If the Interrupt is generated by assertion of TX_FIFO_OVF, then writing a "1" to the TXF_OVF bit clears the Interrupt. If the Interrupt is generated by assertion of RX_FIFO_UNF, then writing a "1" to RXF_UNF bit clears the Interrupt.

> **Guideline for DMA use cases:** In DMA Mode, if Interrupts are enabled, the SPI software driver has to deal with three Interrupts: one from the SPI controller and two from the Tx and Rx GPC DMA channels. This might make it complicated for the driver and also stresses the system. In such cases, software can use the Rx DMA Interrupt as the final one at which point all hardware activities can be assumed to be complete. But, in variable transfer lengths (continuous mode), software has to rely on the controller's Interrupts.

9.9.2.4.4 Clock Initialization and Control

The SPI controller runs at 50 MHz at its interface to external SPI devices. The internal SPI controller clock (spi_clk) should run at the same frequency as the outgoing Interface Clock (SCK_Out) in Master Mode. In Slave Mode, however, the internal SPI interface clock frequency has to be 50% greater than the external clock (SCK_in).

Note: The ratio between the Slave core clock and the Slave interface clock must be maintained at 1.5x. For example if the Slave interface clock is running at 10 MHz, the Slave core clock must not be less than 10x 1.5 MHz.

Controller Enable

The SPI controller can be enabled in two ways:

- Write (CLK_RST_CONTROLLER_CLK_OUT_ENB_SPI<i>_0.CLK_ENB_SPI<i> = 1), where i = 1, 2, 3.
- Write (CLK_RST_CONTROLLER_CLK_OUT_ENB_SPI<i>_SET_0.SET_CLK_ENB_SPI<i> = 1), where i = 1, 2, 3.

Correspondingly, to disable the SPI controller:

- Write (CLK_RST_CONTROLLER_CLK_OUT_ENB_SPI<i>_0.CLK_ENB_SPI<i> = 0), where i = 1, 2, 3.

- Write (CLK_RST_CONTROLLER_CLK_OUT_ENB_SPI<i>_CLR_0.CLR_CLK_ENB_SPI<i> = 1), where i = 1, 2, 3.

In addition, the SPI controller can select its clock source with proper setup:

- Write (CLK_RST_CONTROLLER_CLK_SOURCE_SPI<i>_0.SPI<i>_CLK_SRC = n), where i = 1, 2, 3, and n = 0, 6, 7.

The selected clock is further divided by a factor m set by the following:

- Write (CLK_RST_CONTROLLER_CLK_SOURCE_SPI<i>_0.SPI<i>_CLK_DIVISOR = (2m-2)), where i = 1, 2, 3.

See the Clock Controller and Reset chapter for more information on controller enable and clock configuration.

Controller Reset

The SPI controller can be reset in two different ways:

- Write (CLK_RST_CONTROLLER_RST_DEV_SPI<i>_0.SWR_SPI<i>_RST = 1), where i = 1, 2, 3.
- Write (CLK_RST_CONTROLLER_RST_DEV_SPI<i>_SET_0.SET_SWR_SPI<i>_RST = 1), where i = 1, 2, 3.

Correspondingly, to bring the SPI controller out of Reset mode:

- Write (CLK_RST_CONTROLLER_RST_DEV_SPI<i>_0.SWR_SPI<i>_RST = 0), where i = 1, 2, 3.
- Write (CLK_RST_CONTROLLER_RST_DEV_SPI<i>_CLR_0.CLR_SWR_SPI<i>_RST = 1), where i = 1, 2, 3.

See the Reset chapter for more information on SPI controller reset.

9.9.2.4.5 Slave Mode GPIO Synchronization

Because SPI does not support flow control, the GPIO is required to have the proper communication to avoid any clock loss from the Master. GPIO can be used in the following way for proper synchronization:

- SPI Slave client software writes to the Config Registers of the SPI controller and make the SPI Slave ready.
- After that, the Slave client software toggles GPIO to inform the Master that the Slave is ready.
- Then the SPI Master sends SCK to the Slave device.

If there is no GPIO to tell the Master that the Slave is ready, there is a chance to lose clock/data.

Guidelines

This section provides guidelines for programming the SPI controller:

- Program all the required Register fields (no particular order is required except for the PIO/DMA bit, which has to be programmed last):
  - Clock Mode
  - Packed/Unpacked Mode
  - Tx_EN/Rx_EN
  - BOTH_EN_BIT
  - En_LE_Bit/En_LE_Byte
  - BIT_LEN and BLOCK_SIZE values
  - CS_SW_HW (software based or hardware based)
    - If CS_SW_HW is set, the value on CS_SW_VAL will be driven out.
    - When CS hardware is used program the setup and hold values in the CS Timing Register.
  - Program the DMA Trig values appropriately, if DMA Mode is used
  - Enable Interrupts if PIO Mode is used
- Program PIO or DMA to indicate the start of transfer.

Once the software enables PIO/DMA, no required bits can be changed until the end of transfer except for PIO/DMA bit. Clearing the PIO/DMA bit ends the transaction. In case the SPI controller is configured to Receive Mode and software clears the PIO/DMA bit, then the partial data which the controller received until then will be written into the FIFO. This is true both for Master and Slave.

In Slave Mode after PIO or DMA is set, hardware first waits for the CS to go Low by one of the four Masters. Then on reception of SCK, transfer begins. If CS or SCK is removed by the Master in the middle of a transfer, software must terminate the transaction by clearing the PIO/DMA bit. In case of Rx, hardware then writes the last received packet (incomplete packet) to the Rx FIFO. If software does not clear the PIO/DMA bit, the SPI controller will keep waiting for the SCK from the Master. If SCK resumes later, the transaction continues from where it paused earlier.

If any error conditions occur, hardware will set the corresponding status bits in the SPI FIFO Control/Status Register and stop the transfer. If IE is enabled, an Interrupt is generated. Software has to clear the source of the Interrupt by writing a "1" to it, after the completion of ISR.

Special Guidelines for Slave Mode

Software should not program the controller Register when the interface clock is toggling. Software can disable the external clock input before programming the Registers and then re-enable it once all the Registers bits are programmed.

One bit was added to the SPI controller (EXT_CLK_EN in Register SPI_MISC_0) to enable external Master clock. Software is required to enable this bit after all other Slave programming is done (after PIO/DMA bit is set). This bit works like a GATE for the external master clock.

- Software should not configure Slave Registers or apply reset when external Master clock is toggling. If needed, software should GATE the clock by writing "0" to this bit and enable back when configuration or reset is done.
- For a completely successful transfer, the Master starts the actual transfer (enable both CLK and CS) after at least five cycles after EXT_CLK_EN is enabled in the Slave.
- In all other scenarios where the Master started communication to Slave and EXT_CLK_EN is not enabled in advance, you may see data corruption.

Guidelines for Transferring Data with GPC-DMA

When the external device stops sending data after a (unknown) logical boundary, software programs GPCDMA for the maximum possible data transfer size based on the use case.

- SPI is programmed for 64B/16W Rx mode as trigger level.
- GPC-DMA channel is programmed for 64B (16 Words) MMIO burst size.
- GPC-DMA channel is programmed for 64B (16 Words) MC burst size (default value is in the specification, so there is no need to program explicitly).
- SPI generates an Interrupt to the CPU when all the packets are written to Rx FIFO (for Rx transfers) or all packets are read from Tx FIFO to send out (for Tx transfers).
- Based on this Interrupt, the software waits until the ongoing DMA transfer is complete by checking (RXFIFO_COUNT < BURST) or (DMAREQ_RX == Low) before disabling the DMA.
- If SPI has residual data in the buffer, the software reads it via PIO mode.

9.9.2.4.6 Programming Trimmers

The three programmable trimmers in the SPI controller are used only in Master Mode:

- SPI-Tx trimmer

  This trimmer is used in Master Tx mode to adjust/center the outgoing data with respect to the outgoing clock.

- SPI-Rx trimmer

  This trimmer is used in Master Rx mode to delay the loopback clock to the Rx shift Registers. This trimmer is used to adjust the Master SCK with respect to the SPI Slave device's Tx data.

9.9.3 SPI Registers

Refer to "Reading Register Tables" in the Introduction chapter for the register table protocol as well as recommendations for accessing registers.

There are three sets of the SPI controller registers, one for each SPI controller in the SoC. The register descriptions below give the offset of each register within the SPI controller's address range. The Base Addresses of the registers in these three SPI controllers are specified in the Address Map section of the TRM.

SPI_COMMAND_0

Programming sequence for basic data transmission. After the module-initialization (reset and clock programming), program the corresponding bits of the following registers, preferably in the same sequence:

For PIO Mode:

COMMAND: Set the type of transfer i.e., receive (bit 12), transmit (bit 11)

COMMAND: Set the bit_length (bits [4:0]), number of words (bits [9:5]), mode (mode 0, mode 1, mode 2 and mode 3) as below:

Mode 0 --> 00 // Clock is positive polarity and the data is latched-in on the positive edge of SCK.

Mode 1 --> 01 // Clock is positive polarity and the data is latched-in on the negative edge of SCK.

Mode 2 --> 10 // Clock is negative polarity and the data is latched-in on the negative edge of SCK.

Mode 3 --> 11 // Clock is negative polarity and the data is latched-in on the positive edge of SCK.

TX_FIFO: Write data into TXFIFO COMMAND: Enable the PIO Bit (bit 31).

For DMA mode:

COMMAND: Set the type of transfer i.e., receive (bit 12), transmit (bit 11)

COMMAND: Set the bit_length (bits [4:0]), and mode as described above. Note that the bit length to be set accordingly when packed mode (bit 5) is set in the COMMAND register.

Allowed bit_lengths in Packed Mode: BIT_LENGTH 0x03, 0x07, 0x0f, 0x1f

DMA_CTL: Set the DMA_BLOCK_SIZE (bits [15:0]), RX_TRIG (bits [20:19]), TX_TRIG (bits [16:15])

After setting all the parameters set DMA_EN(bit 31) to 1. The DMA request signals to the APBDMA controller are enabled by the spi COMMAND.Tx_EN and Rx_EN bits. Always program the DMA channels first. Enable the COMMAND.Tx_EN and Rx_EN bits last.

SPI Sub Block Command Register

Offset: 0x0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x43d00000 (0b0100,0011,1101,0000,0000,0xxx,x000,0000)

Bit	Reset	Description
31	0x0	**PIO**: Program 1 after all the other bits in the COMMAND2 and COMMAND are programmed to start the transfer. Hardware clears this bit automatically after the transfer is done. Clearing of the bit by Software will stop the Shifter and latch the partial data into buffer. **0** = STOP **1** = PIO
30	0x1	**M_S**: 1 = Master Mode (internal Clock) 0 = Slave Mode (external Clock) **0** = SLAVE **1** = MASTER
29:28	0x0	**MODE**: 11 = Mode3, 10 = Mode2, 01 = Mode1, 00 = Mode0 (def); When SPI is Master All four Modes are supported. When SPI is Slave Transmit supports Only Mode1 and Mode3 are supported. In Slave Receive all the Modes are supported. **0** = Mode0 **1** = Mode1 **2** = Mode2 **3** = Mode3
27:26	0x0	**CS_SEL**: 11 = chip select3, 10 = chip select2, 01 = chip select1, 00 = chip select0 (def) **0** = CS0 **1** = CS1 **2** = CS2 **3** = CS3
25	0x1	**CS_POL_INACTIVE3**: In Master or Slave Mode, the inactive value of external devices CS value which is connected to CS3 needs to be programmed. 1 = CS3 Inactive value of External device is high 0 = CS3 Inactive value of External device is low **0** = LOW **1** = HIGH
24	0x1	**CS_POL_INACTIVE2**: In Master or Slave Mode, the inactive value of external devices CS value which is connected to CS2 needs to be programmed. 1 = CS2 Inactive value of External device is high 0 = CS2 Inactive value of External device is low **0** = LOW **1** = HIGH
23	0x1	**CS_POL_INACTIVE1**: In Master or Slave Mode, the inactive value of external devices CS value which is connected to CS1 needs to be programmed. 1 = CS1 Inactive value of External device is high 0 = CS1 Inactive value of External device is low **0** = LOW **1** = HIGH

Bit	Reset	Description
22	0x1	**CS_POL_INACTIVE0**: In Master or Slave Mode, the inactive value of external devices CS value which is connected to CS0 needs to be programmed 1 = CS0 Inactive value of External device is high 0 = CS0 Inactive value of External device is low **0** = LOW **1** = HIGH
21	0x0	**CS_SW_HW**: Software control of SPI_CS signal in Master Mode. In Slave Mode this bit need not be programmed. 1 = CS controlled by software 0 = CS controlled by hardware **0** = HARDWARE **1** = SOFTWARE
20	0x1	**CS_SW_VAL**: When CS is controlled by Software this value is driven onto the selected CS line. 1 = CS is high 0 = CS is low **0** = LOW **1** = HIGH
19:18	0x0	**IDLE_SDA**: 11 = Pull High, 10 = Pull Low, 01 = Driven High, 00 = Driven Low **0** = DRIVE_LOW **1** = DRIVE_HIGH **2** = PULL_LOW **3** = PULL_HIGH
17	0x0	**BIDIR**: 1 = bi directional mode. 0 = Normal mode **0** = NORMAL **1** = BIDIR
16	0x0	**En_LE_Bit**: 1 = Enable Little Endian Bit. 0 = Disable Little Endian Bit **0** = LAST **1** = FIRST
15	0x0	**En_LE_Byte**: 1 = Enable Little Endian Byte. 0 = Disable Little Endian Byte **0** = LAST **1** = FIRST
14	0x0	**BOTH_EN_BIT**: 1 = both lines transmit/receive. 0 = one line transmit and other receive **0** = DISABLE **1** = ENABLE
13	0x0	**BOTH_EN_BYTE**: 1 = both lines transmit/receive. 0 = one line transmit and other receive **0** = DISABLE **1** = ENABLE
12	0x0	**Rx_EN**: Receive enable **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
11	0x0	**Tx_EN**: Transmit enable **0** = DISABLE **1** = ENABLE
6	0x0	**CS_GPIO**: This bit is used to control the outgoing CS lines. It can be used to mask the CS lines driven from controller. Controller will always drive CS_POL_INACTIVE* value to CS lines if this bit is set. 0 = Controller will drive CS lines based on CS_SW_HW (normal operation). 1 = Controller will drive CS_POL_INACTIVE* value. **0** = DISABLE **1** = ENABLE
5	0x0	**PACKED**: Packed mode enable bit. 1 = Packed mode is enabled. This is only valid if BIT_LENGTH in COMMAND register is set to 3, 7, 15, or 31. When enabled, all 32-bits of data in the FIFO contains valid data packets of either 4-bit, 8-bit, 16-bit, or 32-bit length. 0 = Packed mode is disabled. **0** = DISABLE **1** = ENABLE
4:0	0x0	**BIT_LENGTH**: 31 = Thirty Two bit Transfers (Max)

SPI_COMMAND2_0

Offset: 0x4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,0000,0000,0000)

Bit	Reset	Description
11:6	0x0	**Tx_Clk_TAP_DELAY**: Delays the clock going out to the external device with these tap values. Useful only in Master Mode.
5:0	0x0	**Rx_Clk_TAP_DELAY**: For master mode, it uses the tap values to delay the internal Rx clock, which is a loopback from Tx Outbound clock (before Tx Trimmer) For slave mode, it uses the tap values to delay the clock coming in from the external device (master)

SPI_TIMING_REG1_0

Offset: 0x8
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**CS_SETUP_3**: number of cycles for CS-3 Setup
27:24	0x0	**CS_HOLD_3**: number of cycles for CS-3 Hold
23:20	0x0	**CS_SETUP_2**: number of cycles for CS-2 Setup
19:16	0x0	**CS_HOLD_2**: number of cycles for CS-2 Hold
15:12	0x0	**CS_SETUP_1**: number of cycles for CS-1 Setup
11:8	0x0	**CS_HOLD_1**: number of cycles for CS-1 Hold
7:4	0x0	**CS_SETUP_0**: number of cycles for CS-0 Setup
3:0	0x0	**CS_HOLD_0**: number of cycles for CS-0 Hold

SPI_TIMING_REG2_0

Offset: 0xc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x20202020 (0bxx10,0000,xx10,0000,xx10,0000,xx10,0000)

Bit	Reset	Description
29	0x1	**CS_ACTIVE_BETWEEN_PACKETS_3**: Specifies if CS stays active in between two packets on CS-3 1 = CS active between two packets 0 = CS inactive between two packets
28:24	0x0	**CYCLES_BETWEEN_PACKETS_3**: Number of cycles in between packets for CS-3
21	0x1	**CS_ACTIVE_BETWEEN_PACKETS_2**: Specifies if CS stays active in between two packets on CS-2 1 = CS active between two packets 0 = CS inactive between two packets
20:16	0x0	**CYCLES_BETWEEN_PACKETS_2**: Number of cycles in between packets for CS-2

Bit	Reset	Description
13	0x1	**CS_ACTIVE_BETWEEN_PACKETS_1**: Specifies if CS stays active in between two packets on CS-1 1 = CS active between two packets 0 = CS inactive between two packets
12:8	0x0	**CYCLES_BETWEEN_PACKETS_1**: number of cycles in between packets for CS-1
5	0x1	**CS_ACTIVE_BETWEEN_PACKETS_0**: Specifies if CS stays active in between two packets on CS-0 1 = CS active between two packets 0 = CS inactive between two packets
4:0	0x0	**CYCLES_BETWEEN_PACKETS_0**: Number of cycles in between packets for CS-0

SPI_TRANSFER_STATUS_0 .

Offset: 0x10
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00ff0000 (0bx0xx,xxxx,1111,1111,0000,0000,0000,0000)

Bit	R/W	Reset	Description
30	RW	0x0	**RDY**: Ready bit. This bit is set to 1 at the end of every transfer and an interrupt is also generated if the corresponding interrupt enable is set in PIO/DMA Mode. Software writes a 1 to clear it. The interrupt is also cleared when this bit is cleared. **0** = NOT_READY **1** = READY
23:16	RW	0xff	**SLV_IDLE_CNT**: In Slave Continuous Mode if Sclk is not received for this number of cycles then the continuous mode is terminated and the status bit FRAME_END is set.
15:0	RO	0x0	**BLK_CNT**: Counts the number of Packets in a transaction (Tx or Rx) DMA/ PIO Mode.

SPI_FIFO_STATUS_0

Offset: 0x14
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00400005 (0b0000,0000,0100,0000,00xx,xx00,0000,0101)

Bit	R/W	Reset	Description
31	RW	0x0	**CS_INACTIVE**: 1 = CS is de-asserted by external Master in-between a transaction. Interrupt is generated in this case.
30	RW	0x0	**FRAME_END**: 1 = External clock is absent abruptly in-between a transaction.
29:23	RO	0x0	**RX_FIFO_FULL_COUNT**: Indicates the number of full slots in the receive FIFO
22:16	RO	0x40	**TX_FIFO_EMPTY_COUNT**: Indicates the number of empty slots in the transmit FIFO
15	RW	0x0	**RX_FIFO_FLUSH**: Flush the RX FIFO **0** = NOP **1** = FLUSH
14	RW	0x0	**TX_FIFO_FLUSH**: Flush the TX FIFO **0** = NOP **1** = FLUSH
9	RW	0x0	**CS_BOUNDARY_TIMEOUT_INTR**: Relevant only in case of Slave_RX_Continuous_Pause mode. 1 = Means there is at least one valid entry in CS Boundary Status FIFO. The valid entry is pushed into status FIFO when CS_BOUNDARY_TIMEOUT counter expires.
8	RW	0x0	**ERR**: Will be set to 1 by Hardware when Errors such as Underflow/ overflow occurs. Write 1 to clear the flag. **0** = OK **1** = ERROR
7	RW	0x0	**TX_FIFO_OVF**: TX FIFO Overflow **0** = OK **1** = ERROR
6	RW	0x0	**TX_FIFO_UNF**: TX FIFO Underflow **0** = OK **1** = ERROR
5	RW	0x0	**RX_FIFO_OVF**: RX FIFO Overflow **0** = OK **1** = ERROR
4	RW	0x0	**RX_FIFO_UNF**: RX FIFO Underflow **0** = OK **1** = ERROR
3	RO	0x0	**TX_FIFO_FULL**: TX FIFO Full **0** = NOT_FULL **1** = FULL

Bit	R/W	Reset	Description
2	RO	0x1	**TX_FIFO_EMPTY**: TX FIFO Empty **0** = NOT_EMPTY **1** = EMPTY
1	RO	0x0	**RX_FIFO_FULL**: RX FIFO Full **0** = NOT_FULL **1** = FULL
0	RO	0x1	**RX_FIFO_EMPTY**: RX FIFO Empty **0** = NOT_EMPTY **1** = EMPTY

SPI_TX_DATA_0

Offset: 0x18
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**Tx_BUFFER**

SPI_RX_DATA_0

Offset: 0x1c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**Rx_BUFFER**

SPI_DMA_CTL_0

Offset: 0x20
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b000x,xxxx,xxx0,0xx0,0xxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_EN**: 1 = DMA mode is enabled, 0 = DMA disabled **0** = DISABLE **1** = ENABLE
30	0x0	**CONT**: 1 = continuous mode is enabled, 0 = continuous mode disabled **0** = DISABLE **1** = ENABLE
29	0x0	**PAUSE**: This bit is mainly to be used in Slave Rx continuous mode or variable length receive with CS de-assert feature. 1 = Pause feature is enabled which means that CS de-assertion does not stop the controller and it will start receiving data again when CS comes back. 0 = Pause feature is disabled which means that CS de-assertion stops the controller. Software will have to re-enable the controller to start receiving the data. **0** = DISABLE **1** = ENABLE
20:19	0x0	**RX_TRIG**: Receive FIFO trigger level. 00: 1 word. DMA trigger is asserted whenever there is at least one word in the Rx FIFO. 01: 4 word. DMA trigger is asserted when there are at least four words in the Rx FIFO. 10: 8 word. DMA trigger is asserted when there are at least eight words in the Rx FIFO. 11: 16 word. DMA trigger is asserted when there are at least 16 words in the Rx FIFO. **0** = TRIG1 **1** = TRIG4 **2** = TRIG8 **3** = TRIG16
16:15	0x0	**TX_TRIG**: Transmit FIFO trigger level. 00: 1 word. DMA trigger is asserted whenever there is space for at least one word in the Tx FIFO. 01: 4 word. DMA trigger is asserted when there is space for at least four words in the Tx FIFO. 10: 8 word. DMA trigger is asserted when there is space for at least eight words in the Tx FIFO. 11: 16 word. DMA trigger is asserted when there is space for at least 16 words in the Tx FIFO. **0** = TRIG1 **1** = TRIG4 **2** = TRIG8 **3** = TRIG16

SPI_DMA_BLK_SIZE_0

Offset: 0x24
Read/Write: R/W

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,0000,0000,0000,0000)

Bit	Reset	Description
15:0	0x0	**DMA_BLOCK_SIZE**: N = N+1 packets

SPI_TX_FIFO_0

SPI Sub Block TX FIFO Buffer Register

Offset: 0x108
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TX_FIFO_REGISTER**

SPI_RX_FIFO_0

SPI Sub Block RX FIFO Buffer Register

Offset: 0x188
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**RX_FIFO_REGISTER**

SPI_INTR_MASK_0

Offset: 0x18c
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000000 (0b0000,000x,xxxx,xxxx,xxxx,xx0x,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**CS_INTR_MASK**: Interrupt enable mask bit for CS de-assert in Slave Mode. 1 = Disable interrupt generation if CS is de-asserted in the middle of transaction in Slave mode. 0 = Enable interrupt generation if CS is de-asserted in the middle of transaction in Slave mode. **0** = DISABLE **1** = ENABLE
30	0x0	**FRAME_END_INTR_MASK**: Interrupt enable mask bit for Frame End in Slave Mode. 1 = Disable interrupt generation if Frame End is set in the middle of transaction in Slave mode. 0 = Enable interrupt generation if Frame End is set in the middle of transaction in Slave mode. **0** = DISABLE **1** = ENABLE
29	0x0	**RDY_INTR_MASK**: Interrupt enable mask bit for RDY. 1 = Disable interrupt generation when RDY bit is asserted. 0 = Enable interrupt generation when RDY bit is asserted. **0** = DISABLE **1** = ENABLE
28	0x0	**TX_FIFO_OVF_INTR_MASK**: Interrupt enable mask bit for TX_FIFO_OVF. 1 = Disable interrupt generation when TX_FIFO_OVF is asserted. 0 = Enable interrupt generation when TX_FIFO_OVF is asserted. **0** = DISABLE **1** = ENABLE
27	0x0	**TX_FIFO_UNF_INTR_MASK**: Interrupt enable mask bit for TX_FIFO_UNF. 1 = Disable interrupt generation when TX_FIFO_UNF is asserted. 0 = Enable interrupt generation when TX_FIFO_UNF is asserted. **0** = DISABLE **1** = ENABLE
26	0x0	**RX_FIFO_OVF_INTR_MASK**: Interrupt enable mask bit for RX_FIFO_OVF. 1 = Disable interrupt generation when RX_FIFO_OVF is asserted. 0 = Enable interrupt generation when RX_FIFO_OVF is asserted. **0** = DISABLE **1** = ENABLE
25	0x0	**RX_FIFO_UNF_INTR_MASK**: Interrupt enable mask bit for RX_FIFO_UNF. 1 = Disable interrupt generation when RX_FIFO_UNF is asserted. 0 = Enable interrupt generation when RX_FIFO_UNF is asserted. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
9	0x0	**CS_BOUNDARY_TIMEOUT_INTR_MASK**: Interrupt enable mask bit for RX_FIFO_UNF. 1 = Disable interrupt generation when CS_BOUNDARY_TIMEOUT_INTR is asserted. 0 = Enable interrupt generation when CS_BOUNDARY_TIMEOUT_INTR is asserted. **0** = DISABLE **1** = ENABLE

SPI_SPARE_CTLR_0

Offset: 0x190
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0fff0000 (0b0000,1111,1111,1111,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0xf	**SPARE_CONTROL_REGISTER_BYTE4**
23:16	0xff	**SPARE_CONTROL_REGISTER_BYTE3**
15:8	0x0	**SPARE_CONTROL_REGISTER_BYTE2**: Need to Program bits 8, 9, and 10 along with Rx_Clk_TAP_DELAY in COMMAND2 Register to adjust clock delay on internal registers. Useful only in Master Mode.
7:0	0x0	**SPARE_CONTROL_REGISTER_BYTE1**

SPI_MISC_0

Offset: 0x194
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x80000000 (0b10xx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)
PROD: 0x00000000 (0b0xxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	PROD	Description
31	0x1	0x0	**CLKEN_OVERRIDE**: Override for spi_clk. Can be used to bypass the SLCG incase of issues. By default SLCG is not enabled. Software should turn on it by writing this bit to 0. 0: spi_clk is gated 1: spi_clk is not gated

Bit	Reset	PROD	Description
30	0x0	_NONE_	**EXT_CLK_EN**: This bit is used to gate the external master clk. In multi slave environment where master clock is connected to all slave, this clock is running even though master is communicating to other slave. Software should enable this once controller is out of reset and all the configuration is done which means slave is ready to receive data. During this time, if any valid data comes in, that will be lost. 0: ext_clk is gated. 1: ext_clk is not gated.

SPI_FATAL_INTR_EN_0

Offset: 0x198
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b00x0,000x,xxxx,xxxx,xxxx,xx0x,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**CS_FATAL_INTR_EN**: Fatal Interrupt enable bit for CS de-assert in Slave Mode. 1 = fatal_intr line will be asserted if CS_INACTIVE bit is set 1 in FIFO_STATUS Register. 0 = fatal_intr line will not be asserted. **0** = DISABLE **1** = ENABLE
30	0x0	**FRAME_END_FATAL_INTR_EN**: Fatal Interrupt enable bit for FRAME_END. 1 = fatal_intr line will be asserted if FRAME_END bit is set 1 in FIFO_STATUS Register. 0 = fatal_intr line will not be asserted. **0** = DISABLE **1** = ENABLE
28	0x0	**TX_FIFO_OVF_FATAL_INTR_EN**: Fatal Interrupt enable bit for TX_FIFO_OVF. 1 = fatal_intr line will be asserted if TX_FIFO_OVF bit is set 1 in FIFO_STATUS Register. 0 = fatal_intr line will not be asserted. **0** = DISABLE **1** = ENABLE
27	0x0	**TX_FIFO_UNF_FATAL_INTR_EN**: Fatal Interrupt enable bit for TX_FIFO_UNF. 1 = fatal_intr line will be asserted if TX_FIFO_UNF bit is set 1 in FIFO_STATUS Register. 0 = fatal_intr line will not be asserted. **0** = DISABLE **1** = ENABLE

Bit	Reset	Description
26	0x0	**RX_FIFO_OVF_FATAL_INTR_EN**: Fatal Interrupt enable bit for RX_FIFO_OVF. 1 = fatal_intr line will be asserted if RX_FIFO_OVF bit is set 1 in FIFO_STATUS Register. 0 = fatal_intr line will not be asserted. **0** = DISABLE **1** = ENABLE
25	0x0	**RX_FIFO_UNF_FATAL_INTR_EN**: Fatal Interrupt enable bit for RX_FIFO_UNF. 1 = fatal_intr line will be asserted if RX_FIFO_UNF bit is set 1 in FIFO_STATUS Register. 0 = fatal_intr line will not be asserted. **0** = DISABLE **1** = ENABLE
9	0x0	**CS_BOUNDARY_TIMEOUT_FATAL_INTR_EN**: Fatal Interrupt enable bit for CS_BOUNDARY_TIMEOUT. 1 = fatal_intr line will be asserted if CS_BOUNDARY_TIMEOUT_INTR bit is set 1 in FIFO_STATUS Register. 0 = fatal_intr line will not be asserted. **0** = DISABLE **1** = ENABLE

SPI_CS_BOUNDARY_TIMEOUT_0

SPI slave Rx continuous pause mode timeout Register

Offset: 0x19c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TIMEOUT_VALUE**: This timeout value is used only in slave Rx continuous pause mode. This timeout counter starts on every CS de-assert and resets on next CS assert. On timeout expiry, controller pushes an entry for one (or more) CS boundary(s) into CS_BOUNDARY_STATUS_FIFO. On timeout expiry, controller starts padding (dummy bytes) if data into main RX_FIFO is unaligned to dma_burst_size (which is always 16 in this case)

SPI_TIMEOUT_BOUNDARY_STATUS_0

SPI slave Rx continuous pause mode timeout boundary status register

Offset: 0x1a0
Read/Write: RO
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:28	0x0	**PADDED_BYTES**: Indicates number of padded bytes for current timeout boundary
27:0	0x0	**NUM_OF_PACKETS**: Indicated number of packets for current timeout boundary

SPI_TIMEOUT_BOUNDARY_FIFO_STATUS_0

SPI Slave Rx Continuous pause mode CS boundary FIFO status register

Offset: 0x1a4
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000001 (0bxxxx,xxxx,x000,0000,xxxx,xxx0,xxxx,0001)

Bit	R/W	Reset	Description
22:16	RO	0x0	**TIMEOUT_FIFO_FULL_COUNT**: Indicates the number of full slots in the Timeout boundary status FIFO
8	RW	0x0	**TIMEOUT_FIFO_FLUSH**: Flush the TIMEOUT FIFO **0** = NOP **1** = FLUSH
3	RW	0x0	**TIMEOUT_FIFO_OVF**: TIMEOUT FIFO Overflow **0** = OK **1** = ERROR
2	RW	0x0	**TIMEOUT_FIFO_UNF**: TIMEOUT FIFO Underflow **0** = OK **1** = ERROR
1	RO	0x0	**TIMEOUT_FIFO_FULL**: TIMEOUT FIFO Full **0** = NOT_FULL **1** = FULL
0	RO	0x1	**TIMEOUT_FIFO_EMPTY**: TIMEOUT FIFO Empty **0** = NOT_EMPTY **1** = EMPTY

SPI_SPI_DEBUG_REGISTER_0

Offset: 0x288
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
27:24	0x0	**CUR_STATE_REG_R_FSM**: Current state of read FSM
23:21	0x0	**CUR_STATE_REG_W_FSM**: Current state of write FSM
20:16	0x0	**CUR_STATE_REG_M_FSM**: Current state of master FSM
15:13	0x0	**CUR_STATE_REG_W_SLAVE_FSM**: Current state of slave FSM
12:9	0x0	**CUR_STATE_CONT_FIFO_WR_REQ_FSM**: Current state of slave_req FSM
8:5	0x0	**CUR_STATE_TIMEOUT_FSM**: Current state of Padding FSM
4:0	0x0	**PM_SYNC_FIFO_RD_COUNT**: pad_macro sync_fifo rd_count

9.10 Quad Serial Peripheral Interface (QSPI)

9.10.1 Overview

The Quad Serial Peripheral Interface (also called Quad SPI or QSPI) controller is a multi-bit serial communications link to compatible flash memory devices. The QSPI controller uses the Serial Peripheral Interface (SPI) protocol over one, two, or four data lanes. The System-on-Chip (SoC) embeds two QSPI controllers, each of which works independently. In the context of the I/O pins, these two QSPI controllers are referred to as QSPI0 and QSPI1.

9.10.1.1 Standard and Compatibility

- The Synchronous Serial Communication Interface, SPI (Serial Peripheral Interface bus) without the Slave Mode.
- Compatible with industry-standard flash devices from Cypress®/Spansion®, Micron®, and Macronix®.

9.10.1.2 Glossary

Note that different names may end up in the same acronym when the entire TRM is put in perspective. The acronyms listed here are within the context of the Quad Serial Peripheral Interface chapter.

Term	Definition
Word	A 32-bit data in the Tx or Rx FIFO.
Packet	Programmable data bit length that is being transferred. A typical packet size may be 8, 16, or 32 in packed/unpacked mode
Tx	QSPI as sender and flash is receiver
Rx	QSPI as receiver and flash is sender

9.10.1.3 Relevant Chapters in the TRM

- Address Map
- Clock Controller and Reset (CAR)
- General-Purpose Direct Memory Access (GP-DMA)
- Multi-Purpose I/O Pins and Pin Multiplexing (PinMux)

9.10.1.4 Features

- Master Mode Operation
  - SDR and DDR mode
  - Clock Polarity and Phase (Mode 0)
- Independent Rx and Tx FIFO
  - FIFO Size: 64 x 32 bits for PIO mode
- Byte-aligned and Packet-based (Packet Size = 8, 16, 32 bits) Packed and Unpacked Mode
  - Packed Mode's minimum transfer size: 32 bits (one FIFO word)
  - Unpacked Mode always used for transfer sizes less than 32 bits
- PIO or DMA Mode
- Single (x1) Mode, Dual (x2) Mode, Quad (x4) SDR and DDR Mode
- Configurable bit and byte endianness.
  - Least Significant Bit of Least Significant Byte (Endianness) Transmitted First and Received First as per NOR flash datasheet.
- Only one Slave Support with Programmable Polarity for Chip Select (CS), Hardware/Software Chip Select Control
- DMA Support

- Support for Combined Sequences Required for WRITE or READ Transactions to reduce CPU overhead.
- Hardware Calibration Support in QSPI Controller

9.10.2 Functional Description

The QSPI Controller is a Master on the QSPI bus. It has independent Tx and Rx FIFOs of 64 x 32 bits each for PIO mode and a FIFO of 40 x 64 bits for DMA mode. Because QSPI has only a half-duplex use-case, a single FIFO is used for Transmit (Tx) and Receive (Rx) in DMA mode. Software can program the controller to generate transactions of the required packet length on the QSPI bus in either direction.

Software can manually read/write to the FIFOs in PIO mode. PIO mode can also use GPC_DMA to read and write from the independent transmit/receive FIFOs as required. For DMA mode, the native DMA engine is used to read and write from the FIFO to memory as required. At the end of each transaction, an interrupt may be generated if the interrupt mask is cleared. Software uses transmit or receive operations in combination with chip select (CS) control to generate commands on the QSPI bus.

There are two system instances of QSPI in the SoC. These two instances QSPI0/1 are used as the primary boot from NOR flash.

9.10.2.1 Flash Devices Configurations

The types of devices and their configurations in the following diagrams are possible with the QSPI controllers. Exact part selection is subject to platform design choices, cost, performance, and overall platform considerations.

Figure 9.33 Single QSPI controller interfacing discrete package of x4 device

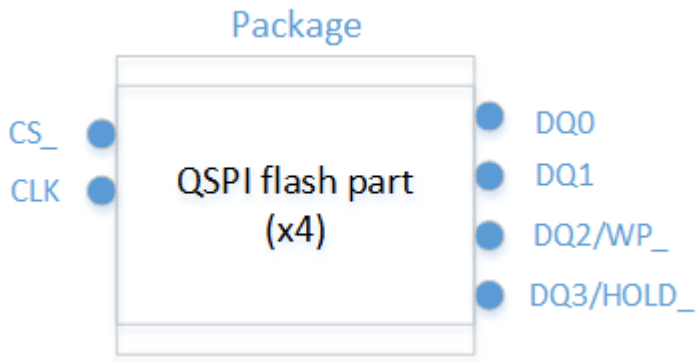


Figure 9.34 Two QSPI controllers driving one discrete package consisting of 2 x4 device

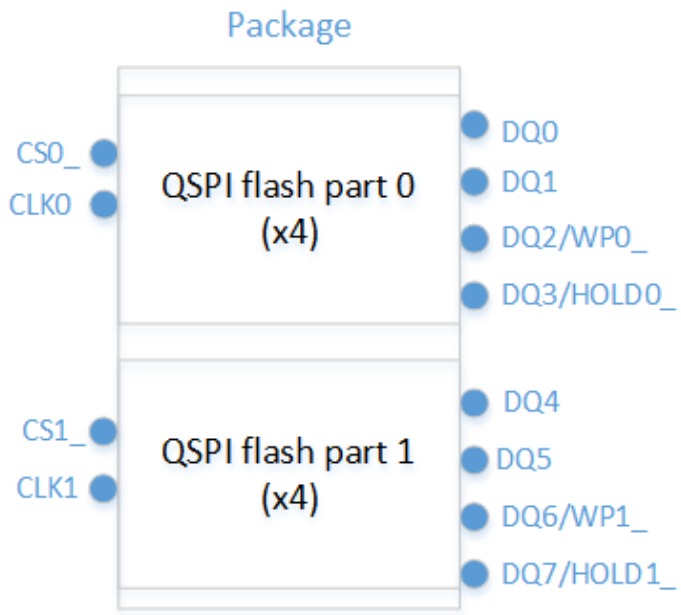
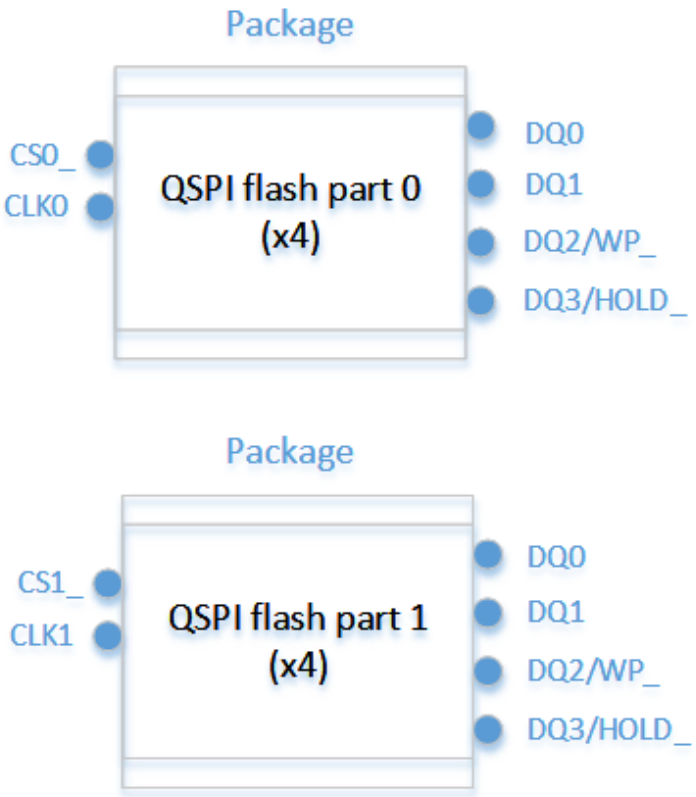


Figure 9.35 Two QSPI controllers driving two discrete packages each consisting of x4 device



9.10.2.2 Dual QSPI Bus Configuration

Two QSPI controller instances are available to speed up boot payload and meet Automotive platform boot KPIs. Each of the controller drives a single x4 port independently, and together effectively results in 2x bandwidth (240 MB/s due to reduced POR frequency). Partitioned payloads shall be flashed in two discrete x4 QSPI flashes or two x4 QSPI flashes on a single die.

9.10.2.3 System Configuration

Platform requirements and usage flow models:

- QSPI flash can be used as the primary boot device and supports 120 MHz DDR configuration.
- Hardware drive strength calibration is used to compensate for variations at high-speed (>80 MHz DDR) operation.
- QSPI usage flow:
  - Power on

- Drive strength calibration (Hardware)
- Trimmer tuning (Software)
- Load Payload for system boot (multiple phases of boot flow)
- Write accesses for minor updates
- Re-loading of payload when required

9.10.2.4 Drive Strength Programming

9.10.2.4.1 Drive Strength Programming Options

The QSPI interface requires the I/O drivers to have constant impedance for high-speed interface operations. Different options are possible to program drive up/down codes of QSPI interfaces. Required settings for different options and source of applied code details summarized in below table. The QSPI controller supports hardware managed interface drive strength calibration. Auto-Drive strength calibration sequence covered in programming section maps to option 3.

BDSDMEMLVCOMP pad is used to calibrate driver impedance and provide the drive code to data/control (BDSDMEMLV) pads. The data/control and comp pads for the QSPI interface only support the 1.8V I/O voltage.

Table 9.52 Drive Strength Programming Options

Option	DRV_TYPE	QSPI_AUTO_CAL_CONFIG_0.AUTO_CAL_ENABLE	QSPI_AUTO_CAL_CONFIG_0.AUTO_CAL_OVERRIDE	Drive Up/Dn Codes to Apply to Data/Control Pads
1	0x11	0 (default)	0 (default)	Drive Up Code: 0xa Drive Dn Code: 0xa This option fixes the drive code (internal default drive strength) applied to the pads and is meant for platforms that conform to the Design Guidelines.
2	0x01	0 (default)	0 (default)	Drive Up Code: (QSPI_QSPI_COMP_CONTROL_0.COMP_PAD_DRVUP_OVR) Drive Dn Code: (QSPI_QSPI_COMP_CONTROL_0.COMP_PAD_DRVDN_OVR) This option fixes the drive code applied by software programming. Select this option when the platform requires qualification work to come up with drive codes.

Option	DRV_TYPE	QSPI_AUTO_CAL_CONFIG_0.AUTO_CAL_ENABLE	QSPI_AUTO_CAL_CONFIG_0.AUTO_CAL_OVERRIDE	Drive Up/Dn Codes to Apply to Data/Control Pads
3	0x01	1	0 (default)	Drive Up Code: (Hardware auto-drive calibration strength + (QSPI_AUTO_CAL_CONFIG_0.AUTO_CAL_PU_OFFSET)) Drive Dn Code: (Hardware auto-drive calibration strength + (QSPI_AUTO_CAL_CONFIG_0.AUTO_CAL_PD_OFFSET)) This option allows the drive code to be applied based on the hardware drive strength calibration as opposed to applying fixed drive codes. Select this option when Option 1 does not work properly, or the static drive code fails such that periodic adjustment of the drive codes is needed to compensate for temperature changes.
4	0x01	1	1	Drive Up Code: (QSPI_AUTO_CAL_CONFIG_0.AUTO_CAL_PU_OFFSET) Drive Dn Code: (QSPI_AUTO_CAL_CONFIG_0.AUTO_CAL_PD_OFFSET) This option fixes the drive code applied by software programming. It is an alternative to Option 2.

9.10.2.5 Safety Mechanisms

9.10.2.5.1 Functional Errors

The error conditions described earlier are functional errors that may be used as diagnostic measures to realize a system-level safety mechanism. These functional errors are classified as Non-Fatal errors and therefore do not have hardware channeling to the Hardware Safety Manager (HSM) block. In contrast, these functional error interrupts when enabled and routed to the Legacy Interrupt Controller (LIC) may be managed by the safety management software to realize the safety mechanism desired.

Each of the QSPI controllers has the following Interrupt masks to select/enable the corresponding functional errors to trigger such interrupts.

Table 9.53 QSPI Functional Errors

Functional Error	Error Description	Interrupt Mask Setting to Enable Interrupt
1	Tx FIFO Overflow	(QSPI_INTR_MASK_0.TX_FIFO_OVF_INTR_MASK = 0)
2	Rx FIFO Overflow	(QSPI_INTR_MASK_0.RX_FIFO_OVF_INTR_MASK = 0)
3	Tx FIFO Underrun	(QSPI_INTR_MASK_0.TX_FIFO_UNF_INTR_MASK = 0)
4	Rx FIFO Underrun	(QSPI_INTR_MASK_0.RX_FIFO_UNF_INTR_MASK = 0)

9.10.2.5.2 Back Channel

In the back channel, data payload sent to the Flash device by the QSPI controller can benefit from a software-based CRC checksum. Such system-level software mechanisms can be employed to detect errors and recover from the detected errors using added redundancy or re-transmission to meet the safety requirements.

Since the APB register configuration does not support Parity checking, errors within a QSPI controller are expected to manifest themselves as I/O bus errors, which can be detected and recovered using the same system-level software-based safety mechanism.

9.10.3 Programming Guidelines

This section describes the programming model of the QSPI Flash controller.

9.10.3.1 Clock Initialization and Control

The following clock programming sequence is to be used for clock initialization or any other phases of boot or re-initialization of QSPI interfaces depending on system use cases. The PLL source should be selected based on the config. The following sequence is an example for PLLC4.

1. Enable PLL source PLLC4 by writing to CLK_RST_CONTROLLER_PLLC4_MISC1_0.
2. Assert QSPI controller reset by writing CLK_RST_CONTROLLER_RST_DEV_QSPI0/1_SET_0 to 0x1.
3. Enable clock to the device by configuring CLK_RST_CONTROLLER_CLK_OUT_ENB_QSPI0/1_SET_0. CLK_ENB_QSPI0/1 to 0x1.
4. Change the clock divisor, clock source to the clock switch (single register write) CLK_RST_CONTROLLER_CLK_SOURCE_QSPI_0/1_0.QSPI_CLK_SRC to appropriate PLL source, and divisor settings based on clock sharing policy guidelines.
5. Wait 2µs to make sure clock source/divider has changed.
6. De-assert device's reset CLK_RST_CONTROLLER_RST_DEV_QSPI0/1_CLR_0.

9.10.3.2 Clock Programming for Various QSPI Modes

1. Program the clock source and clock divider in CLK_RST_CONTROLLER_CLK_SOURCE_QSPI0/1_0.
   a. For SDR 166 MHz, clock divider must be programmed to generate 166 MHz output.
   b. For DDR 120 MHz, clock divider must be programmed to generate 240 MHz output.
      i. Set QSPI_CLK_DIV2_SEL to 1 in the CLK_RST_CONTROLLER_CLK_SOURCE_QSPI0/1_0 for DDR mode.
      ii. Set QSPI_CLK_DIV2_SEL to 0 in the CLK_RST_CONTROLLER_CLK_SOURCE_QSPI0/1_0 for SDR mode.

9.10.3.3 PLL Clocking Policy

PLLC4 is the preferred clock source option; PLLC, PLLC2, and PLLC3 are added as QSPI clock sources.

9.10.3.4 QSPI Controller Configuration

The following programming sequence is for configuring the controller irrespective of PIO/DMA mode.

The sequence of steps is same for Command, Address, or Data bytes to flash. If INTERFACE_WIDTH changes between command, address and data, multiple iterations are required.

1. Ensure that QSPI_COMMAND_0.PIO and QSPI_DMA_CTL_0.DMA_EN fields are in reset state.
2. Program QSPI_COMMAND_0 register based on the flash device and interface modes, and packet selection:
   a. PIO = 0x0
   b. M_S = 0x1
   c. MODE = 0x0 (for Mode0 selection)
   d. CS_SEL = 0x0
   e. CS_POL_INACTIVE = 0x1
   f. CS_SW_HW:
      i. Software control of chip-select, required settings: CS_SW_HW = 0x1, CS_SW_VAL = 0x0
      ii. Hardware control of chip-select, CS_SW_HW = 0x0, ignore CS_SW_VAL = 0x1
   g. IDLE_SDA = default value (This is only for debug purpose. Not to be touched in functional mode.)
   h. Data formatting modes selection if needed to re-arrange the data received from flash in terms of LSB/MSB or endianness:
      a. En_LE_Bit
      b. En_LE_Byte
      c. PACKED setting based on data mode selection
      ii. SDR_DDR_SEL = 0x0 for SDR mode and 0x1 for DDR mode.
      iii. INTERFACE_WIDTH = 0x0/0x1/0x2 depending on width of I/O bus. Note that CMD/ADDR width may be different from data width even for Quad flashes. Different INTERFACE_WIDTHs shall be accomplished with multiple PIO transfers in such cases.
      iv. BIT_LENGTH = depending on packet size selection.
      v. Tx_En or Rx_En depending on write or read transfer.
3. Program num_of_dummy_cycles in QSPI_MISC_0 register (required for fast read, dual/quad read operation) while setting up a Tx transfer (for CMD and ADDR) based on device requirement.

4. Program QSPI_COMMAND2_0 (Rx trimmer tap settings) if required to be changed from default values as recommended by Silicon Characterization.
5. Program QSPI_TIMING_REG1_0/QSPI_TIMING_REG2_0/QSPI_TIMING3_0 based on flash device timing settings.
    i. For hardware-based chip select selection only, setup and hold timing settings in CS Timing register are relevant.
6. Clear QSPI_INTR_MASK_0 register bit fields to enable required interrupt.

9.10.3.5 PIO Data Transfer Using CPU

The following sequence is for PIO mode of data transfer. The sequence of steps is the same for Command, or Address or Data bytes to flash. If INTERFACE_WIDTH changes between command, address, and data, then multiple iterations are required.

1. Follow the steps outlined above for QSPI Controller Configuration.
2. Wait for 1 µs delay so that all configuration bits are settled.
3. Write data to the Tx FIFO for Tx transfer.
4. Enable PIO mode transfer by writing `1' to PIO bit in QSPI_COMMAND_0 register.
5. Wait for RDY interrupt/poll for RDY status in QSPI_TRANSFER_STATUS_0.
6. Program QSPI_COMMAND_0 register CS_SW_VAL=0x1 to de-active the chip select for software driven chip select.
7. Read the received data in the Rx FIFO for Rx transfer.

9.10.3.6 DMA Data Transfer

The following sequence is for DMA mode of data transfer using the QSPI internal DMA engine. In this mode, the QSPI controller transmits or receives the number of packets as indicated by the field DMA_BLOCK_SIZE in QSPI_DMA_BLK_SIZE_0 register from/to the memory. The memory address is configurable in the controller.

1. Follow the steps outlined in QSPI Controller Configuration.
2. Program DMA-related registers:
    a. QSPI_AXI_CTL_0 register
        i. MC_WPROT/MC_RPROT based on the access requirement to secure/non-secure memory. Note that this register can only be modified by TrustZone-OS.
    b. QSPI_DMA_BLK_SIZE_0 register
        i. DMA_BLOCK_SIZE with required packet transfer length. Number of data bytes = (DMA_BLOCK_SIZE_N + 1) X No. of bytes as per BIT_LENGTH and packed selection.
    c. QSPI_DMA_MEM_ADDRESS_0 – lower 32 bits of memory address
    d. QSPI_DMA_HI_ADDRESS_0 – higher 8 bits of memory address
3. Wait for 1 µs delay so that all configuration bits are settled.
4. Enable the transfer by writing `1' to DMA_EN bit in QSPI_DMA_CTL_0 register.

5. Wait for RDY interrupt/poll for RDY status in QSPI_TRANSFER_STATUS_0.

9.10.3.7 Programming Trimmers

There are two programmable trimmers in QSPI: QSPI-Tx trimmer and QSPI-Rx trimmer. These trimmers are used only when QSPI is in Master Mode.

The QSPI-Tx Trimmer is used in Transmit mode to adjust/center the outgoing clock with outgoing data. The QSPI-Rx trimmer is used in Receive mode to delay the loopback clock to center align the Receive data to meet setup/hold and data valid requirements.

The QSPI-Tx Trimmer can be programmed to meet timing requirements between SCK and IO[3:0]. Program Tx_Clk_TAP_DELAY, probe SCK and IO[3:0], and measure the setup/hold time.

9.10.3.7.1 Rx Trimmer Tuning Sequence

The QSPI controller does not implement any auto-tuning hardware procedure to find the optimum value of Rx_Clk_TAP_DELAY settings for clock sampling adjustment. However, it is possible to accomplish the same with software managed tuning sequence. Software sends commands, reads the tuning pattern, logs the pass/fail criterion, changes the trimmer setting, post-process the results to identify passing windows and finally to arrive at the required Rx_Clk_TAP_DELAY settings.

Some QSPI flash vendors support a DLP mechanism, where in flash outputs data pattern during dummy cycle phase. Since the controller generates dummy cycles during Transmit phase, it will not be able to sample DLP data during the dummy_cycle phase. Because the QSPI controller does not sample the DLP pattern in hardware, it will not use that method. However, software can sample the DLP pattern by the following tuning scheme.

1. Programming dummy_cycle is not really required.
   a. That means leave NUM_OF_DUMMY_CLK_CYCLES field of the QSPI_MISC_0 register to its default value (0x0).
2. Program/Increment the RX_CLK_TAP_DELAY setting.
3. Set up receive transfer (length of transfer in num_of_dummy_cycle + some_extra_bytes):
   a. For example, if in x4 mode four dummy cycles are required, software may setup a normal receive transfer of 8 bytes (4 bytes for dummy cycles + 4 bytes of extra bytes). Software needs to program DMA_BLOCK_SIZE to seven packets with BIT_LENGTH=0x7 selection (which means 8 bytes).
4. Extract DLP data and compare with expected data pattern (vendor documentation).
5. Log the pass/fail criterion.
6. Repeat steps 2-5 until entire RX_CLK_TAP_DELAY possible range is completed.
   a. RX_CLK_TAP_DELAY range is fine-tuned based on Si char results to optimize the DLP tuning time.

7. Post-process the pass/fail information to identify largest passing window. If there are multiple passing windows, choose the lower trimmer range. From passing windows, identify the largest passing window.
8. Fix the RX_CLK_TAP_DELAY to the passing window middle point, which is the best margin point for setup/hold time.
   a. Selection from passing window shall be based on Si char. There is no guarantee that middle point is the “safe margin” point.

Once tuning is complete, software can disable the DLP by writing 0x00 in VDLP register (using WVDLR 4Ah cmd). Normal Tx/Rx transfers to be initiated from there on.

9.10.3.8 Auto-Drive Strength Calibration Sequence

Enable auto-calibration.

1. Auto calibration
   a. Set QSPI_COMP_PAD_E_INPUT_OR_E_PWRD = 1 in QSPI_QSPI_COMP_CONTROL_0 register.
   b. Wait for 1us after QSPI_COMP_PAD_E_INPUT_OR_E_PWRD is enabled.
   c. Set AUTO_CAL_START and AUTO_CAL_ENABLE to 1 in QSPI_AUTO_CAL_CONFIG_0 register.
   d. Wait for 1µs.
   e. Wait for AUTO_CAL_ACTIVE in QSPI_AUTO_CAL_STATUS_0 register to become 0.
   f. Clear the QSPI_COMP_PAD_E_INPUT_OR_E_PWRD setting to `0’ in QSPI_QSPI_COMP_CONTROL_0 register to save power.
      i. Software should not clear AUTO_CAL_ENABLE in the QSPI_AUTO_CAL_CONFIG_0 register after calibration. Set to 1 to use calibration codes generated by calibration controller.
      ii. If periodic calibration is required, wait for minimum 100ms and re-run calibration when the QSPI I/F is IDLE.

9.10.4 QSPI Registers

Refer to "Reading Register Tables" in the Introduction chapter for the register table protocol as well as recommendations for accessing registers.

There are two sets of the QSPI controller registers, one for each QSPI controller in the Orin SoC. The register descriptions below give the offset of each register within the QSPI controller's address range. The Base Addresses of the registers in these two QSPI controllers are specified in the Address Map section of the Orin TRM.

QSPI_COMMAND_0

Offset: 0x0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x4050001f (0b0100,00xx,x101,0000,0xx0,0x00,0x01,1111)

Bit	Reset	Description
31	0x0	**PIO**: Program 1 after all the configuration fields are programmed to start the transfer. Hardware clears this bit automatically after the transfer is done. Clearing of the bit by Software will stop the Shifter and latch the partial data into FIFO. **0** = STOP **1** = PIO
30	0x1	**M_S**: 1 = Master Mode (internal Clock) 0 = RSVD **0** = RSVD **1** = MASTER
29:28	0x0	**MODE**: 11 = Mode3 10 = RSVD 01 = RSVD 00 = Mode0 (def); Only Master Mode-0 is supported. **0** = Mode0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3
27:26	0x0	**CS_SEL**: 11 = RSVD 10 = RSVD 01 = RSVD 00 = chip select0 (def) **0** = CS0 **1** = RSVD1 **2** = RSVD2 **3** = RSVD3
22	0x1	**CS_POL_INACTIVE0**: In Master Mode, the inactive value of external devices CS value which is connected to CS0 needs to be programmed 1 = CS0 Inactive value of external device is high 0 = CS0 Inactive value of external device is low **0** = LOW **1** = HIGH
21	0x0	**CS_SW_HW**: Software control of CS signal in Master Mode. 1 = CS controlled by software 0 = CS controlled by hardware **0** = HARDWARE **1** = SOFTWARE
20	0x1	**CS_SW_VAL**: When CS is controlled by Software this value is driven onto the selected CS line. 1 = CS is high 0 = CS is low **0** = LOW **1** = HIGH

Bit	Reset	Description
19:18	0x0	**IDLE_SDA**: 11 = Pull High 10 = Pull Low 01 = Driven High 00 = Driven Low **0** = DRIVE_LOW **1** = DRIVE_HIGH **2** = PULL_LOW **3** = PULL_HIGH
17	0x0	**BIDIR**: 1 = RSVD 0 = Normal mode **0** = NORMAL **1** = RSVD
16	0x0	**En_LE_Bit**: 1 = Enable Little Endian Bit 0 = Disable Little Endian Bit **0** = LAST **1** = FIRST
15	0x0	**En_LE_Byte**: 1 = Enable Little Endian Byte 0 = Disable Little Endian Byte **0** = LAST **1** = FIRST
12	0x0	**Rx_EN**: **0** = DISABLE **1** = ENABLE
11	0x0	**Tx_EN**: **0** = DISABLE **1** = ENABLE
9	0x0	**SDR_DDR_SEL**: Selects between SDR (Single data rate) and DDR (Double data rate) mode. 0 = SDR 1 = DDR **0** = SDR **1** = DDR
8:7	0x0	**INTERFACE_WIDTH**: 00 = Single bit mode (x1 mode) 01 = Dual mode (x2 mode) 10 = Quad mode (x4 mode) 11 = RSVD **0** = SINGLE **1** = DUAL **2** = QUAD
5	0x0	**PACKED**: Packed mode enable bit. 1 = Packed mode is enabled. This is only valid if BIT_LENGTH in COMMAND register is set to 7, 15 or 31. When enabled, all 32-bits of data in the FIFO contains valid data packets of either 8-bit, 16-bit or 32-bit length. 0 = Packed mode is disabled. **0** = DISABLE **1** = ENABLE
4:0	0x1f	**BIT_LENGTH**: This field represents the number of bits in a packet to transmit/receive. The minimum bit_length supported is 1 byte (BIT_LEN=7). Only byte aligned bit_length are supported (BIT_LEN=7,15,31). N = N + 1 bits transfer.

QSPI_COMMAND2_0

Offset: 0x4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,x000,00xx,0000,0000)

Bit	Reset	Description
14:10	0x0	**Tx_Clk_TAP_DELAY**: Delays the clock going out to the external device with these tap values. Useful only in Master Mode.
7:0	0x0	**Rx_Clk_TAP_DELAY**: For master mode, it uses the tap values to delay the internal Rx clock, which is a loopback from the SCK pad.

QSPI_TIMING_REG1_0

Offset: 0x8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:4	0x0	**CS_SETUP_0**: Number of cycles for CS-0 Setup
3:0	0x0	**CS_HOLD_0**: Number of cycles for CS-0 Hold

QSPI_TIMING_REG2_0

Offset: 0xc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000020 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx10,0000)

Bit	Reset	Description
5	0x1	**CS_ACTIVE_BETWEEN_PACKETS_0**: Specifies if CS stays active in between two packets on CS-0 1 = CS active between two packets 0 = RSVD

Bit	Reset	Description
4:0	0x0	**CYCLES_BETWEEN_PACKETS_0**: Number of cycles in between packets for CS-0 RSVD field

QSPI_TRANSFER_STATUS_0

Offset: 0x10
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bx0xx,0000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
30	RW	0x0	**RDY**: Ready bit. This bit is set to 1 at the end of every transfer and an interrupt is also generated if the corresponding interrupt enable is set in PIO/DMA Mode. Software writes a 1 to clear it. The interrupt is also cleared when this bit is cleared. **0** = NOT_READY **1** = READY
27:0	RO	0x0	**BLK_CNT**: Counts the number of packets in a transfer (Tx or Rx) PIO/DMA Mode.

QSPI_FIFO_STATUS_0

Offset: 0x14
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00400005 (0bxx00,0000,0100,0000,00xx,xxx0,0000,0101)

Bit	R/W	Reset	Description
29:23	RO	0x0	**RX_FIFO_FULL_COUNT**: Indicates the number of full slots in the receive PIO FIFO
22:16	RO	0x40	**TX_FIFO_EMPTY_COUNT**: Indicates the number of empty slots in the transmit PIO FIFO
15	RW	0x0	**RX_FIFO_FLUSH**: Flush the PIO Rx FIFO **0** = NOP **1** = FLUSH

Bit	R/W	Reset	Description
14	RW	0x0	**TX_FIFO_FLUSH**: Flush the PIO Tx FIFO **0** = NOP **1** = FLUSH
8	RW	0x0	**ERR**: Will be set to 1 by Hardware when errors such as underflow/ overflow occurs in PIO/GPCDMA mode. Write 1 to clear the flag. **0** = OK **1** = ERROR
7	RW	0x0	**TX_FIFO_OVF**: TX FIFO Overflow **0** = OK **1** = ERROR
6	RW	0x0	**TX_FIFO_UNF**: TX FIFO Underflow **0** = OK **1** = ERROR
5	RW	0x0	**RX_FIFO_OVF**: RX FIFO Overflow **0** = OK **1** = ERROR
4	RW	0x0	**RX_FIFO_UNF**: RX FIFO Underflow **0** = OK **1** = ERROR
3	RO	0x0	**TX_FIFO_FULL**: TX FIFO Full **0** = NOT_FULL **1** = FULL
2	RO	0x1	**TX_FIFO_EMPTY**: TX FIFO Empty **0** = NOT_EMPTY **1** = EMPTY
1	RO	0x0	**RX_FIFO_FULL**: RX FIFO Full **0** = NOT_FULL **1** = FULL
0	RO	0x1	**RX_FIFO_EMPTY**: RX FIFO Empty **0** = NOT_EMPTY **1** = EMPTY

QSPI_TX_DATA_0

QSPI Sub-block Transmit Data Page Buffer Register

Offset: 0x18
Read/Write: RO

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**Tx_BUFFER**

QSPI_RX_DATA_0

QSPI Sub-block Slave Data Page Buffer Register

Offset: 0x1c
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**Rx_BUFFER**

QSPI_DMA_CTL_0

QSPI Sub-block DMA Control Register

Offset: 0x20
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0xxx,xxxx,xxx0,0xx0,0xxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
31	0x0	**DMA_EN**: 1 = DMA mode is enabled, 0 = DMA disabled **0** = DISABLE **1** = ENABLE
20:19	0x0	**RX_TRIG**: Receive FIFO trigger level. **0** = RSVD1 **1** = RSVD4 **2** = RSVD8 **3** = RSVD16

Bit	Reset	Description
16:15	0x0	**TX_TRIG**: Transmit FIFO trigger level. **0** = RSVD1 **1** = RSVD4 **2** = RSVD8 **3** = RSVD16

QSPI_DMA_BLK_SIZE_0

Offset: 0x24
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
27:0	0x0	**DMA_BLOCK_SIZE**: N = N+1 packets Block size to be transferred in PIO/DMA mode

QSPI_DMA_MEM_ADDRESS_0

Offset: 0x28
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**MEM_ADDRESS**: Address for data read/write from/to memory

QSPI_DMA_HI_ADDRESS_0

Offset: 0x2c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**HI_ADDRESS**

QSPI_DMA_FIFO_STATUS_0

Offset: 0x30
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000800 (0b0000,00xx,xxxx,xxxx,xx0x,1000,xxxx,xxxx)

Bit	R/W	Reset	Description
31:26	RO	0x0	**DMA_FIFO_COUNT**
13	RW	0x0	**DMA_FIFO_FLUSH**: Flush DMA FIFO **0** = NOP **1** = FLUSH
11	RO	0x1	**DMA_FIFO_EMPTY**: DMA FIFO Empty **0** = NOT_EMPTY **1** = EMPTY
10	RW	0x0	**DMA_FIFO_OVF**: DMA FIFO Overrun **0** = OK **1** = ERROR
9	RW	0x0	**DMA_FIFO_UNF**: DMA FIFO Underrun **0** = OK **1** = ERROR
8	RW	0x0	**DMA_FIFO_ERR**: Will be set to 1 by Hardware when errors such as Underflow/ overflow occurs in native DMA mode. Write 1 to clear the flag **0** = OK **1** = ERROR

QSPI_DMA_TRANSFER_STATUS_0

Offset: 0x34
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:30	0x0	**AXI_BRESP**: AXI Write response
29:28	0x0	**AXI_RRESP**: AXI Read response
27:0	0x0	**AXI_BLK_CNT**: Number of words transferred to/from memory.

QSPI_TX_FIFO_0

QSPI Sub-block Tx FIFO Buffer Register

Offset: 0x108
Read/Write: WO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**TX_FIFO_REGISTER**

QSPI_RX_FIFO_0

QSPI Sub-block Rx FIFO Buffer Register

Offset: 0x188
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**RX_FIFO_REGISTER**

QSPI_INTR_MASK_0

QSPI Sub-block Interrupt Mask Register

Offset: 0x18c
Read/Write: R/W

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxx00,0000,0xxx,xxxx,xxxx,xxxx,xxxx,xxxx)

Bit	Reset	Description
29	0x0	**RDY_INTR_MASK**: Interrupt enable mask bit for RDY. 1 = Disable interrupt generation when RDY bit is asserted. 0 = Enable interrupt generation when RDY bit is asserted. **0** = DISABLE **1** = ENABLE
28	0x0	**TX_FIFO_OVF_INTR_MASK**: Interrupt enable mask bit for TX_FIFO_OVF. 1 = Disable interrupt generation when TX_FIFO_OVF is asserted. 0 = Enable interrupt generation when TX_FIFO_OVF is asserted. **0** = DISABLE **1** = ENABLE
27	0x0	**TX_FIFO_UNF_INTR_MASK**: Interrupt enable mask bit for TX_FIFO_UNF. 1 = Disable interrupt generation when TX_FIFO_UNF is asserted. 0 = Enable interrupt generation when TX_FIFO_UNF is asserted. **0** = DISABLE **1** = ENABLE
26	0x0	**RX_FIFO_OVF_INTR_MASK**: Interrupt enable mask bit for RX_FIFO_OVF. 1 = Disable interrupt generation when RX_FIFO_OVF is asserted. 0 = Enable interrupt generation when RX_FIFO_OVF is asserted. **0** = DISABLE **1** = ENABLE
25	0x0	**RX_FIFO_UNF_INTR_MASK**: Interrupt enable mask bit for RX_FIFO_UNF. 1 = Disable interrupt generation when RX_FIFO_UNF is asserted. 0 = Enable interrupt generation when RX_FIFO_UNF is asserted. **0** = DISABLE **1** = ENABLE
24	0x0	**DMA_FIFO_OVF_INTR_MASK**: Interrupt enable mask bit for DMA_FIFO_OVF. 1 = Disable interrupt generation when DMA_FIFO_OVF is asserted. 0 = Enable interrupt generation when DMA_FIFO_OVF is asserted. **0** = DISABLE **1** = ENABLE
23	0x0	**DMA_FIFO_UNF_INTR_MASK**: Interrupt enable mask bit for DMA_FIFO_UNF. 1 = Disable interrupt generation when DMA_FIFO_UNF is asserted. 0 = Enable interrupt generation when DMA_FIFO_UNF is asserted. **0** = DISABLE **1** = ENABLE

QSPI_SPARE_CTLR_0

Offset: 0x190
Read/Write: R/W

Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0fff0000 (0b0000,1111,1111,1111,0000,0000,0000,0000)

Bit	Reset	Description
31:24	0xf	**SPARE_CONTROL_REGISTER_BYTE4**
23:16	0xff	**SPARE_CONTROL_REGISTER_BYTE3**
15:8	0x0	**SPARE_CONTROL_REGISTER_BYTE2**
7:0	0x0	**SPARE_CONTROL_REGISTER_BYTE1**

QSPI_MISC_0

Offset: 0x194
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x80000000 (0b1xxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
31	0x1	**CLKEN_OVERRIDE**: Override for qspi_clk. Can be used to bypass the SLCG incase of issues. By default SLCG is not enabled. Software should turn on by writing this bit as 0. 0: qspi_clk is gated 1: qspi_clk is not gated
7:0	0x0	**NUM_OF_DUMMY_CLK_CYCLES**: Number of dummy cycles required in case of Fast read commands. This is useful only in case of read from flash.

QSPI_TIMING3_0

Offset: 0x198
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxx0,0000,xxx0,0000,xxx0,0000,xxx0,0000)

Bit	Reset	Description
28:24	0x0	**DATA3_LINE_TAP_DELAY**: Delays the Data3 coming in from the external device with these tap values

Bit	Reset	Description
20:16	0x0	**DATA2_LINE_TAP_DELAY**: Delays the Data2 coming in from the external device with these tap values
12:8	0x0	**DATA1_LINE_TAP_DELAY**: Delays the Data1 coming in from the external device with these tap values
4:0	0x0	**DATA0_LINE_TAP_DELAY**: Delays the Data0 coming in from the external device with these tap values

QSPI_CMB_SEQ_CMD_0

Offset: 0x19c
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,0000,0000)

Bit	Reset	Description
7:0	0x0	**COMMAND_VALUE**: Contains the Command Value that goes out to flash

QSPI_CMB_SEQ_CMD_CFG_0

Offset: 0x1a0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000007 (0bxxxx,xxxx,xxxx,xxxx,x000,xxxx,0000,0111)

Bit	Reset	Description
14:13	0x0	**COMMAND_X1_X2_X4**: Indicates interface width of CMD 00 = Single bit mode (x1 mode) 01 = Dual mode (x2 mode) 10 = Quad mode (x4 mode) 11 = RSVD
12	0x0	**COMMAND_SDR_DDR**: Indicates whether CMD is in SDR or DDR mode 0 = SDR ; 1 = DDR

Bit	Reset	Description
7:0	0x7	**COMMAND_SIZE**: Contains the Command size in bits (n+1). 0x0 = 1-bit wide cmd. 0x1 = 2-bit wide cmd. ... 0x1f = 32-bit wide cmd.

QSPI_GLOBAL_CONFIG_0

Offset: 0x1a4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xx00)

Bit	Reset	Description
1	0x0	**TPM_WAIT_POLL_EN**: Enable polling for wait state in TPM devices
0	0x0	**CMB_SEQ_EN**: Indicates whether Combined sequence mode is enabled or not.

QSPI_CMB_SEQ_ADDR_0

Offset: 0x1a8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**ADDRESS_VALUE**: Contains the Address Value that goes out to flash

QSPI_CMB_SEQ_ADDR_CFG_0

Offset: 0x1ac
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0x00000017 (0bxxxx,xxxx,xxxx,xxxx,x000,xxxx,0001,0111)

Bit	Reset	Description
14:13	0x0	**ADDRESS_X1_X2_X4**: Indicates interface width of ADDR 00 = Single bit mode (x1 mode) 01 = Dual mode (x2 mode) 10 = Quad mode (x4 mode) 11 = RSVD
12	0x0	**ADDRESS_SDR_DDR**: Indicates whether ADDR is in SDR or DDR mode 0 = SDR; 1 = DDR
7:0	0x17	**ADDRESS_SIZE**: Contains the Address size in bits (n+1). 0x0 = 1-bit wide address. 0x1 = 2-bit wide address. ... 0x1f = 32-bit wide address.

QSPI_QSPI_COMP_CONTROL_0

Offset: 0x1ec
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00a0a00X (0bxxxx,xxx0,1010,xxx0,1010,xxxx,x000,0x0x)

Bit	R/W	Reset	Description
24:20	RW	0xa	**COMP_PAD_DRVUP_OVR**: Used to drive DRVUP input of COMP pad if AUTO_CAL_ENABLE is disabled
16:12	RW	0xa	**COMP_PAD_DRVDN_OVR**: Used to drive DRVDN input of COMP pad if AUTO_CAL_ENABLE is disabled
6:3	RW	0x0	**QSPI_COMP_PAD_VREF_SEL**: This is no more used inside COMP pad 7:7 r/w QSPI_COMP_PAD_REG_ON i=0x0
1	RW	0x0	**QSPI_COMP_PAD_E_INPUT_OR_E_PWRD**: Default value is 0 needed to start Calibration. It should be set as 1 once calibration is done to save power. 2:2 r/w QSPI_COMP_PAD_CLK i=0x0
0	RO	X	**QSPI_COMP_CALIB_STATUS**

QSPI_AUTO_CAL_CONFIG_0

QSPICOMP pad auto-calibration settings

Offset: 0x1f0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00010000 (0b0000,xxxx,xxxx,x001,xxx0,0000,xxx0,0000)

Bit	Reset	Description
31	0x0	**AUTO_CAL_START**: Writing a one to this bit starts the calibration state machine. This bit must be set even if the override is set in order to latch in the override value.
30	0x0	**AUTO_CAL_OVERRIDE**: AUTOCAL override. **0** = NORMAL (normal operation): use AUTO_CAL_PU/PD_OFFSET as an offset to the calibration state machine setting **1** = OVERRIDE: use AUTO_CAL_PU/PD_OFFSET register values directly
29	DISABLED	**AUTO_CAL_ENABLE**: AUTOCAL enable. **0** = DISABLED: use qspi2tmc_cfg* register settings for pullup/down **1** = ENABLED (normal operation): use QSPI generated pullup/down (override or AUTOCAL)
28	0x0	**AUTO_CAL_SLW_OVERRIDE**: AUTOCAL slew rate override **0** = NORMAL (Normal operation) pad DRVDN/UP_SLWR/F tied to AUTO_CAL output DRDVDN/UP_SLWR/F[1:0] = AUTO_CAL_PULLDOWN/UP[4:3] **1** = OVERRIDE: use CFG2TMC_QSPI_DRVDN/UP_SLWR/F pins to control pad slew inputs
18:16	0x1	**AUTO_CAL_STEP**: Calibration step interval (in microseconds)
12:8	0x0	**AUTO_CAL_PD_OFFSET**: 2's complement offset for pull-down value
4:0	0x0	**AUTO_CAL_PU_OFFSET**: 2's complement offset for pull-up value

QSPI_AUTO_CAL_INTERVAL_0

Offset: 0x1f4
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
31:0	0x0	**AUTO_CAL_INTERVAL**: 0: Do calibration once Otherwise, auto-calibration occurs at intervals equivalent to the programmed number of microseconds.

QSPI_AUTO_CAL_STATUS_0

QSPICOMP pad calibration status

Offset: 0x1f8
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0xx0,0000,xxx0,0000,xxx0,0000,xxx0,0000)

Bit	Reset	Description
31	0x0	**AUTO_CAL_ACTIVE**: One when auto calibrate is active valid only after auto-calibrate sequence has completed (AUTO_CAL_ACTIVE == 0)
28:24	0x0	**AUTO_CAL_PULLDOWN_ADJ**: Pulldown code sent to pads
20:16	0x0	**AUTO_CAL_PULLUP_ADJ**: Pullup code sent to pads
12:8	0x0	**AUTO_CAL_PULLDOWN**: Pulldown code generated by auto-calibration
4:0	0x0	**AUTO_CAL_PULLUP**: Pullup code generated by auto-calibration

QSPI_IO_TRIM_CNTRL_0

This register is used to configure NBSDMEM_TRIM cell
Offset: 0x1fc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x0000000a (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxx0,0000,1010)

Bit	Reset	Description
8	0x0	**TRIM_BYPASS_SEL**: 0: Uses trimmer clock output pin for sampling 1: Uses trimmer bypass clock output for sampling (no delay)

Bit	Reset	Description
7:6	0x0	**TRIM_SEL_ATEST**: For testing purpose only - should be used when QSPI is in idle state. Select analog test signals to send to comp pad. 0x0: Not used (float) 0x1: Regulator input voltage 0x2: Regulator output voltage before analog mux 0x3: Regulator output after analog mux
5	ENABLE	**TRIM_PWRSAVE**: Enables power saving mode by clock gating the unused taps in delay chain Active low signal, 0: power saving mode enabled - clock gating is enabled for unused trimmer taps - may affect tap delay 1: no power saving - all the trimmer taps are not clock gated **0** = ENABLE **1** = DISABLE
4:2	VREF_825_MV	**SEL_VREF_LEVEL**: Selects Vref voltage level **0** = VREF_775_MV **1** = VREF_800_MV **2** = VREF_825_MV **3** = VREF_850_MV
1	0x1	**SEL_VREG**: By default, BG is disabled to save power if interface is not used. Software should select BG for error free QSPI operation. PROD value: 0x0 ***Software should set this to 0x0 before accessing QSPI. This setting makes IB trimmer delay independent of VDD_CORE*** 0: Selects regulated reference voltage for trimmer supply - default (recommended option for tunable QSPI modes) 1: Selects VAUXC for trimmer supply and shut down BG+REG circuit (can be used in non-Tunable modes for power saving) Power up time for BG+REG is ~3µs (worst case). Power down time for BG+REG is ~1µs (worst case). If Software wants to turn on/off BG+REG when QSPI is idle, it has to take hit of 3µs power on time.
0	0x0	**SEL_VREF**: Select reference voltage for voltage regulator 0: Selects Bandgap Voltage Reference (recommended option for QSPI tunable modes) 1: Selects resistor divider voltage reference and power down bandgap Switching time between the supplies is 1µs. When switching from one supply to other supply, we need to wait for at least 1µs before doing any data transfers. Providing reference voltage from R divider network is just a backup plan, if: A. Bandgap does not work or, B. Bandgap works very well but we want to save bandgap power when Silicon Characterization results shows that the QSPI interface perform well even by using R divider+REG+TRIMMER

QSPI_GLOBAL_TRIM_CNTRL_0

Offset: 0x200
Read/Write: R/W
Parity Protection: N

Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxx0)

Bit	Reset	Description
0	0x0	**SEL**: 0: Uses clock from NBSDMEM_TRIM cell for sampling (DVFS independent trimmer) 1: Uses clock from prog_dly_256tap for sampling

9.11 Pulse Width Modulator (PWM)

9.11.1 Overview

The SoC has eight Pulse Width Modulator (PWM) outputs. Each PWM output is based on a frequency divider whose pulse width varies. Each has a programmable frequency divider and a programmable pulse width generator. The PWM controller supports one PWM output for each of its eight instances (PWM1 through PWM8). Each instance is allocated a 64 KB independent address space.

Frequency division is a 13-bit programmable value, and pulse division is an 8-bit value. The PWM can run at a maximum frequency of up to 408 MHz.

The PWM controller can source its clock from either CLK_M or PLLP. CLK_M (19.2 MHz) is derived from the OSC clock (38.4 MHz). PLLP operates at 408 MHz.

The PWM clock frequency is divided by 256 before subdividing it based on the programmable frequency division value to generate the required frequency for the PWM output. The maximum output frequency that can be achieved from this configuration is 408 MHz/256 = 1.6 MHz. This 1.6 MHz frequency can be further divided using the frequency divisor in PWM.

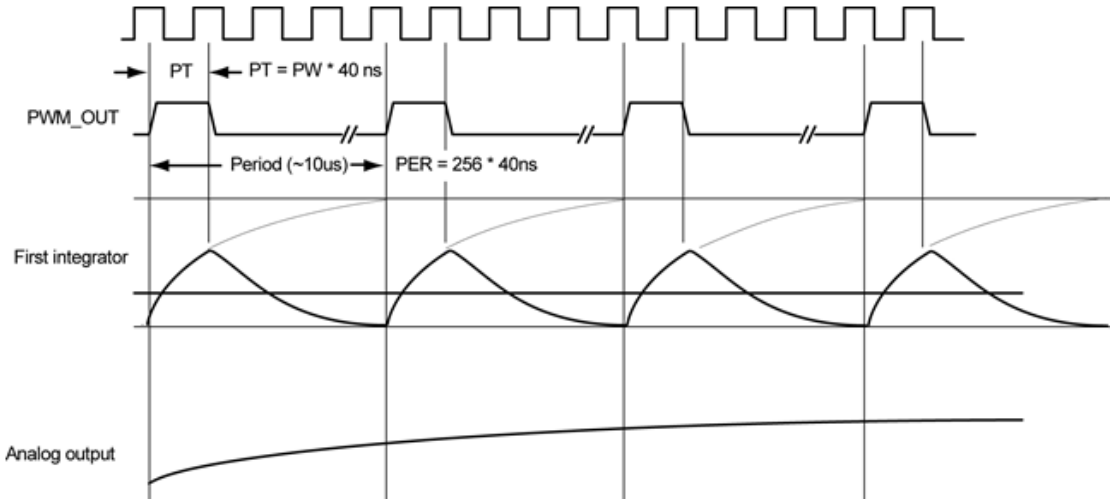
The OSC clock is the primary/default source for the PWM IP clock. For higher PWM output frequency requirements, PLLP is the clock source (up to 408 MHz).

An APB interface connects the register logic to the APB bus (see the chip block diagram in the Introduction chapter).

9.11.1.1 PWM Diagrams

Each PWM controller contains one programmable pulse width modulator. Each generated pulse has an n/256 duty cycle. The figure below shows a pulse width diagram. The maximum frequency of the device clock is 408 MHz.

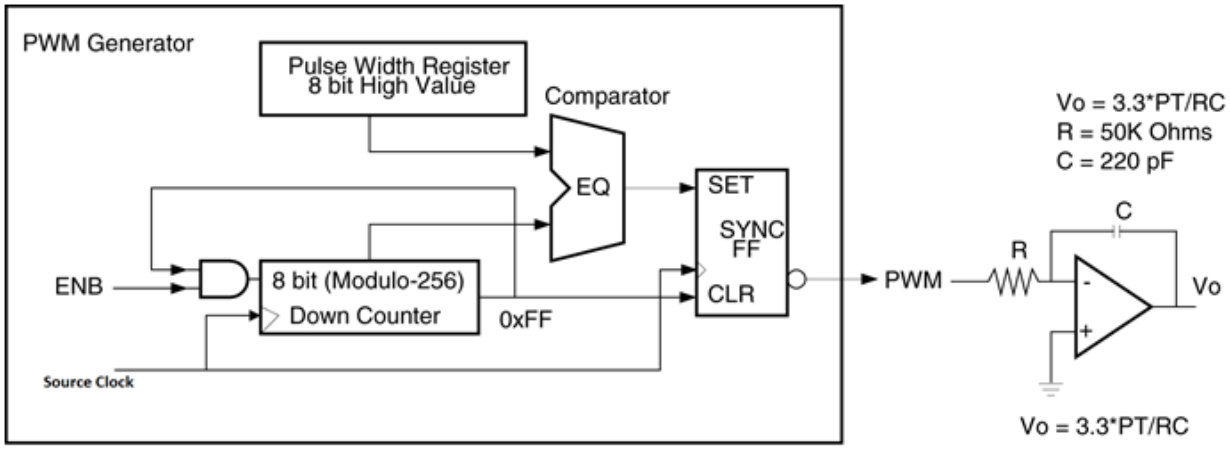
Figure 9.36 Pulse Width Diagram



PWM signals are useful for LCD contrast and brightness control, VCO-generated clocks and other analog voltage references where high precision is not required.

The PWM output is typically connected to a low-pass filter as shown in the figure below.

Figure 9.37 Typical PWM Interface Diagram



9.11.1.2 Use Cases

There are eight potential PWM use cases. Seven of the use cases are driven from PWM instances in LSIO, and one is driven from AON (for fan control). The following table summarizes the instances and their naming conventions.

Table 9.54 Use Case Summary and Naming Convention

PWM Interface	Use Case	Active in Low-Power State (Always On block)	Instance Summary	Clock Naming	Pinmux Naming[1]
PWM1	Display1 Backlight	No	Eight PWM ports, eight instances of PWM controller	pwm1_r_clk	soc_gpio54 (GP_PWM1)
PWM2	GPU OVR	No		pwm2_r_clk	gp_pwm2 (GP_PWM2)
PWM3	CV OVR/General Purpose	No		pwm3_r_clk	gp_pwm3 (GP_PWM3)
PWM4	FAN Control	Yes		pwm4_r_clk	TOUCH_CLK (GP_PWM4)
PWM5	CPU1 OVR/ General Purpose	No		pwm5_r_clk	soc_gpio12 (GP_PWM5)
PWM6	CPU0 OVR (or CPU0 and CPU1 OVR)	No		pwm6_r_clk	soc_gpio13 (GP_PWM6)
PWM7	SOC OVR	No		pwm7_r_clk	soc_gpio10 (GP_PWM7)
PWM8	Display2 Backlight	No		pwm8_r_clk	soc_gpio37 (GP_PWM8)

1. Refer to the pinout for the latest information.

9.11.2 Programming Guidelines

There are eight PWM controllers on eight individual pins on the chip. Each PWM controller has a single register PWM_CONTROLLER_PWM_CSR_0 (see PWM_CONTROLLER_PWM_CSR_0_x for the bit assignments). Each PWM must be enabled (bit 31) to be operational.

Pulse Width [23:16] determines the output pulse width and must be programmed appropriately. Only 8 bits are used for pulse width.

Below is the sequence for programming the PWM_CSR register:

1. Program the PWM_0 and PFM_0 fields based on the required Pulse Width and Frequency, respectively.
2. Set ENB to 0x1 to enable the PWM output.

In order to change the PWM_0 or PFM_0 fields, first program ENB to 0, program the required PWM_0 and PFM_0 fields, and then set ENB to 0x1.

For example, for the PWM to generate a 25-kHz pulse with 50% duty cycle, with a 38.4 MHz input clock:

- For frequency divisor:

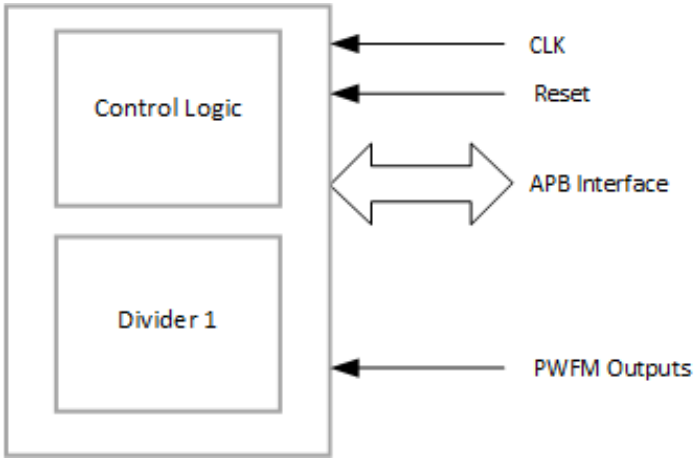
  Required frequency = (38.4 / 256) / (1 + PFM_0) = 150 kHz / (1 + PFM_0)
  25 kHz = 150 kHz / (1 + PFM_0)
  1 + PFM_0 = 6
  PFM_0 = 5

- For pulse width:

  Required pulse width (%)= (N/265) * 100
  50 = (N *100) / 256
  N = 128 = PWM_0

The top-level block diagram (shown below) has the control logic and frequency divider. The APB interface which directly goes to the control logic is also shown. The output is generated from the dividers, with each divider generating one output.

Figure 9.38 PWFM Functional Block Diagram



Note: In the frequency division diagram, the clock is divided by 256 and then by the frequency divider input.

9.11.3 PWM Registers

Refer to "Reading Register Tables" in the Introduction chapter for the register table protocol as well as recommendations for accessing registers.

PWM_CONTROLLER_PWM_CSR_0_0

There is one PWM controller for each individual PWM pin on the chip. For each PWM controller there is a corresponding register. PWM_CSR to program Pulse width and Frequency for that PWM output.

- Each PWM must be enabled [31] to be operational.
- Pulse Width [23:16] determines the output pulse width and must be programmed appropriately.
- Frequency Divider [12:0] determines the Divided clock.

Offset: 0x0
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,xxx0,0000,0000,0000)

Bit	Reset	Description
31	0x0	ENB: Enable Pulse width modulator 0 = DISABLE 1 = ENABLE
30:16	0x0	PWM_0: pulse width that needs to be programmed. 0 = Always low 1 = 1/256 Pulse high 2 = 2/256 Pulse high N = N/256 Pulse high
12:0	0x0	PFM_0: Frequency divider that needs to be programmed.

9.12 Fan Tachometer

9.12.1 Overview

The System-on-Chip (SoC) fan control solution makes use of two hardware modules: the tachometer input controller and PWM output controller. This chapter documents the tachometer input controller module. The main purpose of the fan tachometer input controller is to find the RPM of the fan, which enables control of the fan speed. The tachometer input controller monitors the period and high time of the fan tachometer input.

The tachometer solution is based on a four-wire fan: two wires for the power (V+, GND), one wire for the PWM input for fan speed control (this is an output of the chip), and one wire for the tachometer output for monitoring the revolutions per minute (RPM) (this is an input to chip).

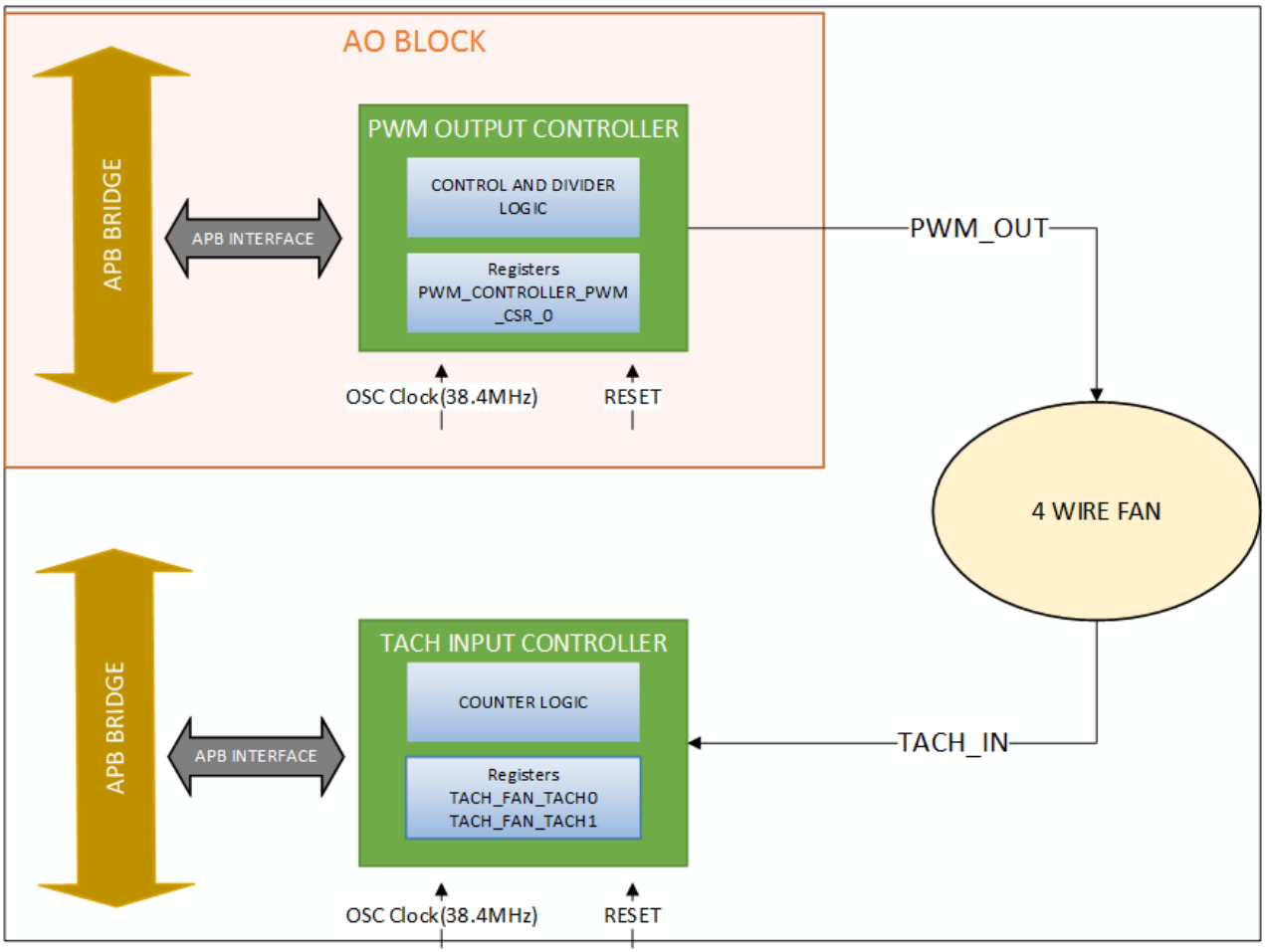
The tachometer input controller is in the LSIO cluster and the PWM output controller (which controls the fan speed) is in the AO block. The PWM controller fan must be in AO to allow a battery-powered device to be charged while in SC7 (also known as deep sleep). If the fan is not active in SC7, the device could get too warm during charging in some implementations. The fan control is active only when the fan is driven by PWM_OUT.

The temperature input functions as a wake event. If the temperature goes beyond/below maximum/minimum thresholds, the system exits from SC7, and the PWM output controller adjusts the fan speed according to the input from temperature versus RPM curve. The fan is not monitored through the tachometer but is controlled through PWM during SC7.

The tachometer input controller register logic interfaces with the APB bus. Additional instance of TACH is added for redundancy during functional state and not in low power states.

The figure below shows the placement of hardware modules:

Figure 9.39 Block Diagram of PWM and Tachometer



9.12.2 Functional Description

9.12.2.1 Fan Tachometer Clocks

The oscillator clock (38.4 MHz) serves as a clock source for the tachometer input controller. The 1-MHz clock is derived from the crystal, and is used as a tachometer counter clock. This 1-MHz clock divider is external to the tachometer module. The input to tachometer is the 1-MHz clock itself. The 1-MHz clock would actually be 38.4/38 or 19.2/19 which is 1.0105263 MHz. Software has to use this clock value in the RPM calculations.

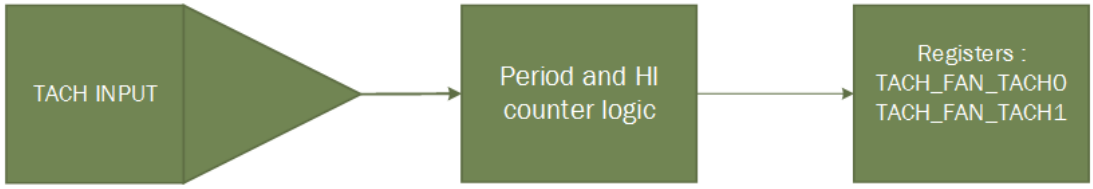
The PLLP, which is another clock source to the tachometer, cannot be used to derive the 1-MHz clock because of the 8-bit divider. The minimum input clock frequency is 3.18 MHz (408 MHz/2^7 MHz) when PLLP is used as a clock source. The clock value used in the RPM calculations should be adjusted accordingly.

As the tachometer input frequency range is 1-400 Hz, and the tolerable precision for the tachometer is < 0.1%, the minimum sampling clock should have frequency > 400 kHz. 1 MHz is chosen to get 1 µs time base for the counter for easy calculations. Operation is at a fixed frequency.

9.12.2.1.1 Fan Tachometer Input Controller

The figure below explains the tachometer input controller in detail.

Figure 9.40 High-Level View of Tachometer Input Controller



A 1-MHz counter is implemented in the period and HI counter logic which provides one microsecond time base. This counter increments for the duration of the window length of the tachometer input. If the counter overflows, it is indicated by the overflow bit. When the window length ends, the last counter value is latched into the software-accessible register.

The glitch filter in the GPIO unit is disabled here. The filter logic is programmable with a range of [1 ms to 128 ms] in steps of 1 ms. If this is programmed with X ms, it filters out glitches of width less than X+1 ms.

The tachometer input frequency range would be 1-400 Hz. So the minimum time period is 1/400 = 2.5 ms and the minimum width that can be programmed in the GPIO filter is 1 ms. If the duty cycle

goes below 50%, the glitch filter might filter out the entire pulse and so the glitch filter has to be disabled.

A basic RPM calculation is as follows: for a two-pulse per revolution (PPR) fan (WIN_LENGTH=1), if the fan returns 60 pulses per second, then it is 30 (60/2) revolutions per second, which is 1800 (30*60) RPM.

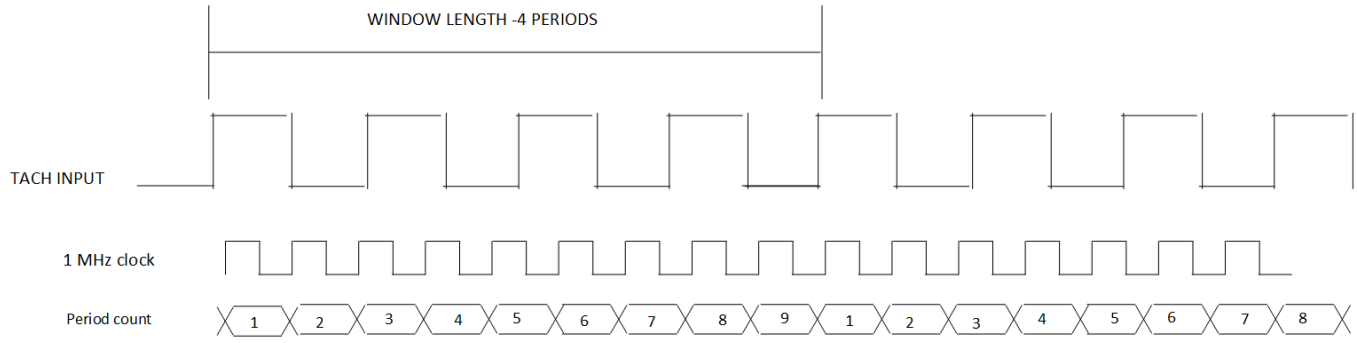
To capture the time period of the tachometer input for the duration of one window length:

$$RPM = \frac{Tach_source_clock_freq\ (MHz) \times 60 \times TACH_FAN_TACH0_0_WIN_LENGTH}{Tach_source_clock_programmed_divider \times (TACH_FAN_TACH0_0_PERIOD + 1) \times 10(-6) \times PPR}$$

In this case, the tachometer counter clock should be programmed to 1 MHz.

The figure below shows the RPM calculation in detail.

Figure 9.41 Tachometer Counter



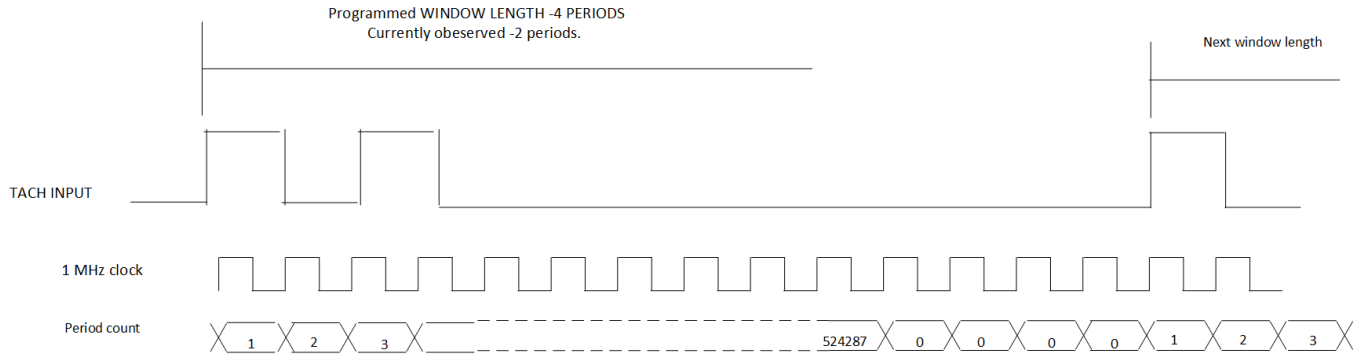
The window length is the number of tachometer input periods that are to be monitored. This is programmed by software using the WINDOW_LENGTH field of the TACH_FAN_TACH0 register. As shown in the above figure, the counter starts incrementing on the first rising edge on every window length and resets to 1 at the beginning of each window length. At the end of each window length, the period value is latched into PERIOD [18:0] field of TACH_FAN_TACH0 register. Software periodically reads these registers for RPM calculation. The reason for having WINDOW_LENGTH is that some FANs provide more than one pulse (2/4/8) for a single rotation. To reduce the overhead in software for these calculations, window length can be modified so that the value read from PERIOD is always the time taken for one rotation, making it easier for RPM calculations.

When the fan stops spinning, the rising edge of tachometer input cannot be seen, and the window length never ends. The period value reaches its maximum value. The overflow bit of the TACH_FAN_TACH0 register is set to “Detected” until software clears it. Tachometer monitoring is disabled until the next rising edge of the tachometer is detected.

To start counting, a pulse is generated indicating the detection of the 1st positive edge on the tachometer input. This pulse triggers the counter to start (starting from 0) and latches the period value when the next edge is detected. Since a 1 MHz clock is used to generate the trigger, the positive edge detection and triggering of counter takes one clock (one unit time), and the counter is run for N-1 unit time (N being the total time period of the pulse). So the counter reports N-1 value instead of N.

Change of the WIN_LENGTH field takes effect only when the current period calculation is complete. For example, if the current calculation counts four pulses (WIN_LEN = 2) and software programs this field to start counting one pulse (WIN_LEN = 0), the design completes the current PERIOD calculation based on WIN_LEN = 2 and then (for next PERIOD calculation) starts using WIN_LEN = 0.

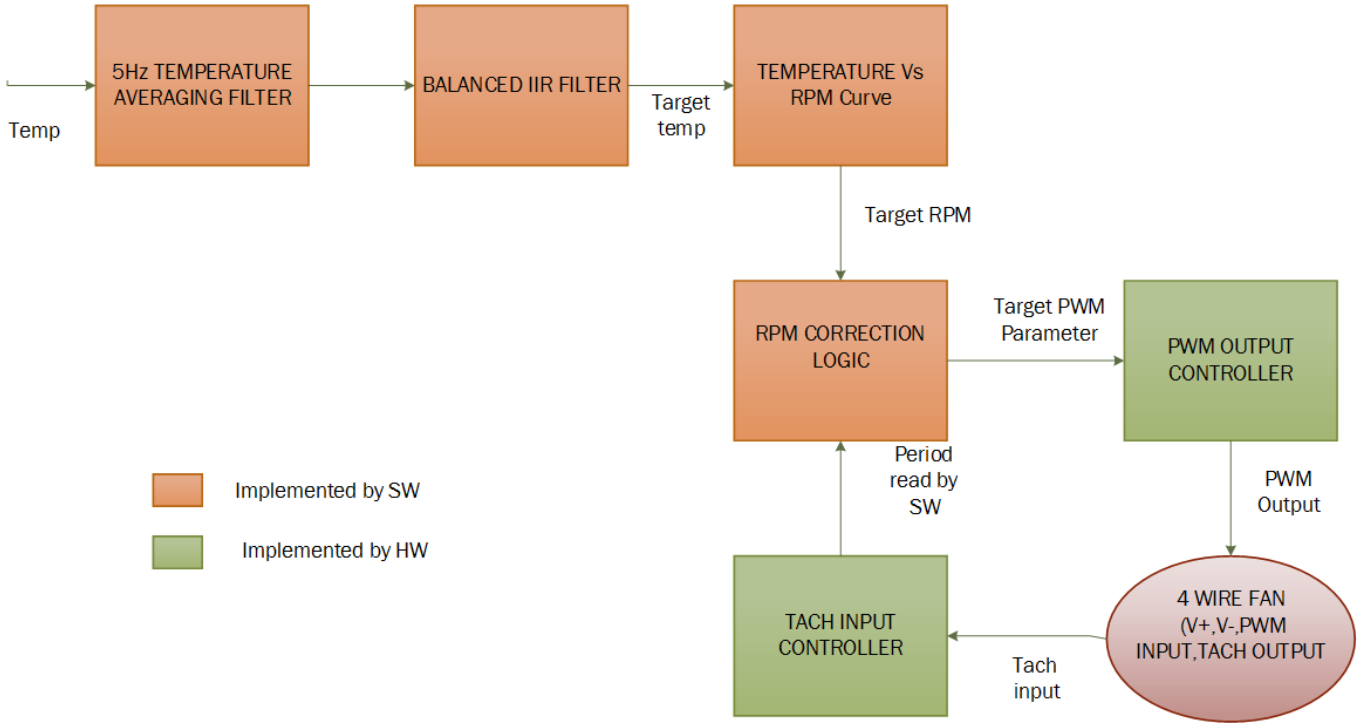
Figure 9.42 Overflow Scenario



9.12.2.1.2 Fan Control

The figure below provides the overview of the tachometer solution for fan control implementation.

Figure 9.43 Tachometer Solution for Fan Control



The input temperature is mapped to the RPM, based on the temperature vs RPM curve, in order to get the target RPM. The PWM output controller sets the period and duty cycle of PWM_OUT which is fed as an input to the four-wire fan.

Based on the tachometer input, the period and window length are stored in the tachometer registers. Software reads the data from these registers periodically, calculates the RPM, and compares with the target RPM. PWM parameters are determined in the RPM correction logic block and serves as an input to the PWM output controller.

Interrupt Functionality

If Fan RPMs (revolutions per minute) are not monitored, the device can run into power and thermal issues if the Fan RPM moves outside of an expected range. To detect this, interrupt functionality is added to the TACH controller. The TACH controller has the ability to assert interrupt when PERIOD in the WIN_LENGTH field or the number of revolutions programmed into MONITOR_TIME goes higher than UPPER_THRESHOLD or lower than LOWER_THRESHOLD. A configuration bit is provided for software to select the type of monitoring required.

For possible conditions when the Fan is spinning too fast:

- Fan inlet and or exhaust is blocked
- Damage such as dust

For possible conditions when the Fan is spinning too slow:

- Over aging Fan
- Physical damage to the fan

By default, the interrupt functionality is disabled and should be enabled by software after programming the thresholds. When any threshold violation happens, the ERR_STATUS register will capture the error condition (i.e., UNDERRUN or OVERRUN) and the period counter value (ERR_PERIOD) when the error was detected. Hardware provides a configuration bit (STOP_ON_ERR) for software to select if the controller should be stopped on error or continue in the next window.

Note: In case of a delay in servicing previous interrupt (e.g., UNDERRUN), if the hardware encounters another error condition (e.g., OVERRUN), hardware will clear the previous logged error from the status register and update the status register with the current error condition.

9.12.3 Programming Guidelines

For RPM calculation (no RPM monitoring required)

1. Program the WIN_LENGTH field in register TACH_FAN_TACH0_0 based on the Fan Pulse Per Rotations (PPR) value.
2. Read the PERIOD value from register TACH_FAN_TACH0_0 periodically to calculate RPM using RPM calculation.

To enable interrupt functionality to monitor the RPM variation for every window length

1. Program the WIN_LENGTH field in register TACH_FAN_TACH0_0 based on FAN PPR value.
2. Program the lower and upper limit of the PERIOD variation in units of micro-seconds in TACH_FAN_TACH_UPPER_THRESHOLD_0 and TACH_FAN_TACH_LOWER_THRESHOLD_0.
3. Enable UNDERRUN/OVERRUN/COUNTER_OVERFLOW interrupts by setting TACH_FAN_TACH_INTERRUPT_ENABLE_0 fields.
4. Write ‘0’ to TACH_FAN_TACH_CONTROL_0_ERR_CONFIG.
5. Program TACH_FAN_TACH_CONTROL_0_STOP_ON_ERR based on the requirements and set FAN_TACH_CONTROL_0_LOAD_CONFIG.

To enable interrupt functionality to monitor the RPM variation by monitoring the number of pulses over time

1. Program the lower and upper limit of the expected number of pulses in TACH_FAN_TACH_UPPER_THRESHOLD_0 and TACH_FAN_TACH_LOWER_THRESHOLD_0.
2. Program the time duration (in micro-seconds) during which the TACH input is monitored in TACH_FAN_TACH_CONTROL_0_MONITOR_TIME.
3. Enable UNDERRUN/OVERRUN/COUNTER_OVERFLOW interrupts by setting TACH_FAN_TACH_INTERRUPT_ENABLE_0 fields.
4. Write ‘1’ to TACH_FAN_TACH_CONTROL_0_ERR_CONFIG.

5. Program TACH_FAN_TACH_CONTROL_0_STOP_ON_ERR based on the requirements and set FAN_TACH_CONTROL_0_LOAD_CONFIG.

9.12.3.1 Interrupt Handling

1. When software receives an interrupt from the controller, read the TACH_FAN_TACH_0_ERR_STATUS register for the source of interrupt.
2. Software can read the TACH_FAN_TACH_0_ERR_PERIOD register to know the period value/ pulse count captured by the TACH controller when the error occurred.
3. Software can clear the interrupt by writing '1' to the error field in TACH_FAN_TACH_0_ERR_STATUS.

9.12.4 Fan Tachometer Registers

Refer to "Reading Register Tables" in the Introduction chapter for the register table protocol as well as recommendations for accessing registers.

The Base Addresses of the registers related to Fan Tachometer are specified in the Address Map section of the TRM.

TACH_FAN_TACH0_0
Offset: 0x0
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,x000,0000,0000,0000,0000,0000,0000)

Bit	R/W	Reset	Description
26:25	RW	0x0	**WIN_LENGTH**: This field defines the window of the FAN TACH monitor. The window indicates how many periods of the input fan tach signal that the FAN TACH logic will monitor. The window starts from the rising edge of the fan tach signal. ONE: Set window length to 1. Monitor only 1 period. TWO: Set window length to 2. Monitor the consecutive two periods. FOUR: Set window length to 4. Monitor the consecutive four periods. EIGHT: Set window length to 8. Monitor the consecutive eight periods. This is needed as the fan generates Pulses per rotation (PPR) which are detected by the input. For example, in a 2PPR Fan, in one time period, we might be covering 172 degrees. So we should measure for two cycles in this case. Similarly in 4PPR, it should be four cycles. Change of WIN_LENGTH field on the fly takes effect only when the current PERIOD calculation is complete. **0** = ONE **1** = TWO **2** = FOUR **3** = EIGHT

Bit	R/W	Reset	Description
24	RW	0x0	**OVERFLOW**: This field indicates the window period of the fan tach input exceeds 16s. This happens if the window period is too long. When the hardware first detects the period between rising edges exceeding 16s, this field is set to DETECTED, and PERIOD is set to MAX. Monitoring of the fan tach input is disabled until the next window rising edge is detected. This field remains at DETECTED until Software writes it to CLEAR, only then will it return to NONE. (Writing 1 clears the bit and writing 0 keeps overflow bit unchanged) **0** = NONE **1** = DETECTED
23:0	RO	0x0	**PERIOD**: This field indicates the period (N-1) of the fan tach input in units of 1 micro-second in a WIN_LENGTH. Software needs to add 1 to this value to get the correct period value. This field updates on the first rising edge of the fan tach input for every new window.

TACH_FAN_TACH1_0

Offset: 0x4
Read/Write: RO
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:0	0x0	**HI**: This field indicates the HIGH time (N-1) of the fan tach input in units of 1 micro-seconds of every fan tach monitor window. It is updated on the first rising edge of the fan tach input of every new window.

TACH_FAN_TACH_UPPER_THRESHOLD_0

Offset: 0x8
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00ffffff (0bxxxx,xxxx,1111,1111,1111,1111,1111,1111)

Bit	Reset	Description
23:0	0xffffff	**UPPER_THRESHOLD**: For ERR_CONFIG=MONITOR_PERIOD - The threshold value has to be programmed in micro-seconds units For ERR_CONFIG=MONITOR_PULSES - The threshold value has to be programmed in terms of number of pulses

TACH_FAN_TACH_LOWER_THRESHOLD_0

Offset: 0xc
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,0000,0000,0000,0000,0000,0000)

Bit	Reset	Description
23:0	0x0	**LOWER_THRESHOLD**: For ERR_CONFIG=MONITOR_PERIOD - The threshold value has to be programmed in micro-seconds units For ERR_CONFIG=MONITOR_PULSES - The threshold value has to be programmed in terms of number of pulses

TACH_FAN_TACH_INTERRUPT_ENABLE_0

Offset: 0x10
Read/Write: R/W
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0bxxxx,xxxx,xxxx,xxxx,xxxx,xxxx,xxxx,x000)

Bit	Reset	Description
2	0x0	**COUNTER_OVERFLOW**: When enabled, an interrupt is asserted when OVERFLOW bit is set. By default the interrupt functionality is disabled and should be enabled by Software. **0** = DISABLE **1** = ENABLE
1	0x0	**UNDERRUN**: When enabled, an interrupt is asserted when UNDERRUN status is set. By default the interrupt functionality is disabled and should be enabled by Software after programming the lower threshold. **0** = DISABLE **1** = ENABLE
0	0x0	**OVERRUN**: When enabled, an interrupt is asserted when OVERRUN status is set. By default the interrupt functionality is disabled and should be enabled by Software after programming the upper threshold. **0** = DISABLE **1** = ENABLE

TACH_FAN_TACH_CONTROL_0

Offset: 0x14
Read/Write: R/W
Parity Protection: N
Shadow: N

SCR Protection: 0
Reset: 0xffffff00 (0b1111,1111,1111,1111,1111,1111,xxxx,x000)

Bit	Reset	Description
31:8	0xffffff	**MONITOR_TIME**: Time in micro-seconds that the controller counts the incoming pulses and compares against UPPER_THRESHOLD/LOWER_THRESHOLD when time expires. This field is used when ERR_CONFIG=MONITOR_PULSES
2	0x0	**ERR_CONFIG**: Hardware can monitor the tach input and detect variation in RPM. This can be done in two ways: 1. MONITOR_PERIOD - Detect variation in RPM by monitoring the tach input period for the programmed win_length (absolute RPM monitor). 2. MONITOR_PULSES - Detect variation in RPM by monitoring the tach input pulses (number of rising edges of tach input) for a programmed time (MONITOR_TIME). In this configuration, Hardware does not wait for a tach pulse to be detected (average RPM monitor). **0** = MONITOR_PERIOD **1** = MONITOR_PULSES
1	0x0	**STOP_ON_ERR**: If stop_on_error field is set, Tach will wait until the interrupt is serviced and cleared by Software, else it will continue monitoring. **0** = DISABLE **1** = ENABLE
0	0x0	**LOAD_CONFIG**: Software is required to set this bit to 1 after programming the threshold values and ERR_CONFIG into respective registers. Setting the bit to 1 triggers Hardware to update the threshold values internally and reset the current counters. Hardware starts counting the period on the next pulse from fan when ERR_CONFIG = MONITOR_PERIOD or Hardware starts counting the pulses immediately when ERR_CONFIG = MONITOR_PULSES. Once the internal update is done, Hardware auto clears this bit. Since Hardware will be busy with internal update, Software should not write the threshold register or ERR_CONFIG again until this bit is cleared by Hardware. **0** = DISABLE **1** = ENABLE

TACH_FAN_TACH_ERR_STATUS_0

Offset: 0x18
Read/Write: See table below
Parity Protection: N
Shadow: N
SCR Protection: 0
Reset: 0x00000000 (0b0000,0000,0000,0000,0000,0000,xxxx,xx00)

Bit	R/W	Reset	Description
31:8	RO	0x0	**ERR_PERIOD**: ERR_CONFIG=MONITOR_PERIOD - Period value (N-1) in µs is captured when error is detected ERR_CONFIG=MONITOR_PULSES - Number of pulses are captured when error is detected
1	RW	0x0	**UNDERRUN**: This field is set when the PERIOD value or the number of pulses is lower than or equal to programmed LOWER_THRESHOLD value. **0** = NO_UNDERRUN **1** = UNDERRUN
0	RW	0x0	**OVERRUN**: This field is set when the PERIOD value or the number of pulses is greater than or equal to programmed UPPER_THRESHOLD value. **0** = NO_OVERRUN **1** = OVERRUN

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation (“NVIDIA”) makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality. NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer (“Terms of Sale”). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer’s own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer’s sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer’s product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, “MATERIALS”) ARE BEING PROVIDED “AS IS.” NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA’s aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

Trademarks

NVIDIA, the NVIDIA logo, are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

VESA DisplayPort

DisplayPort and DisplayPort Compliance Logo, DisplayPort Compliance Logo for Dual-mode Sources, and DisplayPort Compliance Logo for Active Cables are trademarks owned by the Video Electronics Standards Association in the United States and other countries.

HDMI

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC.

Arm

Arm, AMBA, and Arm Powered are registered trademarks of Arm Limited. Cortex, MPCore, and Mali are trademarks of Arm Limited. All other brands or product names are the property of their respective holders. "Arm" is used to represent Arm Holdings plc; its operating company Arm Limited; and the regional subsidiaries Arm Inc.; Arm KK; Arm Korea Limited.; Arm Taiwan Limited; Arm France SAS; Arm Consulting (Shanghai) Co. Ltd.; Arm Germany GmbH; Arm Embedded Technologies Pvt. Ltd.; Arm Norway, AS, and Arm Sweden AB.

OpenCL

OpenCL is a trademark of Apple Inc. used under license to the Khronos Group Inc.

Copyright

© 2023 NVIDIA Corporation. All rights reserved.

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
http://www.nvidia.com